

Automatic Conversion of Natural Language to 3D Animation

Minhua Ma

B.A., M.A., M.Sc.

Faculty of Engineering

University of Ulster

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

July 2006

Table of Contents

List of Figures	vi
List of Tables	ix
Acknowledgements	x
Abstract	xi
Abbreviations	xii
Note on access to contents	xiv
1. INTRODUCTION.....	1
1.1 Overview of language visualisation	2
1.1.1 Multimodal output.....	2
1.1.2 Animation.....	3
1.1.3 Intelligent	4
1.2 Problems in language visualisation	4
1.3 Objectives of this research.....	5
1.4 Outline of this thesis	5
2. APPROACHES TO MULTIMODAL PROCESSING.....	8
2.1 Automatic text-to-graphics systems	8
2.1.1 CarSim	9
2.1.2 WordsEye.....	9
2.1.3 Micons and CD-based language animation.....	12
2.1.4 Spoken Image and SONAS	12
2.2 Embodied agents and virtual humans	14
2.2.1 Virtual human standards	15
2.2.2 Virtual humans	19
2.2.3 BEAT and other interactive agents	20
2.2.4 Divergence on agents' behaviour production.....	21
2.2.5 Gandalf.....	22
2.2.6 Humanoid Animation.....	23
2.3 Multimodal storytelling.....	23
2.3.1 Interactive storytelling	24
2.3.2 AESOPWORLD	25
2.3.3 Oz.....	26
2.3.4 KidsRoom	27
2.3.5 Computer games.....	28
2.3.6 Film-inspired computer animations	29
2.3.7 Computer graphics films	31
2.3.8 Video-based computer animation generation.....	31
2.4 Summary of previous systems.....	32
2.5 Multimodal allocation	32
2.6 Non-speech audio	35
2.6.1 Auditory icons.....	36
2.6.2 Earcons.....	37
2.6.3 Sonification	37
2.6.4 Music synthesis	37
2.7 Mental imagery in cognitive science.....	39

2.8 Summary	40
3. NATURAL LANGUAGE SEMANTICS	41
3.1 Natural language semantic representations	41
3.1.1 Semantic networks	41
3.1.2 Conceptual Dependency theory and scripts	41
3.1.3 Lexical Conceptual Structure (LCS)	45
3.1.4 Event-logic truth conditions	46
3.1.5 X-schemas and f-structs	47
3.2 Multimodal semantic representations	49
3.2.1 Frame representation and frame-based systems	49
3.2.2 XML representations	50
3.2.3 Summary of knowledge representations	52
3.3 Temporal relations	52
3.3.1 Punctual events	55
3.3.2 Verb entailment and troponymy	56
3.4 Computational lexicons	56
3.4.1 WordNet	57
3.4.2 FrameNet	58
3.4.3 The LCS database and VerbNet	59
3.4.4 Comparison of lexicons	61
3.4.5 Generative lexicon	62
3.5 Language ontology	62
3.5.1 Top concepts	62
3.5.2 Ontological categories of nouns	63
3.5.3 Ontological categories of verbs	65
3.6 Summary	68
4. LEXICAL VISUAL SEMANTIC REPRESENTATION	69
4.1 Multimodal representation	69
4.2 Ontological categories of concepts (conceptual “parts of speech”)	70
4.3 Lexical Visual Semantic Representation (LVSR)	71
4.3.1 Finer EVENT predicates	74
4.3.2 Under-specification and selection restrictions	76
4.4 Visual semantics of events	78
4.4.1 Action decomposition	78
4.5 Temporal relationships in language	79
4.5.1 Sentence level temporal relationship	80
4.5.2 Temporal relations in lexical semantics of verbs	84
4.6 Categories of nouns for visualisation	91
4.7 Visual semantic representation of adjectives	92
4.7.1 Categories of adjectives for visualisation	92
4.7.2 Semantic features of adjectives relating to visualisation	94
4.7.3 Entity properties for visual and audio display	96
4.8 Visual semantic representation of prepositions	97
4.8.1 LVSR definitions and semantic features of spatial prepositions	98
4.8.2 Distributive feature for location and motion	99
4.8.3 Semantic representation of prepositional phrases	99
4.9 Semantic field of verbs and prepositions	100
4.10 Summary	102
5. NATURAL LANGUAGE VISUALISATION	103
5.1 Problems in language visualisation	103

5.2 Verb ontology and visual semantics	105
5.2.1 Visual valency	105
5.2.2 Somatotopic factors in visualisation	106
5.2.3 Level-Of-Detail (LOD) — Basic-level verbs and their troponyms.....	107
5.2.4 Visual semantic based verb ontology.....	108
5.3 Verb ontology and audio semantics	111
5.4 Word Sense Disambiguation	112
5.5 Commonsense reasoning using WordNet.....	116
5.6 Negation	118
5.7 Summary	119
6. 3D ANIMATION GENERATION	120
6.1 The structure of animation generation.....	120
6.2 Virtual human animation	121
6.2.1 Animated narrator	121
6.2.2 Agents and avatars – how much autonomy?.....	122
6.2.3 Animating human motions	123
6.2.4 Virtual character animation with the H-Anim standard	125
6.2.5 Simultaneous animations and multiple animation channels.....	126
6.2.6 Facial expression and lip synchronisation.....	129
6.2.7 Space sites of virtual humans	130
6.2.8 Multiple character synchronisation and coordination	131
6.3 Object modelling	132
6.3.1 Default attributes in object visualisation	132
6.3.2 Space sites of 3D objects and grasping hand postures	133
6.4 Collision detection.....	136
6.5 Automatic camera placement	138
6.6 Summary	140
7. CONFUCIUS: AN INTELLIGENT MULTIMEDIA STORYTELLING SYSTEM	141
7.1 Architecture of CONFUCIUS	141
7.2 Input and output.....	142
7.3 Knowledge base	143
7.4 NLP in CONFUCIUS.....	146
7.4.1 Syntactic parsing	147
7.4.2 Semantic analysis	150
7.4.3 Using WordNet for semantic inference and WSD	152
7.4.4 Action representation	155
7.4.5 Representing active and passive voice	155
7.5 Media allocation	156
7.6 Animation engine	158
7.6.1 3D object modelling	162
7.6.2 Virtual human animation.....	165
7.6.3 Java in VRML Script node.....	167
7.6.4 Applying narrative montage to virtual environments.....	168
7.7 Text-to-speech	169
7.8 Using presentation agents to model the narrator	170
7.9 CONFUCIUS worked examples	173
7.10 Summary	178
8. EVALUATION	180
8.1 Subjective evaluation of animation generation	180
8.1.1 Subjects	180

8.1.2 Questionnaires	181
8.1.3 How to choose candidate words?	183
8.1.4 Results	184
8.1.5 Comparison with computer games	185
8.2 Diagnostic Evaluations	185
8.3 Syntactic parsing	186
8.3.1 Subordinate clauses without conjunctions	186
8.3.2 PP attachment	187
8.3.3 Coordination and ellipsis	187
8.3.4 Word order and discontinuous constituents	188
8.4 Semantic analysis	189
8.5 Anaphora resolution	190
8.6 Summary	191
9. CONCLUSION AND FUTURE WORK	192
9.1 Summary	192
9.2 Relation to other work	194
9.2.1 Comparison with previous systems	194
9.2.2 Comparison of LVSR and LCS	196
9.2.3 Comparison of interval algebra and Badler's temporal constraints	197
9.2.4 Comparison of visual semantic ontology and Levin's classes	197
9.2.5 Comparison of action decomposition and scripts	198
9.2.6 Commonsense knowledge reasoning	199
9.3 Future work	199
9.4 Conclusion	201
APPENDICES	203
Appendix A. A VHML example from virtual storyteller	204
Appendix B. Working files of semantic analysis	206
B.1 Analysis of Machine Syntax for English:	206
B.2 The output file after adding semantic features	207
Appendix C. LCS notation	208
C.1 Logical arguments	208
C.2 Logical modifiers	209
C.3 Thematic role specification	210
Appendix D. Connexor FDG notation	211
D.1 English dependency functions	211
D.2 English morphological tags	213
D.3 English functional tags	217
D.4 English surface syntactic tags	219
Appendix E. List of existing H-Anim models	220
Appendix F. H-Anim example code	221
F.1 Example code of an animation file	221
F.2 Nana's joints list	223
Appendix G. CONFUCIUS' evaluation questionnaire	225
Appendix H. Test set for syntactic analysis (from HORATIO)	231
Appendix I. Test set for anaphora resolution	235
Appendix J. Test set for semantic analysis of verbs	237
Appendix K. Publications	240
REFERENCES	241

List of Figures

Figure 1.1: Simulation of cognition, communication, and re-cognition	2
Figure 1.2: Multimodal I/O of CONFUCIUS	5
Figure 2.1: CarSim GUI and created 3D scene	9
Figure 2.2: An example of a WordsEye generated picture	10
Figure 2.3: Verb frames of <i>say</i> in WordsEye	11
Figure 2.4: An example story: going to a restaurant (Narayanan et al. 1995)	13
Figure 2.5: An example of human-computer interaction in SI	13
Figure 2.6: Motion classes in SONAS	14
Figure 2.7: Four levels of virtual human representation	15
Figure 2.8: Representing parallel temporal relation	18
Figure 2.9: REA and BEAT	20
Figure 2.10: Dialogue between two autonomous agents	21
Figure 2.11: Gandalf's interface	23
Figure 2.12: Executable story script of Larsen's storytelling system	25
Figure 2.13: The <i>Edge of Intention</i> world in OZ	27
Figure 2.14: KidsRoom	28
Figure 2.15: Functional unification formalism in COMET	35
Figure 2.16: Four types of non-speech audio	38
Figure 2.17: Mental architecture and meaning processing	39
Figure 3.1: Conceptual Dependency primitives	42
Figure 3.2: Conceptual Dependency representation of "I gave John a book."	42
Figure 3.3: Conceptual structures from ontological categories in LCS	45
Figure 3.4: <i>slide</i> x-schema (Bailey et al. 1997)	48
Figure 3.5: Multimodal semantic representations and visual semantic representations	53
Figure 3.6: The ontology of MOOSE	55
Figure 3.7: Fellbaum's classification of verb entailment	56
Figure 3.8: Basic relations between synsets in WordNet	58
Figure 3.9: Bipolar adjective structure in WordNet (Gross and Miller 1990)	58
Figure 3.10: A verb entry of "cut" in LCS database	60
Figure 3.11: A verb entry of "cut" in VerbNet	61
Figure 3.12: Hierarchy of top concepts in EuroWordNet (Vossen et al. 1998)	63
Figure 3.13: Major semantic clusters of nouns in WordNet	64
Figure 3.14: Noun tops in WordNet	64
Figure 3.15: Dimension of causation-change	67
Figure 4.1: Multimodal semantic representation	70
Figure 4.2: Predicate-argument forms of some conceptual categories	72
Figure 4.3: Alternate readings between STATE and EVENT	73
Figure 4.4: Examples of lexical entries	74
Figure 4.5: Incorporated arguments of verb entries	74
Figure 4.6: LCS representation of "John sat on the chair"	75
Figure 4.7: LCS examples of phrasal causations and lexical causatives	75
Figure 4.8: LVSR examples of lexical causatives	76

Figure 4.9: LCS and LVSR of “John sat on the chair”	76
Figure 4.10: LCS and LVSR of “The weathervane pointed north”	76
Figure 4.11: Lexical entries for “enter”	77
Figure 4.12: Using ontology categories to differentiate transitive and intransitive verbs	77
Figure 4.13: Visual definition and word sense	78
Figure 4.14: The action decomposition structure.....	79
Figure 4.15: A troponymy tree.....	85
Figure 4.16: Decomposition of “turn” using interval algebra.....	88
Figure 4.17: Visual definitions of “eatOut”	89
Figure 4.18: Examples of punctual events’ visual definitions	89
Figure 4.19: Verbs defined by repeatable subactivities	90
Figure 4.20: Concrete noun categories	92
Figure 4.21: Categories of adjectives.....	93
Figure 4.22: Example LVSR adjective entries.....	95
Figure 4.23: The relation between spatial prepositions and geometric configurations.....	98
Figure 4.24: The distributive feature of prepositions.....	101
Figure 4.25: Semantic representation of temporal information	102
Figure 5.1: Somatotopic effectors of some action verbs.....	107
Figure 5.2: Hierarchical tree of verbs of motion.....	108
Figure 5.3: Verb ontology based on visual semantics.....	109
Figure 5.4: Verb ontology for audio semantics.....	112
Figure 5.5: The hypernym tree of “lancet” in WordNet	116
Figure 5.6: Hypernym tree of “knife” in WordNet.....	117
Figure 5.7: Hypernym trees of “flower”, “jewel”, and “flower arrangement”	118
Figure 6.1: Structure Diagram of an <i>animation producer</i>	121
Figure 6.2: Autonomy of virtual humans.....	123
Figure 6.3: Joints and segments of LOA2.....	126
Figure 6.4: “Nancy ran across the field.”	127
Figure 6.5: An example of motion integration.....	128
Figure 6.6: Cardinal vowels quadrilateral.....	130
Figure 6.7: Lip synchronisation of viseme a, i, o.....	130
Figure 6.8: Site nodes on the hands and feet of a virtual human	131
Figure 6.9: Site nodes around a virtual human’s body.....	131
Figure 6.10: Alice in Wonderland: Down the Rabbit-hole.....	133
Figure 6.11: Space sites of a 3D desk	134
Figure 6.12: Verbs of hand movements	134
Figure 6.13: Four hand postures for physical manipulation of objects.....	136
Figure 6.14: Bounding volumes.....	137
Figure 6.15: Over-the-shoulder shot for presenting verbs of communication	138
Figure 6.16: Viewpoints of the animation “Bob left the gym”	139
Figure 7.1: Architecture of CONFUCIUS	142
Figure 7.2: Input/output of CONFUCIUS	143
Figure 7.3: Knowledge base of CONFUCIUS.....	144
Figure 7.4: Composition of the graphic library.....	146
Figure 7.5: Composition of CONFUCIUS’ NLP module.....	147
Figure 7.6: Connexor output for “Jack put the jug in his pocket.”	149
Figure 7.7: Example verb entries in the LCS database	151
Figure 7.8: Dependency tree of “Once upon a time there was a poor widow.”	151
Figure 7.9: Using “value of” relation to look for adjectives’ properties.....	153
Figure 7.10: Verb-frames in WordNet.....	153

Figure 7.11: Theta roles and verb frames of hit verbs	154
Figure 7.12: Relations with features to differentiate semantic implication	155
Figure 7.13: Event structures used in CONFUCIUS	156
Figure 7.14: CONFUCIUS' multimedia presentation planning	157
Figure 7.15: Flowchart of animation engine algorithm	159
Figure 7.16: Example VRML code for <i>following</i> verbs	160
Figure 7.17: Snapshot of animation for <i>following</i> verbs.....	160
Figure 7.18: Example VRML code for <i>chasing</i> verbs	161
Figure 7.19: Two situations with same effect by placing appropriate viewpoints.....	162
Figure 7.20: Examples of ROUTE statement.....	163
Figure 7.21: Examples of Viewpoint node.....	164
Figure 7.22: External prototype of H-Anim animation.....	165
Figure 7.23: Nancy's joints list.....	166
Figure 7.24: Applying one animation on two H-Anim bodies.....	167
Figure 7.25: <i>The Tortoise and the Hare</i> from Aesop's Fables	172
Figure 7.26: Narrator Merlin speaks alongside a story	172
Figure 7.27: The HTML code of Figure 7.26	173
Figure 7.28: Connexor parser output of "John put a cup on the table."	174
Figure 7.29: After removing HTML format and adding semantic marking	174
Figure 7.30: After semantic analysis and WSD	175
Figure 7.31: After verb replacement	175
Figure 7.32: The output animation of "John put a cup on the table."	175
Figure 7.33: Connexor parser output of "John left the gym."	176
Figure 7.34: After removing HTML format and adding semantic marking	176
Figure 7.35: After semantic analysis and WSD	176
Figure 7.36: After verb replacement	176
Figure 7.37: The output animation of "John left the gym."	176
Figure 7.38: Connexor parser output of "The waiter came to me: 'Can I help you, Sir?'"	177
Figure 7.39: After removing HTML format and adding semantic marking	177
Figure 7.40: After semantic analysis and WSD	177
Figure 7.41: After verb replacement.....	178
Figure 7.42: The output animation of "The waiter came to me: 'Can I help you, Sir?'"	178
Figure 8.1: A screenshot of the evaluation questionnaire	181
Figure 8.2: An example of word-level agreement rating	182
Figure 8.3: An example of sentence-level agreement rating.....	182
Figure 8.4: An example of word-level comprehension measurement	182
Figure 8.5: An example of sentence-level comprehension measurement.....	183
Figure 8.6: Dependency tree of "He is sure I will tell them what to read"	186
Figure 8.7: Dependency tree of "Have you read the letter to the teacher about the library?"	187
Figure 8.8: Dependency tree of "The man reading a book in the library is a teacher."	187
Figure 8.9: Dependency tree of "John and the students want to put off the workshop."	188
Figure 8.10: Dependency tree of "Mary teaches linguistics and John mathematics."	188
Figure 8.11: An example of (ROOT, NP, PARTICLE).....	188
Figure 8.12: "They took the problems he had seen into account."	189
Figure 8.13: "They took into account the problems they had seen."	189
Figure 8.14: Comparing Resnik's WSD approach with baselines	190
Figure 8.15: An incorrect entry of the verb "know" in the LCS database	191
Figure 9.1: Comparison of related systems.....	195
Figure 9.2: Relation of CONFUCIUS' verb ontology and Levin's verb classes	198
Figure 9.3: The physics engine in animation generation	200

List of Tables

Table 1.1: Relating a storytelling system to Aristotle's six parts of a Tragedy.....	3
Table 2.1: MPEG4 visemes	17
Table 2.2: MPEG4 expressions.....	17
Table 2.3: Summary of the I/O of related systems	33
Table 3.1: F-struct of one verb sense of push using slide x-schema.....	48
Table 3.2: Categories of knowledge representations	53
Table 3.3: Allen's thirteen interval relations	54
Table 3.4: Frames and FEs of "drive" in FrameNet.....	59
Table 3.5: Comparison of verb lexicons	61
Table 3.6: Some linguistic subcategoriation frames and example verbs	65
Table 3.7: WordNet verb files	67
Table 4.1: The definition of PATH and PLACE predicates	73
Table 4.2: Temporal boundedness of events.....	83
Table 4.3: Temporal relations in verb entailments	86
Table 4.4: Launching and entraining causation	90
Table 4.5: Examples of graded adjectives	95
Table 4.6: Visually representable properties.....	96
Table 4.7: Audio representable properties	97
Table 4.8: LVSR definitions of prepositions	99
Table 4.9: Jackendoff's basic conceptual clause modification	100
Table 4.10: Temporal relations of clause modification	100
Table 5.1: Word senses of "leave" ruled out by semantic analyser	115
Table 5.2: Default instruments of verbs.....	117
Table 6.1: The animation registration table	128
Table 6.2: Three simplified visemes.....	129
Table 6.3: Taxonomy of ergotic hand movements	135
Table 6.4: Predefined viewpoints of 3D props and characters.....	140
Table 7.1: Character Merlin specification.....	171
Table 8.1: Results of the comprehension measures for animations 1-10.....	184
Table 8.2: Subjective agreement rating results for animations 11-18.....	184
Table 8.3: General linguistic phenomena	186
Table 9.1: Comparison of conceptual categories of Jackendoff's LCS and LVSR	196
Table 9.2: Comparison of interval algebra and Badler's temporal constraints	197
Table 9.3: Comparison of CONFUCIUS' action decomposition and scripts	199

Acknowledgements

First and foremost I would like to thank Prof. Paul Mc Kevitt. As my supervisor, Paul has contributed greatly to my work and life here in Derry/Londonderry. Paul has guided me into the promising area of Intelligent Multimedia. His suggestions and guidance have helped me tremendously in my research within the past four years.

Prof. Sally McClean, Head of Graduate School, and Prof. Mike McTear provided invaluable comments on my confirmation report. Many thanks to those who commented on my papers and presentations at conferences and workshops, especially to Dr. David McSherry, Dr. Norman Creaney, Dr. Tim Fernando, Prof. Harry Bunt, Dr. Francisco Azuaje, and Peter Fröhlich. Also I am grateful to the staff and colleagues, especially Dr. Michael McNeill, Tony Solon, and Glenn Campbell, for their help and encouragement, and Ted Leath, Bernard McGarry and Pat Kinsella, for their technical support.

Last but not least, my wholehearted thanks go: to my husband Jim, for his patience and understanding; to my baby daughter Brenda, who has witnessed the final phase and cheered it up with her cuteness; to my parents and sister, for showing the way in life and for always being there.

Abstract

The purpose of this research is to investigate the process of mental imagery from a computational perspective, employing theories and resources from linguistics, natural language processing, and computer graphics about human language visualisation. This thesis presents our progress toward the automatic creation of 3D animation from natural language text. Lexical Visual Semantic Representation (LVSR) is proposed, which connects linguistic semantics to the visual semantics and is suitable for action execution (animation). We investigate visual semantics of verbs, and introduce the notion of visual valency which is used as a primary criterion to construct a visual semantic based ontology. The visual valency approach is a framework for modelling deeper semantics of verbs. Lexicon-based approaches used for word sense disambiguation are also discussed. The context and the senses of the ambiguous verb are analysed using hypernymy relations and word frequency information in WordNet and thematic roles in LCS (Lexical Conceptual Structure) database. The significance of this research is also related to an animation blending approach which combines precreated and dynamically generated animation facilities into a unified mechanism and an object-oriented object modelling approach for decentralising the control of animation engine.

An intelligent storytelling system called CONFUCIUS, which visualizes single sentences into 3D animation, speech, and sound effects, has been implemented in Java and VRML. CONFUCIUS is an overall framework of language visualisation, using computer graphics techniques with NLP to achieve high-level animation generation. We conducted an evaluation experiment where subjects were asked to complete a questionnaire either rating agreement for the generated animation or selecting the closest text from four candidates which describes the animation best. The results show a low error rate of comprehension measures of animation (8.33%) and 3.82 average agreement score. We also evaluated the syntactic parsing by test-suite based diagnostic evaluation, and anaphora resolution and semantic analysis by corpus-based adequacy evaluation. CONFUCIUS gives promising results on word sense disambiguation (70% accuracy) with regard to the dataset it is tested on. Future work is suggested for extending the knowledge base and improving commonsense reasoning from lexicons to present more verb classes, extending language visualisation to discourse level, and applying physics, such as dynamics and kinematics, to 3D animation.

Keywords: natural language processing, visual semantics, language visualisation, 3D computer animation, virtual environment, storytelling, intelligent multimedia

Abbreviations

AML	Avatar Markup Language
AP	Adjective Phrase
API	Application Programming Interface
BAML	Body Animation Markup Language
BAP	Body Animation Parameter
BDP	Body Deformation Parameter
BIFS	Binary Format for Scenes
BNC	British National Corpus
CD	Conceptual Dependency
CFG	Context-Free Grammar
CG	Computer Graphics
DMML	Dialogue Manager Markup Language
EAI	External Authoring Interface
EML	Emotion Markup Language
FAML	Facial Animation Markup Language
FAP	Facial Animation Parameter
FDG	Functional Dependency Grammar
FDP	Face Deformation Parameter
GB	Government-Binding
GPSG	Generalized Phrase Structure Grammar
GUI	Graphic User Interface
H-Anim	Humanoid Animation
HTML	Hyper Text Markup Language
HPSG	Head-Driven Phrase Structure Grammar
IK	Inverse Kinematics
JSAPI	Java Speech Application Programming Interface
JSML	Java Speech API Markup Language
KB	Knowledge Base
LCS	Lexical Conceptual Structure

LFG	Lexical Functional Grammar
LOA	Level Of Articulation
LOD	Level Of Detail
LVSR	Lexical Visual Semantic Representation
MPEG	Moving Picture Experts Group
MUD	Multi-User Domain
NLP	Natural Language Processing
NP	Noun Phrase
OBJ	OBJect
ODE	Open Dynamics Engine
PAR	Parameterized Action Representation
POS	Part Of Speech
PP	Prepositional Phrase
RRL	Rich Representation Language
SAPI	Speech Application Programming Interface
SDK	Software Development Kit
SMIL	Synchronized Multimedia Integration Language
SML	Speech Markup Language
SNHC	Synthetic/Natural Hybrid Coding
TTS	Text-To-Speech
UI	User Interface
VP	Verb Phrase
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
VHML	Virtual Human Markup Language
WSD	Word Sense Disambiguation
XML	eXtensible Markup Language

Note on access to contents

I hereby declare that with effect from the date on which the thesis is deposited in the Library of the University of Ulster, I permit the Librarian of the University to allow the thesis to be copied in whole or in part without reference to me on the understanding that such authority applies to the provision of single copies made for study purposes or for inclusion within the stock of another library. *This restriction does not apply to the British Library Thesis Service (which is permitted to copy the thesis on demand for loan or sale under the terms of a separate agreement) nor to the copying or publication of the title and abstract of the thesis.* IT IS A CONDITION OF USE OF THIS THESIS THAT ANYONE WHO CONSULTS IT MUST RECOGNISE THAT THE COPYRIGHT RESTS WITH THE AUTHOR AND THAT NO QUOTATION FROM THE THESIS AND NO INFORMATION DERIVED FROM IT MAY BE PUBLISHED UNLESS THE SOURCE IS PROPERLY ACKNOWLEDGED.

Chapter 1

Introduction

Before the days of widespread books and printing, storytellers would conjure up visions of events and places, providing their listeners with an impression of realities in time and space. Theatre, fine art, animation, and cinema have added to the richness of the explicit visual experience available to the viewer. Over the last two decades, researchers in the Natural Language Processing (NLP) community and the computer graphics community have been developing techniques to enable computers to understand human natural language and to aid artists to create virtual reality for storytelling. Traditionally, NLP systems use knowledge bases containing linguistic information to analyse sentences and to produce data structures representing their syntactic structure and semantic dependency. In the computer games and animation industry, computer artists create virtual characters, props, and whole scenes of stories. The construction of these story scenes is labour intensive, time consuming, and hardly reusable.

In trying to get a machine to automatically understand our natural language and to generate virtual reality to tell stories, we can look at the way people do it. This leads to the other motivation behind the practical consideration of integrating NLP and animation generation: we attempt to solve the question “How patterns of perception are interpreted in human beings’ brains?” through the development of a virtual storytelling system. This question is a perpetual topic in the disciplines of analytical philosophy, experimental psychology, cognitive science, and more recently, artificial intelligence. Early in ancient Greece, the important relation between language and mental imagery had been noticed by classical philosophers. Aristotle gave mental imagery a central role in cognition. He asserted that “The soul never thinks without a mental image” (Thomas 1999), and maintains that the representational power of language is derived from imagery, spoken words being the symbols of the inner images. In effect, for Aristotle images play something very like the role played by the more generic notion of “mental representation” in modern cognitive science. This was almost universally accepted in the philosophical tradition, up until the 20th century. The analytical philosophy movement, which arose in the early 20th century, and still deeply influences most English speaking philosophers, originated from the hope that philosophical problems could be definitively solved through the analysis of language, using the newly invented tools of formal logic (Thomas 1999). It thus treated language as the fundamental medium of thought, and argued strongly against the traditional view that linguistic meaning derives from images in the mind.

This research contributes to the study of mental images and their relevance to the understanding of natural language and cognition, and is an attempt to simulate human perception of natural language. It integrates and improves state-of-the-art theories and techniques in the areas of NLP, intelligent multimedia presentation and 3D animation. We develop a storytelling system to automatically create imagery (virtual reality) by presenting natural language as animation and other concomitant modalities.

Figure 1.1 depicts the model of the processes of cognition, communication and re-cognition. Language visualisation is a simulation of the re-cognition process (language understanding), i.e. it extracts information in language and constructs the virtual world in mental space.

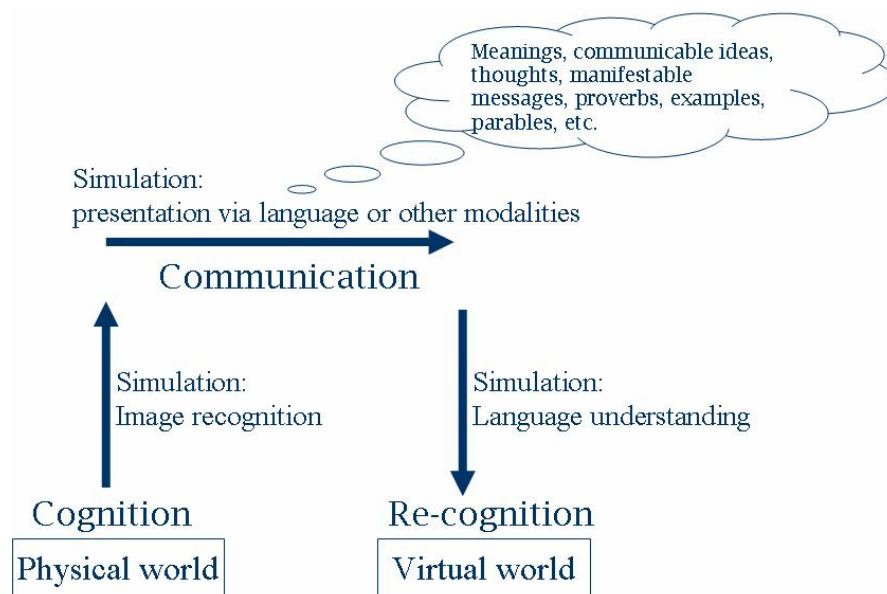


Figure 1.1: Simulation of cognition, communication, and re-cognition

1.1 Overview of language visualisation

The motivation here comes from the domain of the integration of natural language and vision processing (Mc Kevitt 1995, 1996, Maybury 1993, 1994, Maybury and Wahlster 1998, Qvortrup 2001, and Granstrom et al. 2002). There are two directions of such integration. One is to generate natural language descriptions from computer vision input. This requires integration of image recognition, cognition, and natural language generation. The other is to visualise natural language (either spoken or typed-in). The latest progress in the latter area reaches the stage of automatic generation of static images and iconic animations. Language visualisation systems usually have three central features: multimodality, animation, and intelligence.

1.1.1 Multimodal output

Pictures (and animations) often describe objects or physical actions more clearly than language does. In contrast, language often conveys information about abstract objects, properties, and

relations more effectively than pictures can. Using these modalities together they can complement and reinforce each other to enable more effective communication than can either modality alone. In this sense, multimedia storytelling systems may present stories more effectively than oral storytelling and newspaper strip cartoons.

The elements of an automatic storytelling system inspired by performance arts should correspond to those in conventional theatre arts — *Aristotle's six parts of a Tragedy* (Wilson and Goldfarb 2000). Table 1.1 shows the corresponding relationships among theatre art, a storytelling system, and its output modalities, which ensures its potential applications in automatic play/cinema direction.

<i>Aristotle's six parts of a Tragedy</i>	<i>Elements of a storytelling system</i>	<i>Output modalities of a storytelling system</i>
1. Plot	Story	Animation
2. Character	Virtual character	
3. Theme (idea)	Story	
4. Diction (Language)	Dialogue and narrative	Speech
5. Music (sound)	Nonspeech audio	Nonspeech audio
6. Spectacle	User/story listener	/

Table 1.1: Relating a storytelling system to Aristotle's six parts of a Tragedy

1.1.2 Animation

If a picture tells a thousand words then animation tells a million. 3D animations are one of the best ways to tell stories. The processes in conventional manual animation shed light on automatic animation generation. The most successful animation is probably Disney's movies. Usually, they are made by an animation generation group to create the graphics with the aid of graphics software. Although most of the graphics processing tasks are performed by computer, creating characters and animations is still a difficult and time-consuming task. A language-to-animation system that can generate animations dynamically from human natural language will spare much labour on animation direction and creation.

Most text-to-graphic conversion systems like Spoken Image/SONAS (Ó Nualláin and Smith 1994, Kelleher et al. 2000) and WordsEye (Coyne and Sproat 2001) have been able to represent text information by static pictures, animated icons, or low-quality 3D animations, e.g. simplified polygon mesh. In the area of computer graphics much work has been done on polygon-based 3D modeling, morphing, and more realistic animations, and little attention has been paid to automatic animation generation. Therefore, recent research on virtual human animation such as (Badler 1997, Esmerado et al. 2002, Lemoine et al. 2003, and Gutierrez et al. 2004) achieves high quality simulations of human motions, while little consideration is given to automation in terms of language animation. Embodied interface agents (Cassell et al. 2000) that emulate lip movements, facial expressions and body poses are restricted in conversational

movements. There are few systems which can convert English text into 3D animation. The use of animated characteristics would enable movie makers and drama directors to preview stories by watching the animated effects of actors (protagonists) with props in the scene.

1.1.3 Intelligence

As the need for high flexibility of presentation grows, the traditional manual authoring of presentations becomes less feasible. The development of mechanisms for automated generation of multimedia presentations has become a shared goal across many disciplines. To ensure that the generated presentations are understandable and effective, these mechanisms need to be *intelligent* in the sense that they are able to design appropriate presentations based on *presentation* knowledge (both visual and audio) and domain knowledge. The *intelligence* of language visualisation is embodied in the automatic generation of animation with minimal user intervention at the animation generation stage to add new objects and animations if required. For example, an animation engine could enable users to load new keyframe animations or animation specifications for dynamic animation generation, if the animation described in the input sentence is not available in the animation library. The animation engine then generates the scene, instantiates the animation to a character, and synthesizes his/her speech to present events in the sentence.

1.2 Problems in language visualisation

The research challenges of visualising natural language using virtual reality have several dimensions. They involve many problems not only in NLP and computer graphics but also in the connection between these two modalities. Many problems in NLP involve disambiguation, especially word sense disambiguation. For instance, the word “line” can be a telephone connection, a queue, a line in mathematics, or an airline. It is requisite for a language visualisation system to disambiguate different meanings and hence to interpret natural language. There are various computer graphics problems in human-like character animation, such as object manipulation, motion control in different virtual environments, and face animation and lip synchronisation, in object layout, and in automatic camera placement and control.

In terms of requirements for linking language and visual modalities, at least two issues need to be addressed. First, a data structure for expressing events, apart from expressing sentences, is required. The data structure needs to be able to express roles involved in the event, features of each role, spatial information (including path or place), etc. for visualising language. Second, a language ontology based on visual semantics for the categorisation mainly of verbs is also needed for visualisation.

1.3 Objectives of this research

The main purpose of this research is to investigate the process of language visualisation from a computational perspective, employing techniques from NLP and computer graphics. In order to achieve this, we developed a storytelling system to present natural language text using temporal media (e.g. animation, speech, and nonspeech audio). The primary objectives of this research are summarized below:

- To interpret natural language text input and to extract semantics from the input
- To generate virtual worlds automatically, with 3D animation, speech and nonspeech audio
- To integrate the above components to form an intelligent multimedia storytelling system called CONFUCIUS for presenting multimodal stories
- To evaluate CONFUCIUS and compare it to other work in the field

The objectives of this research meet several challenging problems in language animation: (1) mapping language primitives with visual primitives to present objects, actions/events, and properties, (2) visualisation of a story in natural language requires a large knowledge base of “common senses”, which requires a media-dependent intermediate semantic representation to link visual semantics with linguistic semantics, (3) representing stories by temporal multimedia (speech, non-speech audio, and animations) requires high coordination to integrate them in a consistent and coherent manner, and (4) mapping language to vision includes sophisticated spatial relations between spatial cognition and prepositions in English.

As illustrated in Figure 1.2, CONFUCIUS uses natural language text input to generate 3D animation, speech (dialogue/monologue), and non-speech audio outputs. The stories are presented by a presentation agent, Merlin the narrator. It gives the audience a richer perception than the usual linguistic narrative such as books and traditional storytellers. Since all the output media are temporal, CONFUCIUS requires coordination and synchronisation among these output modalities.

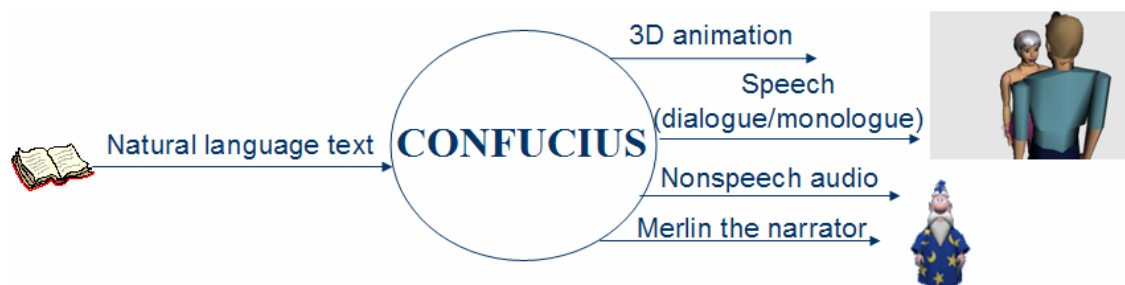


Figure 1.2: Multimodal I/O of CONFUCIUS

1.4 Outline of this thesis

This thesis is organised into nine chapters, where chapter 2 reviews approaches to multimodal processing, including virtual human standards, multimedia allocation, and nonspeech audio. Previous automatic text-to-graphics systems, embodied agents and virtual humans, and

multimodal storytelling systems are described and compared. Related work of mental imagery in cognitive science is also discussed. Chapter 3 discusses previous work in the areas of language and multimodal semantic representations, temporal relations, computational lexicons, and ontological categories of nouns and verbs.

Lexical Visual Semantic Representation (LVSR) is proposed in Chapter 4, which is capable of connecting meanings across language and visual modalities. We also investigate challenges which LVSR may encounter when certain linguistic phenomena are involved.

In Chapter 5, we study relationships between concepts and multiple modalities and propose the language ontology based on visual and audio semantics. A language ontology based on visual and audio semantics is proposed. In particular, we introduce the notion of *visual valency* and Level-Of-Detail for language visualisation. We use an automatic word sense disambiguation approach for mapping verbs to Lexical Conceptual Structure (LCS) entries using frequency information of WordNet senses, thematic grids and lexical-semantic representations from the LCS database (Dorr and Olsen 1997). This considerably improves the precision of verb sense disambiguation.

Chapter 6 discusses various issues of automatic 3D animation generation, especially virtual human animations. An object-oriented method is used to organize visual/auditory knowledge. The 3D object models encapsulate not only their intrinsic visual properties, nonspeech auditory information, but also manipulation hand postures, describing possible interactions with virtual humans. This method decentralizes animation/audio control by storing information of object interaction and sound effects in the objects. Object-specific computation is removed from the main animation control. Additionally, we combine precreated and dynamically generated (procedural) animation facilities into a unified mechanism, and blend simultaneous animations using multiple animation channels. This approach enables the intelligent storytelling to take advantage of procedural animation effects in the same manner as regular animations, adding an additional level of flexibility and control when animating virtual humans.

Next, Chapter 7 describes a virtual storytelling system, CONFUCIUS, which automatically converts natural language text to 3D animation. Its architecture and implementation issues of media allocation, knowledge base, NLP, 3D animation generation, and the story narrator are explained. The main NLP problems addressed are action representation, and applying lexical knowledge to semantic analysis. Object modelling, human animation, collision detection, and application of narrative montage in virtual environments are also discussed.

This is followed by the evaluation of the animation and NLP modules in CONFUCIUS in Chapter 8, using both subjective and objective methods to identify their adequacy and limitations. An evaluation experiment for generated animations, a test-suite based diagnostic

evaluation for syntactic parsing, and corpus-based adequacy evaluations for anaphora resolution and semantic analysis are conducted. They produce favourable results though CONFUCIUS has limited 3D models and pre-created animations. Finally, Chapter 9 concludes with a comparison with related work, and discusses areas for future research.

Chapter 2

Approaches to Multimodal Processing

Here, we investigate the elements of intelligent storytelling that we believe make traditional storytelling (e.g. literature, drama, film, animation) powerful: *characters* and *presentation*. Our research focuses on how to interpret natural language and create believable *characters* and make realistic *presentations* to tell stories. Toward this goal we explore previous work in: automatic text-to-graphics systems, multimodal storytelling (for *presentation*), virtual humans and their standards (for *characters*), and non-speech audio, and compare previous systems in this chapter. We also investigate the topic of mental imagery from the field of cognitive science.

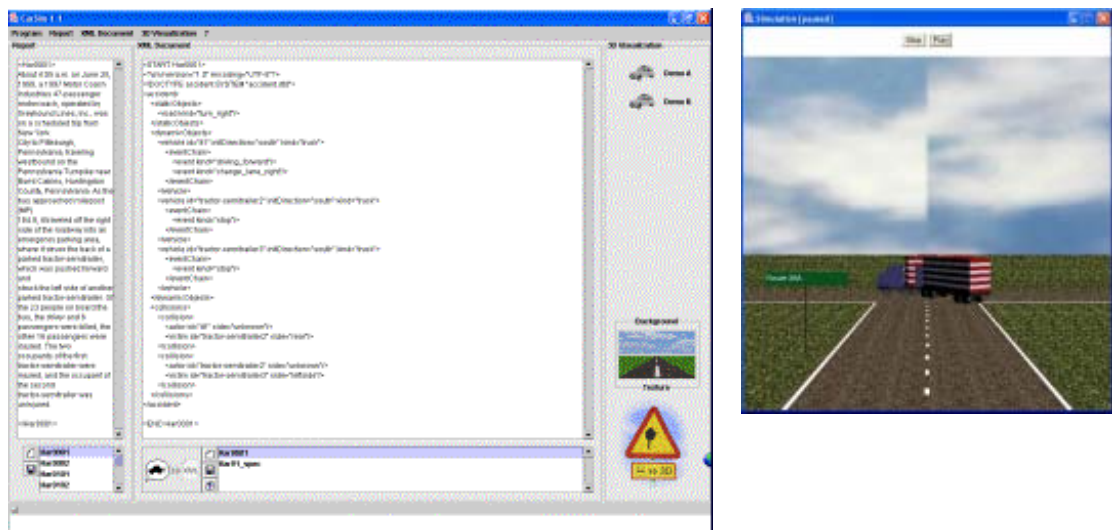
2.1 Automatic text-to-graphics systems

Visual modalities are one of the most important modalities in any multimedia presentation. As 3D computer graphics hardware and software grow in power and popularity, potential users are increasingly confronted with the daunting task of using them effectively. Making the decisions that result in effective graphics requires expertise in visual design with significant effort and time, all of which are indispensable for traditional 3D graphic authoring. However, effort and time could be spared by using automated knowledge-based design of 3D graphics and virtual worlds. Progress has been made in visualisations of abstract data (Bishop and Tipping 1998), whilst little has been done in language visualisation which connects the visual modality with another important modality in multimedia presentation — language.

In automatic text-to-graphics systems a natural language sentence is parsed and semantically interpreted, resulting in pictures depicting the information in the sentence. A graphical scene can be generated from a linguistic description as in CarSim (Dupuy et al. 2001), WordsEye (Coyne and Sproat 2001) and the Spoken Image system (Ó Nualláin and Smith 1994, Kelleher et al. 2000). In Nenov and Dyer (1988), a linguistic description of objects is visualised as a sequence of graphical pictures and vice versa. The key issues that researchers face are understanding spatial relationships by correctly interpreting prepositional phrases in language, extracting semantics of natural language and representing it in multiple modalities, in particular, dynamic visual modality. Since things (objects) have a maximum dimensionality of three which requires 3D graphic representation like WordsEye, while events have a maximum dimensionality of only one, to wit: the time, which requires animated graphic representation like Narayanan et al.'s (1995) CD-based language animation system. This section discusses recent progress in aforementioned automatic text-to-graphics (language visualisation) systems.

2.1.1 CarSim

CarSim (Dupuy et al. 2001) is an automatic text-to-scene conversion system that visualises car accidents from written reports of motor vehicle accidents. It understands the accident conditions by automatically extracting pieces of information from texts and presenting visually the settings and the movements of the vehicles in 3D scenes. CarSim has been applied to a corpus of French and Swedish texts for which it can currently synthesise visually 35 percent of the texts. CarSim is currently being ported to English. Figure 2.1A shows the graphical user interface, which visualises how information is processed in a more understandable way for the analyst. The text is displayed in the left pane, and the right pane contains XML code of the extracted information. Figure 2.1B displays the 3D scene created from the XML code in A.



A. CarSim GUI

B. 3D scene

Figure 2.1: CarSim GUI and created 3D scene

CarSim represents accidents by applying information extraction techniques to input texts, which reduce the text content to formalized templates that contain road names, road configuration, number of vehicles, and sequence of movements of the vehicles involved. The visualiser reproduces approximately 60% of manually created templates. CarSim's NLP module combines regular expression matching with dependency parsing to carry out the linguistic analysis of the texts. A regular expression grammar is used to identify proper nouns. CarSim focuses on collision verbs which are vital in the domain of motor vehicle accidents. The visualisation module recreates the 3D scene and animates the vehicles. It represents both the entities and the motions symbolically, without taking into account physics laws in the real world.

2.1.2 WordsEye

WordsEye (Coyne and Sproat 2001) can convert text into representative 3D scenes automatically. It relies on a large library of 3D models and poses to depict entities and actions.

Every 3D model can have associated shape displacements, spatial tags, and functional properties to be used in the depiction process. WordsEye generates static scenes rather than animation. Hence it focuses on the issues of semantics and graphical representation without addressing all the problems inherent in automatically generating animation. Figure 2.2 shows a picture generated from the input:

The Broadway Boogie Woogie vase is on the Richard Sproat coffee table. The table is in front of the brick wall. The van Gogh picture is on the wall. The Matisse sofa is next to the table. Mary is sitting on the sofa. She is playing the violin. She is wearing a straw hat.



Figure 2.2: An example of a WordsEye generated picture

WordsEye works by first tagging and parsing the input text, using Church's (1988) part of speech tagger and Collins' (1999) parser. The parser output is then converted to a dependency structure. Lexical semantic rules are applied to the dependency structure to derive the components of the *scene description*. For instance, the verb *throw* invokes a semantic rule that constructs a scene component representing an action (ultimately mapped to a pose) where the left hand noun phrase dependent represents an actor, the right hand noun phrase dependent a patient, and some dependent prepositional phrases the path of the patient. The depiction module of WordsEye interprets the scene description to produce a set of low-level 3D depictors representing objects, poses, spatial relations, and other attributes. Transduction rules are applied to resolve conflicts and add implicit constraints. Finally, the resulting depictors are used to manipulate the 3D objects that constitute the renderable scene. WordsEye also performs reference resolution, which is obviously crucial for deciding whether a just-named object or a pronoun is the same as an object previously named in the discourse.

WordsEye uses frames to represent verb semantics and to construct its dependency structure. Figure 2.3 shows the semantic entry for the verb *say*. It contains a set of verb frames, each of them defines the argument structure of one sense of the verb *say*. For example, the first verb frame, named `SAY-BELIEVE-THAT-S-FRAME`, has as required arguments a subject and a that-clause object, such as “John said that the cat was on the table”. Optional arguments include `ACTIONLOCATION` (e.g. “John said in the bathroom that ...”) and `ACTIONTIME` (e.g. “John said yesterday that ...”). Each of these argument specifications causes a function to be invoked to check the dependencies of the verb for a dependent with a given property, and assigns such a dependent to a particular slot in the semantic representation fragment.

```
(SEMANTICS :GENUS say
  :VERB-FRAMES
  ( (VERB-FRAME
    :NAME SAY-BELIEVE-THAT-S-FRAME
    :REQUIRED (SUBJECT THAT-S-OBJECT)
    :OPTIONAL (ACTIONLOCATION ACTIONTIME) )
    (VERB-FRAME
    :NAME SAY-BELIEVE-S-FRAME
    :REQUIRED (SUBJECT S-OBJECT)
    :OPTIONAL (ACTIONLOCATION ACTIONTIME) )
    . . . ) )
```

Figure 2.3: Verb frames of *say* in WordsEye

At the core of WordsEye is the notion of a *pose*, which can be loosely defined as a figure (e.g. a human figure) in a configuration suggestive of a particular action. For example, a human figure holding an object in its hand in a throwing position would be a pose that suggests actions such as *throw* or *toss*. Substituting for the figure or the object allows one to depict different statements, such as “John threw the egg” or “Mary tossed the small toy car”.

Objects’ spatial tags depend on their shape and aid in visualisation of spatial relations. WordsEye specifies different types of spatial tags such as *canopy area*, *top surface*, *base*, *cup*, *wall*, *cap*, *enclosed-area*, *ridge*, and *peak*. The longest axis is a vector indicating the object’s longest axis and pointing to the end of the object. It is usable for prepositions such as “end” and “along” when indicating spatial relations with objects like “road”, “river”, and “way” (in e.g. “at the end of the river”, “along the road”).

WordsEye can translate information expressed in language into a graphic representation. But when the semantic intent is ambiguous or beyond the system’s common-sense knowledge, the resulting scene might loosely match what is expected. An important area of recent research not covered by WordsEye is coordinating temporal media, e.g. speech and animation, where information is presented over time and needs to be synchronised with other media.

2.1.3 Micons and CD-based language animation

Moving icons (animated icons) are simple gif animations with a little motion to spice up web pages or operating systems' GUI, e.g. a book pages turn, a letter flies to a mail box. The term Micons (moving icons) was first coined by Sassnet (1986), and then Baecker (Baecker et al. 1991) made some initial steps in language animation with the idea of "micons". He used a set of atomic micons to describe a set of primitives (objects and events) and developed a general purpose graphical language, CD-Icon, based on Schank's (1972) Conceptual Dependency (CD). CD-Icon indicated some major limitations of methods closely based on CD theory: they work well for representing physical things but have difficulty in representing abstract concepts and are restricted in closed sets (e.g. primitive actions), and complex messages can only be constructed by physical relations such as space, time and causality.

Narayanan et al. (1995) discuss the possibility of developing visual primitives for language primitives where CD is used. A 3D dynamic visualisation system (language animation) is developed to represent story *scripts* continuously. It maps language primitives onto visual primitives and animation sequences and achieves maximal continuity by animation. Using this system an example story 'Going to a restaurant' is provided. Figure 2.4 shows the process of "the waiter moves towards John and hands a menu to John. John scans it, decides what to eat and tells the waiter. The waiter then informs the chef". The animation shown in Figure 2.4 comes from the following script:

```
Waiter PTRANS Waiter to table
Waiter ATRANS menu to John
John MTRANS menu to John
John MBUILD choice
John MTRANS choice to Waiter
Waiter PTRANS Waiter to Chef
Waiter MTRANS choice to Chef
```

The representation of actors and objects in this system is iconic, and no image details are given. A person is symbolized by a cone with a ball on the top and differentiated by different colors, while the restaurant is just a rectangular cube. By changing the micons' position, color, and shape, actions performed and changes of objects' state are presented. Hence it may be also regarded as a micon system and has the limitations discussed above.

2.1.4 Spoken Image and SONAS

In Ó Nualláin and Smith's (1994) Spoken Image (SI), a 3D dynamic graphic visualisation is displayed, giving verbal scene descriptions spoken by a human user. The output graphics can be incrementally reconstructed and modified as the user gives more detailed descriptions or mentions new objects in the scene. A user's description of an urban street environment and the corresponding visualisation are shown in Figure 2.5.

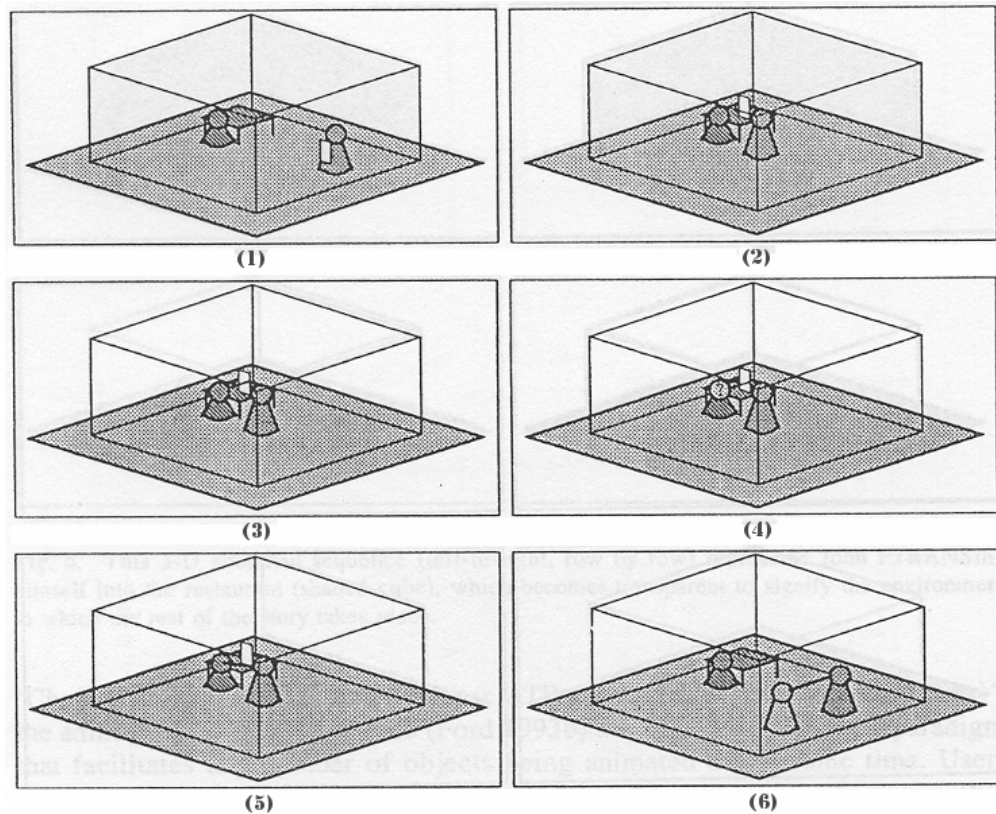


Figure 2.4: An example story: going to a restaurant (Narayanan et al. 1995)

(The screen is initially black, and the system waits for the user to begin.)

User: "You are standing on a suburban street corner."

(Some typical suburban houses with lawns, sidewalks along the edge of the street, and trees etc. appear on the screen.)

User: "The house on the corner has a red door, and green trim around the windows."

(Scene adjusts to fit the new descriptive detail.)

User: "Walk down the street to your left, which is Rowan Crescent."

(A street sign appears, with the new name on it, and the scene changes to reflect movement of the observer.)

Figure 2.5: An example of human-computer interaction in SI

The SONAS system (Kelleher et al. 2000), the successor to Spoken Image, is an intelligent multimedia multi-user system that uses a synergistic combination of several input modalities such as spoken natural language and gesture. The environment is a 3D model of a town. The user can navigate and interact with the environment through multiple modalities. One goal of the system is the manipulation of objects in a 3D environment using natural language. For example, in the instruction "Move the tree in front of the house", the user should see the tree moving in front of the house. To achieve this kind of task, firstly the sentence must be parsed and broken down into the figure "the tree", the reference object "the house", the action "move", and the spatial relation "in front of", then SONAS searches the visual model for the figure and reference objects. Once they have been identified, an instance of the appropriate telic action class is instantiated. Kelleher et al. develop a motion class set to deal with telic actions

(Figure 2.6). Each class has a function that takes a geometrical conceptualisation of the action figure (`Geo-Concept`), the initial position (`InitPoint`) and the final position (`FinPoint`) of the figure as parameters and returns an array of points (`Point[]`) representing the path that the figure must take to mimic the action. Each telic action verb inherits from one of these motion classes and uses these functions to calculate the transform applied to the figure. Both the action of `StackMotion` and `SlideMotion` inherit from the `Motion` class in Figure 2.6.

SI and SONAS lack knowledge of naïve physics such as collision detection, e.g. any physical object may not pass through another one. In SI/SONAS the viewer could be in or pass through a non-hollow object when he is navigating the world, and a newly-added object could be placed coincidentally in a position where it intersects with other existing objects.

```
Motion
-----
Point[] MotFunc(InitPoint, FinPoint, Geo-Concept, ...);
```

A. General form of the Motion class

```
StackMotion
-----
Point[] StackFunc(InitPoint, FinPoint, Point);

SlideMotion
-----
Point[] SlideFunc(InitPoint, FinPoint, Point, Surface);
```

B. Examples of Motion classes in the hierarchy

Figure 2.6: Motion classes in SONAS

2.2 Embodied agents and virtual humans

Embodied animated agents, either based on real video, cartoon style drawings or 3D models, are likely to become integral parts of multimodal interfaces where the modalities are the natural modalities of face-to-face communication among humans, i.e. speech, facial expressions, hand gestures, and body stance. Since character is one of the most essential elements in storytelling, creating believable and realistic characters is the crucial task of impressive storytelling. It is practical to turn agents into actors for storytelling because agents' looks, the way they move and how they express their intentions are similar to those of characters in a story, though we have to take the step from generating descriptions of possible behaviours in possible worlds to expressing behaviours in a chosen material in a certain environment (i.e. the story world).

In this section current virtual human standards are investigated and classified into four groups according to the levels of abstraction, and various virtual human animation applications such as Jack (Badler 1997) and Improv (Perlin and Goldberg 1996) are discussed. Cartoon style interface agents like Gandalf (Thórisson 1996), Disney animation, and model-based 3D agents like *REA* (Cassell et al. 2000) are also discussed. Some of them focus on the agents' behaviour

model and personality (REA and Disney's characters), others focus on accurate simulation of human motions such as Jack or multimodal human-computer communication as in Gandalf.

2.2.1 Virtual human standards

3D graphics web applications such as online games, virtual environments, and intelligent agents, are more and more demanding 3D graphics modelling languages that represent not only virtual objects but virtual humans and their animation. Existing virtual humans and animation on the Web are created by various authoring tools (e.g. 3D Studio Max, Maya, Poser, and motion capture devices) and in different formats. There are a wide range of languages and technologies for behaviour and animation of virtual humans. Current 3D human standards (e.g. VRML, MPEG-4) aim for various levels of abstraction, especially for lower level representations.

We investigated current virtual human representation languages and found that they can be classified to four groups according to the levels of abstraction, starting from 3D geometry modelling to language animation. Figure 2.7 illustrates these four levels of virtual human representation. Most work on virtual human modelling and animation focuses on the lower levels.

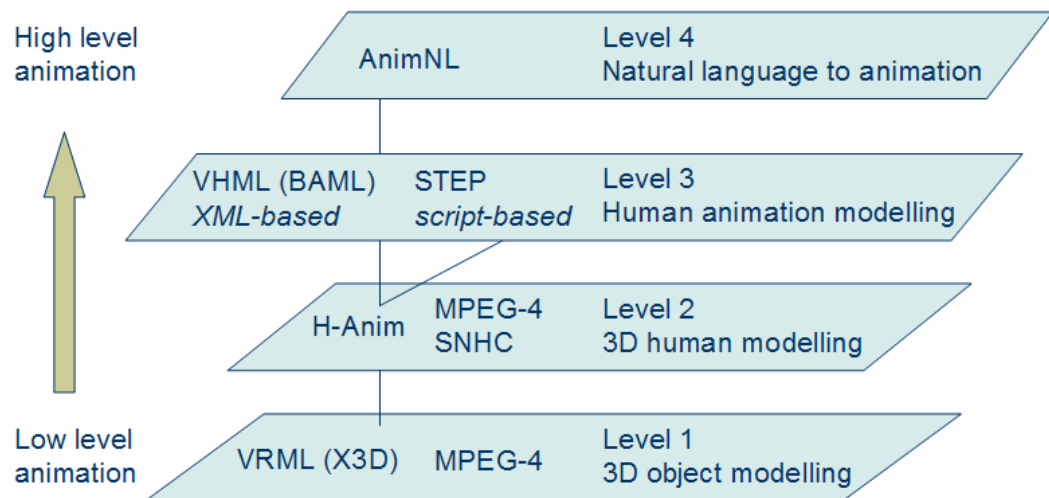


Figure 2.7: Four levels of virtual human representation

VRML (X3D) and MPEG-4 for Object Modelling

The first level is for 3D object modelling. VRML (X3D) and MPEG-4 are two leading standards of 3D content for Web applications. VRML (Virtual Reality Modelling Language), developed by the Web3D Consortium (originally the VRML Consortium), is a hierarchical scene description language that defines the geometry and behaviour of a 3D scene and the way in which it is navigated by the user. X3D is the successor to VRML. It extends VRML with new features, advanced APIs, additional data encoding formats (VRML97 and XML), and a component-based architecture that permits a modular approach. VRML (X3D) is the standard

used most widely on the Internet to describe 3D objects/humans and users' interaction with them.

Unlike VRML, MPEG-4 uses BIFS (Binary Format for Scenes) for real-time streaming, i.e. a scene does not need to be downloaded in full before it can be played, but can be built up on the fly. BIFS borrows many concepts from VRML. BIFS and VRML can be seen as different representations of the same data. In VRML, the objects and their actions are described in text, but BIFS code is binary, and thus is shorter for the same content, typically 10 to 15 times.

H-Anim and MPEG-4 SNHC for Humanoid Modelling

The second level is for 3D human modelling. H-Anim (2001) is a standard VRML97 representation for humanoids. It defines standard human Joints articulation (e.g. knee and ankle), Segments dimensions (e.g. thigh, calf, and foot), and Sites (e.g. hand_tip, foot_tip) for "end effector" in IK and attachment points for clothing. Each joint node may contain other joint nodes and a segment node that describes the body part associated with the joint. Each segment is a normal VRML transform node describing the body part's geometry and texture. H-Anim humanoids can be animated using keyframing, inverse kinematics (IK), and other animation techniques. Appendix E lists H-Anim models available on the Internet.

MPEG-4 SNHC (Synthetic/Natural Hybrid Coding) is concerned with the compression of media streams, such as geometry, animation parameters, or text-to-speech, beyond traditional audio and video, the representation and coding of synthetic objects as well as their natural audiovisual counterparts, and the spatial-temporal composition of these natural and synthetic objects. MPEG-4 SNHC offers an appropriate framework for 3D virtual human animation, gesture synthesis and efficient compression/transmission of these animations.

SNHC incorporates H-Anim and provides an efficient way to animate virtual human bodies and tools for the efficient compression of the animation parameters associated with the H-Anim articulated human model. It defines two sets of parameters: body definition and animation parameters. Body definition includes Face Deformation Parameters (FDPs) and body deformation parameters (BDPs). These let the decoder specify shape and texture of a model. Animation parameters include Face Animation Parameters (FAPs) and body animation parameters (BAPs). BAPs are a set of rotation angles of body parts to specify posture. MPEG4 defines 14 visemes and 6 expressions represented by low level FAPs, which are represented as a set of feature points on the face. Table 2.1 and 2.2 list H-Anim suggested displacer nodes which are taken from MPEG4 FAPs. Each FAP is controlled by a specific muscle (e.g. eyes, lips, jaw, brows). Visemes usually concern lips and jaw movement, and expressions concern lips, eyes and eyebrows.

VHML and STEP for Human Animation Modelling

The third level is for human animation modelling. Following the lead of W3C's Synchronised Multimedia Integration Language (SMIL 2005), the Virtual Human Mark-up Language

(VHML) community develops a suite of XML-based language for expressing humanoid behaviour, including facial animation, body animation, speech, emotional representation, and multimedia. The H-Anim specification describes the geometry and structure of a virtual human, however it doesn't provide a way to specify animation. VHML provides an intuitive way to define virtual human animation. It is composed of several sub-languages: DMML (Dialogue Manager Markup Language), FAML (Facial Animation Markup Language), BAML (Body Animation Markup Language), SML (Speech Markup Language), and EML (Emotion Markup Language). Figure 2.8A shows a VHML example. With this human animation language it is easy to specify generic animations for virtual humans in a wide variety of applications. Like other XML-based markup languages, VHML is declarative and requires a Java or other XML consumer to interpret XML-based markup with H-Anim or MPEG-4 formats.

<i>Viseme</i>	<i>Displacer Name</i>	<i>Phonemes</i>	<i>Example</i>
1	viseme_pbm	p, b, m	<u>put</u> , <u>bed</u> , <u>mill</u>
2	viseme_fv	f, v	<u>far</u> , <u>voice</u>
3	viseme_th	T,D	<u>think</u> , <u>that</u>
4	viseme_td	t, d	<u>tip</u> , <u>doll</u>
5	viseme_kg	k, g	<u>call</u> , <u>gas</u>
6	viseme_ts	tS, dZ, S	<u>chair</u> , <u>join</u> , <u>she</u>
7	viseme_sz	s, z	<u>sir</u> , <u>zeal</u>
8	viseme_nl	n, l	<u>lot</u> , <u>not</u>
9	viseme_r	r	<u>red</u>
10	viseme_a	A:	<u>car</u>
11	viseme_e	e	<u>bed</u>
12	viseme_i	I	<u>tip</u>
13	viseme_q	Q	<u>top</u>
14	viseme_u	U	<u>book</u>

Table 2.1: MPEG4 visemes

#	<i>Expression name</i>	<i>Textual description</i>
1	joy	The eyebrows are relaxed. The mouth is open and the mouth corners pulled back toward the ears.
2	sadness	The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.
3	anger	The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or opened to expose the teeth.
4	fear	The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.
5	Disgust	The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.
6	surprise	The eyebrows are raised. The upper eyelids are wide open, the lower relaxed. The jaw is opened.

Table 2.2: MPEG4 expressions

There are several other languages on the third level. Based on H-Anim, STEP (Huang et al. 2003) is a script-based language for human actions. It allows for a precise definition of a complex repertoire of gestures or postures. STEP shares a number of interests with the VHML community. It has a Prolog-like syntax, which makes it compatible with most standard logic programming languages. The formal semantics of STEP is based on dynamic logic. Figure 2.8B shows how to use STEP to define “walk”.

```
<left-calf-flex amount="medium">
<right-calf-flex amount="medium">
  <left-arm-front amount="medium">
  <right-arm-front amount="medium">
    Standing on my knees I beg you pardon
  </right-arm-front></left-arm-front>
</right-calf-flex></left-calf-flex>
```

A. A VHML example

```
script(walk_pose(Agent), ActionList):-
  ActionList =
    [parallel([
      turn(Agent,r_shoulder,back_down2,fast),
      turn(Agent,r_hip,front_down2,fast),
      turn(Agent,l_shoulder,front_down2,fast),
      turn(Agent,l_hip,back_down2,fast)]),
  parallel([turn(Agent,l_shoulder,back_down2,fast),
    turn(Agent,l_hip,front_down2,fast),
    turn(Agent,r_shoulder,front_down2,fast),
    turn(Agent,r_hip,back_down2,fast)])].

Script(walk_forward_step(Agent),ActionList):-
  ActionList=[parallel(
    [script_action(walk_pose(Agent),
      move(Agent,front,fast)])].
```

B. A STEP example

Figure 2.8: Representing parallel temporal relation

RRL (Rich Representation Language) (Piwek et al. 2002) is an SMIL influenced markup language used in the NECA system, which generates interactions between two or more animated characters. RRL focuses on representations of agent behaviour in dialogue and supports the integrated representation of various types of information, even including some linguistic information (e.g. pragmatic, semantic, syntactic, and prosodic). AML (Avatar Markup Language) (Kshirsagar et al. 2002) is a VRML influenced markup language for describing

avatar animation. It encapsulates Text To Speech content, Facial Animation and Body Animation in a unified manner with appropriate synchronisation information.

Natural Language to 3D Animation

The fourth level includes high level animation applications which convert natural language to virtual human animation. Little research on virtual human animation focuses on this level. One of the first projects is the AnimNL project (Webber et al. 1995) that aims to enable people to use natural language instructions, such as “go to the kitchen and get the coffee pot”, to tell virtual humans what to do. We believe that the integration of linguistic knowledge with these modelling languages can achieve higher level representation of virtual human modelling and animation and lead to powerful web-based applications.

2.2.2 Virtual humans

There have been a number of virtual human animation systems and approaches varying in function, autonomy, and levels of detail according to different application domains such as medical (Gutierrez et al. 2004), art (Esmerado et al. 2002), training and maintenance (Badler 1997), interface agents, and virtual reality (Lemoine et al. 2003). Much research has been conducted in the field of virtual human modelling, motion and behaviour. Some models such as Jack are very advanced in a narrow area (e.g. biomechanical robot simulation) but lack other desirable features such as real-time communication.

Jack

Jack (Badler 1997) is an interactive system for definition, manipulation, animation, and performance analysis of virtual human figures. Jack is applied in industries to improve the ergonomics of product designs and workplace tasks. It enables users to position biomechanically accurate digital humans of various sizes in virtual environments, assign them tasks and analyze their performance. The virtual human *Jack* can tell what it can see and reach, and can simulate how a human performs specific tasks. Jack focuses on proper task planning and biomechanical simulation and its general goal is to produce accurate simulations of biomechanical robots. Applications in industrial design determined that Jack has too much accuracy and includes far more detail on human body modelling than what is necessary for general-purpose virtual agent applications.

Improv

Improv (Perlin and Goldberg 1996) is an animation system for scripting interactive virtual characters. It allows artists to create powerful scripted scenes with virtual characters. It emphasizes agents’ believable movement. Improv produces animations of compelling movement in human figures. It uses script language to produce virtual actors’ activities and applies them to drama performance. It consists of an Animation Engine which uses procedural techniques to generate layered, continuous motions and transitions between them, and a rule-

based *Behaviour Engine* which governs how actors communicate with each other and make decisions reacting to the ever-changing environment.

2.2.3 BEAT and other interactive agents

REA (Real Estate Agent) is an animated human simulation on a screen that can understand the conversational behaviours of the human standing in front of it via computer vision techniques, and respond with automatically generated speech and face, hand gesture and body animation (Cassell et al. 2000). The system consists of a large projection screen on which REA is displayed and in front of which the user stands. Two cameras mounted on top of the projection screen track the user's head and hand positions. Users wear a microphone for capturing speech input. REA's application domain is real estate and she acts as a real estate agent showing users the features of various models of houses that appear onscreen behind her, as shown in Figure 2.9A.

REA integrates a natural language generation engine (SPUD), and an animator's tool, BEAT (Figure 2.9B), which allows animators to input typed text that they wish to be spoken by an animated human figure. In the same way as Text-to-Speech (TTS) systems realise written text in spoken language (McTear 2002) BEAT realises written text (dialogues) in embodied expressive verbal and nonverbal behaviours such as face expression, head nods, gaze, and hand gestures¹ on 3D virtual agents. And in the same way as TTS systems are permeable to trained users, allowing them to tweak intonation, pause-length and other speech parameters, BEAT is permeable to animators, allowing them to write particular gestures, define new behaviours and tweak the features of movement. It facilitates behaviours generation in dialogue-rich stories/scripts.



A. User interacting with REA



B. "You just have to type in some text."(BEAT)

Figure 2.9: REA and BEAT

¹ This is also a principle of the Disney animators for animating expressions and movements of a character to concomitant dialogue.

Sam (Cassell et al. 2000), another 3D animated conversational agent, can tell stories and share experiences together with children by sharing physical objects across real and virtual worlds. It acts as a peer playmate in a shared collaborative space by using the real-time video of the child's environment as Sam's background, so that Sam seems to exist in the child's play space.

2.2.4 Divergence on agents' behaviour production

The linkage between agents' speech and behaviours which has been explored in synthesising realistic animation of autonomous agents is a result of physiological and psychological research in the relation of dialogue, facial expressions, and gestures in human communication. There is a divergence of opinion in animating accompanying expressions and movements of agents' speech.

On the one hand, Cassell et al. (1998) create intelligent agents whose motions and expressions are generated automatically by computer programs. Their approach tends to extract information from the agents' speech and the flow of conversation. Such motions (e.g. hand gestures) and face expressions support and expand on information conveyed by words. Cassell et al. (1998, p. 583) state the following: "The fact that gestures occur at the same time as speech, and that they carry the same meaning as speech, suggests that the production of the two are intimately linked. In fact, not only are the meanings of words and of gestures intimately linked in a discourse, but so are their functions in accomplishing conversational work: it has been shown that certain kinds of gestures produced during conversation act to structure the contributions of the two participants (to signal when an utterance continues the same topic or strides out in a new direction), and to signal the contribution of particular utterances to the current discourse. ... Gesture and speech are so intimately connected that one cannot say which one is dependent on the other. Both can be claimed to arise from a single internal encoding process."

Figure 2.10 presents a fragment of dialogue in which two animated agents' gesture, head and lip movements, and their inter-synchronisation were automatically generated by a rule-based program. A is a bank teller, and B has asked A for help in obtaining \$50.

```
A: Do you have a blank check?
B: Yes, I have a blank check.
A: Do you have an account for the check?
B: Yes, I have an account for the check.
A: Does the account contain at least fifty dollars?
B: yes, the account contains eighty dollars.
A: Get the check made out to you for fifty dollars and then I can
   withdraw fifty dollars for you.
B: All right, let's get the check made out to me for fifty dollars.
```

Figure 2.10: Dialogue between two autonomous agents

The gestures and facial expressions depend on sentence type (declarative or interrogative), meaning (affirmative or negative), and stressed words (italic words in the example) of the speech. In this example, every time B replies affirmatively (“yes”) he nods his head, and raises his eyebrows. A and B look at each other when A asks a question, but at the end of each question A looks up slightly. In saying the word “check”, A sketches the outlines of a check in the air between him and his listener. In saying “account” he forms a kind of box in front of him with his hands: a metaphorical representation of a bank account in which one keeps money. When he says the phrase “withdraw fifty dollars”, he withdraws his hand towards his chest.

On the other hand, in traditional manual animation art rules are different. The Disney animators’ principle for animating expressions and movements to accompany dialogue is expressed in the following: “The expression chosen is *illustrating the thoughts* of the character and *not the words* he is saying; therefore it will remain constant no matter how many words are said. For each single thought, there is one key expression, and while it can change in intensity it will not change in feeling. When the character gets a new thought or has a realisation about something during the scene, he will change from one key expression to another, with the timing of the change reflecting what he is thinking.” (Loyall 1997, p. 23) Traditional animation artists tend to generate characters’ expressions and motions from their feeling, personality, and attitude rather than their speech. We can see the results of this principle in Disney’s cartoons.

Generating expressions and motions from a character’s thoughts instead of their speech is a challenge for synthesised animation. Gesture and expressional behaviour had been virtually absent from attempts to animate autonomous agents until Cassell et al.’s research. The approaches in both traditional and intelligent animated character behaviour production are useful to associate characters’ nonverbal behaviours with their speech and personalities.

2.2.5 Gandalf

Thórisson (1996) developed a system that addresses many issues in face-to-face communication. The agent, *Gandalf*, is rendered as a computer-animated face and associated hand. Gandalf is the interface of a blackboard architecture which includes perceptual integration of multimodal events, distributed planning and decision making, layered input analysis and motor-control with human-like characteristics and an inherent knowledge of time. People interacting with the system must wear sensors and a close microphone to enable Gandalf to sense their position, sense what they are looking at and their hand position over time, and perform speech recognition (Figure 2.11).

Using this system he has tested his theory for psychologically motivated, natural, multimodal communication using speech, eye contact, and gesture. Gandalf participates in conversations by attempting to produce all of these modalities at moments appropriate to the ongoing conversation. Because of the focus of the research, Gandalf does not attempt to express

a personality, have realistic 3D graphic representation, other body movement (besides hand gestures) outside of the conversation, or other aspects needed for autonomous agents. Gandalf uses canned text rather than performing natural language generation in answering. Nevertheless, the techniques used in Gandalf address a subset of the requirements for language use in agents, and could clearly be useful in multimodal communication with embodied agents.



A. Gandalf



B. A user prepares to interact with Gandalf

Figure 2.11: Gandalf's interface

2.2.6 Humanoid Animation

Representing humanoid kinematics is a main task in animation generation. The kinematic animation techniques vary from a simple application of precreated animation frame data (keyframes, either hand-animated or motion-captured), to a complex on-the-fly inverse kinematics. The traditional approach of animating characters (agents/avatars) provides a set of animations from which the user/system can select. In most current graphical chat rooms the user can control his avatar behaviour by selecting an animation sequence from a list of available motions. The avatar can only play one animation at a time, i.e. only apply one precreated animation for the entire duration of the animation sequence.

Improv uses procedural animation combined with behavioural scripting for creating flexible characters for virtual theatre. Improv divides the actions of avatars into a set of groups. The action, in this case, is defined as a single atomic or repetitive activity that does not require explicit higher-level awareness or conscious decisions. Actions within a group are mutually exclusive of one another; activating one causes the action currently active to end. Actions in different groups can operate simultaneously, so activities of certain parts of the body can be layered over those involving others. Using this structure, the basic set of actions can be combined to create dozens of composite animations while minimising the risk of inadvertently creating a behaviour that is either unbelievable or not lifelike. The solution serves as the mechanism for user-controlled avatars by enabling multiple levels of abstraction for the possible actions.

A common method of animation blending in the games industry is using a finite state machine for character state transitions over a timeline. Transitions must be constrained by behaviour rules, e.g. a character cannot crawl if (s)he is holding a gun. The method may result in too many rules, especially when there are a large number of animations available.

2.3 Multimodal storytelling

Rapid progress in the development of multimedia technology promises more efficient forms of human computer communication. However, multimedia presentation design is not just merging output fragments, but requires a fine-grained coordination of communication media and modalities. It requires intelligent multimedia systems to have the ability of reasoning, planning, and generation. Research in this area was initiated during the mid 1980s (Maybury 1993, 1994, Maybury and Wahlster 1998, Qvortrup 2001, Mc Kevitt 1995, 1996, and Granstrom et al. 2002).

In terms of storytelling, Schank (1995) looks closely at the way in which the stories we tell relate to our memory and our understanding. People talk about what happens to them, and they tell others what they remember. Telling stories and listening to other people's stories shape the memories we have of our experiences. Schank explores some aspects and implications of our ability to recall stories and relate them to new ones we are hearing. "Our interest in telling and hearing stories is strongly related to the nature of intelligence," Schank (1995, p. 40) observes. "In our laboratory today, we are attempting to build machines that have interesting stories to tell and procedures that enable them to tell these stories at the right time." Schank's research on CD theory and scripts forms the theoretical basis of many storytelling systems. Moreover, projects in interactive storytelling/drama integrate the progress in multimedia presentation, multimodal interfaces, and computer games. KidsRoom (Bobick et al. 1996), Larsen and Petersen's (1999) multimodal storytelling environment and the Oz project (Loyall 1997) are typical interactive multimodal storytelling systems. These systems provide an interactive way to change a story dynamically according to users' activities (including speech) during storytelling, and therefore extend users as story designers participating in the storytelling and exploring the story in immersive environments. So users play the role of both storyteller and story listener. Unlike these multimodal storytelling systems, AESOPWORLD (Okada 1996) focuses on mental activities of protagonists and is not interactive.

2.3.1 Interactive storytelling

Larsen and Petersen (1999) describe an interactive storytelling system. The story told by the system is built as a film shot from the eyes of the user (subjective camera). When the story begins the camera is placed in a forest. Bird sounds coming from the trees are heard while the camera looks around, and starts moving forward. A chewing sound becomes hearable, and the camera looks around again and spots a sheep and starts moving towards the sheep. But when the

camera comes close to the sheep, it gets scared, cries out a heartbreaking sound and starts running away from the camera. The camera then continues the journey through the virtual world. Two signs appear, and the camera approaches them. On the left sign the word “Farm” is written and on the right sign the word “Castle”. When the camera is in front of the signs, a voice is heard, “Please choose left or right”. It then waits for the user to decide which direction should be taken in the story.

The system receives multimodal input in the form of scripts to obtain the storyline from a storywriter, and speech, vision input as well to achieve user interaction, and produces multimodal output. The main input is not natural language stories but executable scripts. The scripts comprise rules that trigger events in the story, through the use of a rule-based architecture. These rules are activated in parallel while the storyline is still sequential. The script fragment shown in Figure 2.12 checks a timer first, if the condition is met, the viewpoint of the virtual observer (i.e. the user/player) is moved and rotated. The speech input of the system is just a substitution of mouse activities in usual 3D games. Since the language component is of limited scope, the speech input is restricted to simple commands such as `turn left`, `turn right`. For example, when the camera is in front of a signpost, a voice prompt is heard, “Please choose left or right.” It is up to the user to decide which direction should be taken in the story.

```
If TimeBeenInRuleset == 1000
Then  Camera.MoveTo((880,100,-9000));
      Camera.RotateToward((-100,50,-3300));
Endif;
```

Figure 2.12: Executable story script of Larsen’s storytelling system

Although not implemented in the system, Larsen and Petersen intended to use autonomous agents as actors in the story and apply *behaviour models* to their action selection. The behaviour models can give the autonomous actors “a life of their own” in the virtual world and reduce the work of story generation. Larsen and Petersen’s story presents high quality graphic output, but it is created from executable computer language and hence has the same limitation as KidsRoom, i.e. it can tell only one story specified in the scripts.

2.3.2 AESOPWORLD

Different from most storytelling systems that focus on storylines, AESOPWORLD (Okada 1996) aims at developing a human-like intelligent, emotional agent and focusses on modelling the mind. AESOPWORLD is an integrated comprehension and generation system for integration of vision, cognition, thought, emotions, motion, and language. It simulates the protagonist of the fox of an Aesop fable, *the Fox and the Grapes*. The fox has desires, and makes plans to satisfy the desires. He recognizes the real world, takes action to execute the plans, and sometimes gets emotional with events. He utters his mental states or thinking processes as monologue, and produces dialogue when he meets someone. His mental and

physical behaviours are shown by 2D graphic displays, a speech synthesiser, and a music generator which expresses his emotional states.

The character's mind model consists of nine domains according to the contents of mental activities: (1) sensor, (2) recognition-understanding, (3) planning-creation, (4) action-expression, (5) actuator, (6) desire-instinct, (7) emotion-character, (8) memory-learning, and (9) language, and five levels along the process of concept formation: (1) raw data, (2) cognitive features, (3) conceptual features, (4) simple concepts, and (5) interconnected-synthesised concepts. Two of these domains are language and image recognition coupled with vision understanding. The system generates a simulation of three of the nine domains that function in parallel in the character's mind. Language generation is based on *propositional* and *modal logic* encoded in case frames, whereby linguistic knowledge is organized around verb senses. Each sense is associated with elements from a set of cases, e.g. instrumental, locative, etc. The story is generated via a chain activation of the modules that make up the various domains. Focussing on simulating mental activities, AESOPWORLD does not produce 3D graphic displays of the story of *the Fox and the Grapes*.

2.3.3 Oz

The Oz project (Smith and Bates 1989, Loyall 1997) is aimed at constructing interactive characters and enables people to create and participate in interactive stories, and develop computational methods for varying the presentation style of the experience, thus providing the interactive analogue of film technique and writing style. Figure 2.13 shows a story world with two agents in it. The bodies of the agents are simplified to ellipsoids that can jump, move, squash, stretch and can have transformations performed on them. Every body also has two eyes that are each composed of a white and a black sphere (pupil) which can be moved within constraints to simulate eye gaze.

Primitive actions are specified by names and parameters, such as `Jump`, `Put` (moving action), `Squash`, `Spin` (changing the orientation of the body), `OpenEyes`, `CloseEyes`, `SpinEyes` (look at) and `ElevateEyes` (look up), `StartLookPoint` and `StopLook` (the eyes track objects or points), `ChangeBodyRadii` (changing size), `ChangeColor`. The agents in OZ can "speak" by issuing text strings to appear in a speech bubble above the agent's head (as shown in Figure 2.13). These strings appear at the next available position in the text bubble at the time the `Say` action is issued. Thus using these primitive actions in sequence can create meaningful behaviours such as greeting another agent, sleeping, or going to a place in the world. Behaviours and goals are grounded in primitive actions. Primitive actions are of two types, physical actions and mental actions. Physical actions are the set of actions the body is directly capable of such as `Jump` and `Spin`. Mental actions concerns mental activities such as `amuse self`. The initial goals of characters, the behaviours, the subgoals they give rise to, and their behaviours are all written by the story author, and comprise a large part of the personality

of the agent. Oz focusses on simulation of behaviour, emotion, and personality of autonomous agents in storytelling which makes characters in a story more believable. Similar to AESOPWORLD, Oz only provides low quality graphic output.

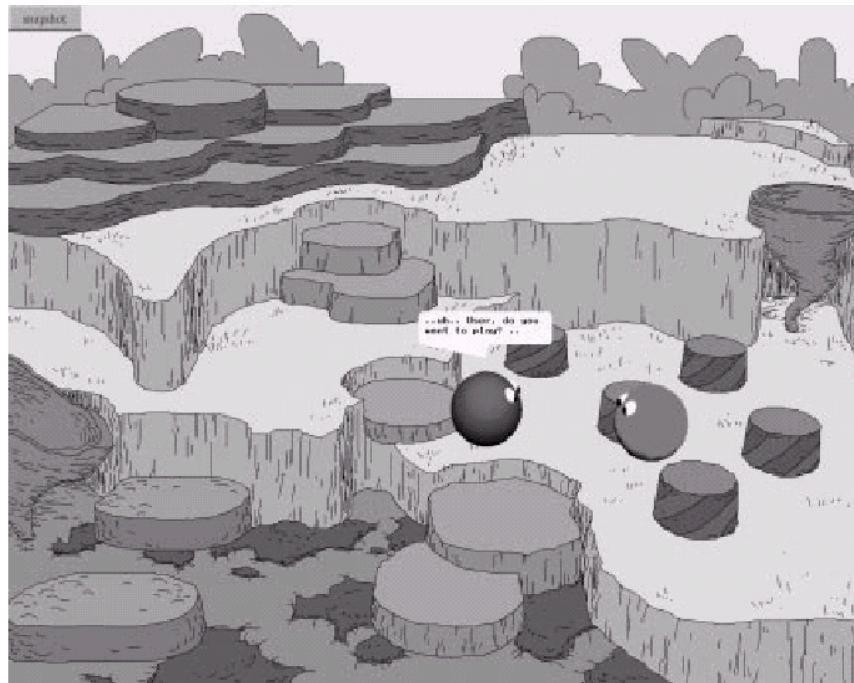


Figure 2.13: The *Edge of Intention* world in OZ

To summarise, most of the above storytelling systems focus on interaction between the user (player) and the story. The graphic presentations are high quality, but either prefabricated (KidsRoom) or from executable computer language (Larsen and Petersen's storytelling), which reduces the flexibility of the systems.

2.3.4 KidsRoom

KidsRoom (Bobick et al. 1996) combines the physical and the virtual world into an interactive narrative play space for children. Using images, lighting, sound, and computer vision action recognition technology, a child's bedroom was transformed into an immersive fantasy world. Objects in the room (e.g. furniture) become characters in an adventure, and the room itself actively participates in the story, guiding and reacting to the children's choices and actions (Figure 2.14).

KidsRoom uses three video cameras for computer vision, two digital AlphaStations for displaying animations, and four SGI R500 workstations for tracking, playing sound effects, MIDI light output, and action recognition. Children's positions and actions were tracked and recognized automatically by computer and used as input for the narration control system. Computer vision techniques were tightly coupled to the narrative, exploiting the context of the story in determining both what needed to be seen and how to see it. Moreover, the room affected the childrens' behaviour (e.g. coaxing them to certain locations) to facilitate its own vision processes.



A. Everyone rows on the correct side and the rock is avoided.

B. Room: “Finally, we’ve come to land. Push the boat towards the trees, onto the sand.”

Figure 2.14: KidsRoom

The narrative control program is the core of KidsRoom. It queries the sensor programs for information about what is happening in the room at a given time and then changes how the room responds so that participants are guided through the narrative. The narrative control program is composed of *event loop* and *timers*. The main control program is an event loop that continuously monitors the state of the room, checking all inputs as fast as possible all the time. Vision processes, such as the object tracker, are continuously running and generating data. There are many situations that require an immediate response from the control program. For example, when someone enters the room the system must start tracking the person and the control program must immediately learn of the person’s presence. The narrative control program keeps track of events using *timers* where each event has a timer associated with it. When the event is activated, the timer is reset. The event timer can then be queried each pass through the event loop to see if the event has timed out. The most general event timer is simply used to time story events. For example, a timer is initiated for each short segment of the story. If the timer runs out, the narrative control program may then take some action like playing a narration or moving on to another part of the story. Timers are also used in cross-media coordination to control sound effects and narrations so that sounds don’t play on top of one another. Since it plays pre-fabricated video rather than generating animation on the fly KidsRoom’s intelligence is restricted to only one story, and hence its flexibility and reusability are limited.

2.3.5 Computer games

Interactive storytelling (or story generation) is also found in modern computer games, where the story is altered and reconstructed dynamically according to how the player changes the game world. Most network virtual communities like MUDs (Multi-User Domains), where people meet in a world of virtual reality to socialize and build/change the world, have until recently had only text-based interfaces which describe a scenario and invite users to type in actions for their

characters to perform. More recently, such games use 3D graphic techniques to present a virtual 3D world, putting the same kind of story into a realistic environment. They are called action games, to differentiate them from text games. Virtual reality is the most recent technique of interactive action games. A user who logs into an action game participates in the world, explores it, and might fulfil a task via his/her graphic representation-avatar. (S)he can see the scenes of the virtual reality, other avatars currently logged-into the game or system characters, converse with them, move around in the virtual world, and somehow change the world. Hence the user can reconstruct or modify the story interactively.

Some areas, such as cinema, have influenced action games in fascinating depth, because both are storytelling. The closer coordination between game and film studios undoubtedly results in higher quality games based on films, and vice versa. Narrative in video games is very non-linear where there are subplots that may not lead to anything, but the user(s) has to work them all out to find out what he has to do to win. A labyrinthine plot and convincing design may create a world in which players like to linger, but winning the game is always the final goal.

There are two areas where current computer games could improve their realism and intelligence. First is multimodal interaction. For some commercial reasons, communication in most modern games is still mostly based on text messages or digitized speech streams. This restrains the story from being more lifelike and natural albeit rich graphic presentation is provided. Secondly, game developers devote more resources to advancing games' graphics technology than to enhancing their AI. Within several years, however, the emphasis on graphics will likely have run its course as incremental improvements in the underlying technology lead to only marginal improvements in the game experience (Laird 2001). In the near future, more development and runtime resources will be available to increase game AI complexity and realism. The trend of using AI in computer games is promising. It includes developing intelligent and social autonomous agents, path-finding, animation control, scripting, learning, and various decision-making techniques.

2.3.6 Film-inspired computer animations

There is a large body of knowledge about film montage, accumulated and investigated over the last century, which inspired computer generated animations on how to present stories artistically. The artistic language for virtual environments, however, is not yet clearly defined. This language should be a joint effort for artists and computer scientists. In this section we investigate the cinematic conventions and film techniques (Smith and Bates 1989) and discuss how to apply these montage techniques to create "multi-threaded" narratives in storytelling.

1. *Cut*, probably the most fundamental montage technique, joins separated shots together in the editing process.
2. *Lap dissolve (dissolve)* is a method of making a transition from one shot to another by briefly superimposing one image upon another and then allowing the first image to

disappear. A dissolve is a stronger form of transition than a cut since it establishes a conceptual link between the two scenes.

3. *Pan shots*. In a pan shot a stationary camera turns horizontally and smoothly scans the scene to reveal new areas.
4. *Strange camera angles*. Unusual viewpoints can suggest unusual situations or convey symbolic meaning. *Citizen Kane* provides numerous examples. As Kane's mistress sings, the camera pulls higher, mimicking the soaring of her voice; and the camera shoots down at Susan, forcing the viewer to consider her condescendingly.
5. *Cross-cutting (parallel editing)* is a method of editing in which the point of view switches alternately from events at one location to those of another related action. The action is usually simultaneous and used to create a dynamic tension as in a chase scene or to establish links between the scenes presented in parallel.
6. *Flashback* is a segment of film that breaks normal chronological order by shifting directly to time past. Flashback may be subjective (showing the thoughts and memory of a character) or objective (returning to earlier events to show their relationship to the present).
7. *Subliminal shots* is in the development of scene X, the film quickly flashes some image Y that recalls or emphasizes some important idea such as to underscore some psychological problems of a character. The most extreme example of this technique is probably Friedkin's use of actual subliminal shots to try to heighten the horror of *The Exorcist*.
8. *Visual rhythm and distortion of natural rhythms*. Visual rhythm is the regular, coordinated linking of things like image, movement, and action to time. Smith and Bates (1989) cite several examples of battlefields and marching soldiers. The purpose of the technique is apparently to provide some deeper aesthetic consistency. Distortion of natural rhythms are usually used in some situations to feature special feeling such as using slow-motion to present helplessness in a nightmare or looming dangers, and fast-motion to express ridiculousness.
9. *Zoom-freeze*. The camera zooms in on some important facet of the scene and freezes there. This technique lends extra emphasis to that facet by arresting the viewer's attention.
10. *Iris* is a somewhat archaic technique, and is not often seen in contemporary cinema. Irising to some important detail means physically masking out everything else in the scene. This is similar to the close-up except the exclusion of the non-emphasized details is more deliberate.
11. *Imagery* is a visual allusion, a technique which can greatly enhance the effect of a film. It may be subsumed in cut or flashback. Computer graphics offers a new way to express characters' imagery, i.e. opening a second window and presenting the allusion in it. This

new channel which is impossible in conventional film-editing allows direct communication of character's thoughts and mental-related activities. It resolves visually questions that are left open textually which somebody argues as a flaw of the film media which renders full texts by sacrificing narrative momentum.

12. *Voiceover* is the voice of an unseen narrator or of an onscreen character not seen speaking. It concerns the theory of narration in movie — *no narrator*, *omniscient external narrator*, *character as narrator*, etc. Each of these has its own purposes in communication of information to the user, for example, character as narrator voiceovers communicate directly the information spoken and indirectly the beliefs and opinions of the speaker.

We see the challenge in intelligent storytelling is not presenting contents by these montage techniques, but selecting contents, and more ambitious, by choosing available techniques automatically to achieve communicative purpose. That is, why should the system choose this detail and this technique, not some other to present the story? Therefore, in addition to being able to implement a technique, an intelligent storytelling system also needs some mechanisms for choosing to implement it. For example, to express the urgency of being chased, which montage technique the system would choose, an ordinary shot showing both escaper and chaser running, cross-cut of escaper and chaser, distortion of natural rhythms by slow-motion of running, or imagery of being caught? This is a more artistic task and use of each technique is linked to deeper aesthetic vision, not something easily specified for automation. Hence, we dichotomise tasks roughly between the directly automatable techniques that would be supplied as part of a kernel package and the ones whose calling mechanisms require substantial creative imagination from a human which would be supplied as customisable options for advanced users such as movie directors and computer animation artists.

2.3.7 Computer graphics films

Hand-drawn animated films dating back to Disney's *Snow White and the Seven Dwarves* in the 1930s, where artists created 2D cartoons by drawing each frame on sheets of paper. Now, 3D computer graphics films such as Pixar's *Toy Story*, *Monsters, Inc.*, *Finding Nemo* and Blue Sky's *Ice age* demonstrate the latest advancement to computer animation technologies. Making a computer animate film involves 3D artists and programmers working together in close collaboration. The process starts with the development of the story and continues with modelling the geometry, adding articulation controls and using these controls to animate the characters, simulating things like water in *Finding Nemo* and the monster's fur in *Monsters, Inc.*, defining the look of the surfaces, putting lights in the scene, adding special effects, rendering, and post-production. The process of adding articulation controls and animating characters is closely related to our topic.

Thanks to graphic authoring software dedicated to 3D modelling and animation such as 3D Studio Max (2006), Maya (2006), and Poser (2006) which are able to automate things like

interpolating the behaviour of millions of frames between key frames, and physics engines like Havok Physics (2006), PhysX (Novodex Physics 2006), ODE (2006), and Newton game engine (2006), which provide rigid body simulation such as modelling force and weights, gravity, springs, wind, and other conditions of the animated environment, the process of whole animation production is much less tedious comparing to the early days 2D cartoons. However, many works such as 3D modelling and key frame setting are still manually generated on computers. Our research tries to automate this process as much as possible by converting natural language text to 3D animations.

2.3.8 Video-based computer animation generation

Another direction of computer animation generation is video-based computer animation generation which creates controlled animations by re-arranging recorded video frames of a moving object (Schödl and Essa 2002, Schödl et al. 2000). Video-based computer animation is a useful alternative method of computer animation generation. With this technique, the user can specify animations using an algorithm which is automatically optimized by repeated replacement of video sprite sub-sequences and compute video sprite transitions in the input footage. This technique is also used to create character animations of animals, which are difficult to train in the real world and to animate as 3D models.

2.4 Summary of previous systems

We have surveyed text-to-graphics conversion, embodied agents and virtual humans, and multimodal storytelling in section 2.1-2.3. Table 2.3 gives an overall comparison focusing on input and output modalities of these systems. We see that text-to-graphics systems have good language understanding components but fewer input channels. Although most have 3D animation they don't provide high quality graphics. Multimodal storytelling systems and virtual humans have more enriched I/O interfaces and better graphic quality but weaker NLP, and some of them simply ignore NLP completely.

Most related systems mix text, graphics (2D or 3D) and speech (some with additional non-speech audio) modalities. Static graphical displays in WordsEye constrain presentation of dynamic information such as actions and events. The graphics of some systems which present animations is 2D or ready-made, such as KidsRoom (Bobick et al. 1996) and Gandalf (Thórisson 1996). SI/SONAS (Ó Nualláin and Smith 1994, Kelleher et al. 2000), Larsen and Petersen's (1999) interactive storytelling, Cassell's agents, and Narayanan's primitive-based language animation (Narayanan et al. 1995) present 3D animations. However, SI/SONAS needs user intervention to navigate and modify the 3D world and hence its performance is like a 3D browser which responds to speech commands. The application domain of SI/SONAS is limited to scene description and spatial relationships. CarSim (Dupuy et al. 2001) interprets texts of car accident reports using information extraction techniques and creates 3D animations. SAM and

Categories	SYSTEMS	NLP		Multimodal interaction								
		Natural language generation	Natural language understanding	Input Media			Output Media					
				Typed-in text	Speech recognition	Vision recognition	Text	Audio		visual		
								Text to speech	Non-speech audio	Static graphics	Animation	
									2D	3D		
Text-to- Graphics	CarSim		v	v								v
	WordsEye		v	v					v			
	Narayanan's animated icons		v	v								v iconic
	SI/SONAS		v	v	v							v
Embodied agents/ Virtual humans	Jack		v	v								v
	Improv (virtual theatre)			script	v	v		v				v
	Virtual human & smart object					v			v			v
	Cassell's SAM & REA	v	v	v	v	v		v				v
	Gandalf		v		v	v		v			v	
Multimodal Storytelling	Larsen & Petersen's Interactive Storytelling				v	v	v	v	v			v
	AESOPWORLD	v	v	v			v	v	v		v	
	OZ	v	v	v			v				v	
	KidsRoom				v	v		v	v		v	

Table 2.1: Summary of the I/O of related systems

REA (Cassell et al. 2000) focus on simulating humanoid behaviour in conversation, and are applied to human-computer dialogue. Although Larsen and Petersen's interactive storytelling environment presents non-agent 3D animations to tell stories, the processes of converting language to graphics are not automatic, i.e. it is not "intelligent" storytelling, because its animation generation relies on programming-language-like scripts. Although Narayanan's language animation may generate 3D animation automatically, the image quality of icons is inadequate.

Badler's (1997) *Jack*, Improv (Perlin 1996), and virtual human and smart object (Kallmann and Thalmann 2002) focus on virtual human simulation and animation. Jack has proper task planning and accurate biomechanical simulation, but it depends on user interaction and hence provides little automation. Like Larsen & Petersen's interactive storytelling, Improv relies on script input and has little NLP. Kallmann and Thalmann's virtual humans have high quality human modelling and motion, but don't have NLP at all.

2.5 Multimodal allocation

Multimodal presentations convey redundant and complementary information. The fusion of multiple modalities asks for synchronising these modalities. Typically the information and the modality (modalities) conveying it has the following relationship:

- A single message is conveyed by at least one modality.
- A single message may be conveyed by several modalities at the same time.
- A specific type of message is usually conveyed by a specific modality, i.e. a specific modality may be more appropriate to present a specific type of message than other modalities. For instance, visual modalities are more fitting for colour and spatial information than language.

An optimal exploitation of different media requires a presentation system to decide carefully when to use one medium in place of another and how to integrate different media in a consistent and coherent manner. Media integration requires the selection and coordination of multiple media and modalities. The selection rules are generalized to take into account the system's communicative goal, a model of the audience, features characterizing the information to be displayed and features characterizing the media available to the system. To tell a story by complementary multi-modalities available, a system needs to divide information and assign it to particular media according to their features and cognitive economy. Since each medium can perform various communicative functions, designing a multimedia presentation requires determination of what information is conveyed by which medium at first, i.e. media allocation according to *media preferences*. For example, presenting spatial information like position, orientation, composition and physical attributes like size, shape, color by graphics; presenting

events and actions by animation; presenting dialogue between characters and temporal information like “ten years later” by language.

Feiner and McKeown (1991) have introduced the media preferences for different information types in their COMET knowledge based presentation system. COMET uses a *Functional Unification Formalism* (FUF) to implement its media allocation rules, for example, COMET requires all actions to be presented by both graphics and text (Figure 2.15A), and the input is represented using the same formalism, a set of attribute-value pairs (Figure 2.15B). The annotation is accomplished by unifying the task grammar (Figure 2.15A) with the input (Figure 2.15B). For each attribute in the grammar that has an atomic value, any corresponding input attribute must have the same value. If the values are different, unification fails. When the attributes match and the values are the same, if the input does not contain some grammar attributes, the attributes and their values are added to the input. Any attributes that occur in the input but not in the grammar remain in the input after unification. Thus, the attribute-value pairs from both input and task grammar are merged. Figure 2.15C is the result after unifying A and B.

The above methods in media allocation give useful insights into the problem of choosing appropriate media to express information and to achieve more economical and effective presentation.

```
((process-type action) ;; If process is an action
(media-graphics yes)   ;; use graphics
(media-text yes)       ;; use text
...))
```

A. Task grammar of COMET

```
(substeps
 [((process-type action)
 (process-concept c-push)
 (roles (...))...])
```

B. Input representation in FUF form

```
(substeps
 [((process-type action)
 (process-concept c-push)
 (roles (...))
 (media-graphics yes)
 (media-text yes)
 ...)])
```

C. Result after unification

Figure 2.15: Functional unification formalism in COMET

2.6 Non-speech audio

Here we consider the use of non-speech audio in multimodal presentation. The use of non-speech audio to convey information in multimedia presentation is referred to in the human factors literature

as auditory display. Non-speech auditory information is prevalent in the real world. Furthermore, the human auditory system has special processing abilities for various aspects of non-speech sound such as music. Besides basic advantages, such as reducing visual clutter, avoiding visual overload, and not requiring focused attention, auditory displays have other benefits. At the cognitive level, experiments showed that detection times for auditory stimuli were shorter than for visual stimuli — Speeth’s (1961) experiments showed that sonified seismograph data could be interpreted by listeners more rapidly than visual seismograph data, and that short-term memory for some auditory information is superior to the short-term memory for visual information.

Current research in the use of non-speech audio can generally be divided into two approaches. The first focuses on developing the theory and applications of specific techniques of auditory display. The techniques of *auditory icons*, *earcons*, *sonification*, and *music synthesis* have dominated this line of research and are discussed in detail here below. The second line of research examines the design of audio-only interfaces — much of this work is concerned with making GUIs accessible to visually-impaired users, or explores the ways in which sound might be used to extend the existing visual interface, i.e. where and how audio might be utilized to increase the effectiveness of visual interfaces.

There is a mapping between audio and objects, events, status, emotions or other data being transmitted. Typically the mapping is chosen to be easily understood by the listener, so the cultural and natural mappings of sound in the users head should be considered. Similarly, the position along the sound dimension is chosen to be non-annoying or to capitalize on the perception of melodies. This is more important for synthesised music.

2.6.1 Auditory icons

Auditory icons are caricatures of naturally occurring sounds which convey information by analogy with everyday events (Gaver, 1986). Gaver motivates this technique by questioning our basic notion of listening. In Gaver’s view when we listen to sounds in our daily lives we do not hear the pitch or the duration of the sound. Rather, we hear the source of the sound and the attributes of the source. He refers to two types of listening: *musical listening* and *everyday listening*. Everyday listening includes common sounds such as the sound of pouring water, tearing paper, a car engine, or a telephone ringing. People tend to identify these sounds in terms of the object and events that caused them, describing their sensory qualities only when they could not identify the source events (Gaver, 1989). Supposing that everyday listening is often the dominant mode of hearing sounds, Gaver argues that auditory displays should be built using real-world sounds. Auditory icons accompanying daily life events can also be a major source of nonspeech audio in multimodal storytelling systems. Certainly the intuitiveness of this approach to auditory display results in a more vivid story presentation.

Gaver has successfully applied these auditory icons to several different domains. The SonicFinder is a Macintosh interface that has been extended with auditory icons, conveying information with real-world sounds such as the clink of glass when a window is selected or the crash of a metal trash can when a file is placed in the trashcan graphical icon (Gaver 1989). Informal feedback from users indicated that the auditory icons enhanced the interface. Two primary advantages are cited. Users feel an increased sense of engagement with the model world of the computer. The use of audio feedback increases the flexibility of the system because, among other things, the users don't have to always attend to the screen for information. Theoretically, the advantage of auditory icons seems to be in the intuitiveness of the mapping between sounds and their meaning. Gaver almost regards the mapping as a part of the hearing process: we do not seem to hear sounds, but instead the sources of sound.

2.6.2 Earcons

Earcons are melodic sounds, typically consisting of a small number of notes, with musical pitch relations (Gaver 1989). They relate to computer objects, events, operations, or interactions by virtue of a learned mapping from experience. The basic idea of earcons is that by taking advantage of sound dimensions, such as pitch, timbre, and rhythm, information can be communicated to the user efficiently. Of the four basic techniques for auditory display, earcons have been used in the largest number of computer applications. The simplest earcons are auditory alarms and warning sounds such as incoming e-mail notification and program error in the Windows operating system sounds properties, and low battery alarm on mobile phones. The effectiveness of an earcon-based auditory display depends on how well the sounds are designed. Well-designed earcons can capitalize on the auditory systems abilities for musical processing, psycho-acoustic capabilities, and cognitive level memory performance.

2.6.3 Sonification

Sonification is the technique of translating multi-dimensional data directly into sound dimensions. Typically, sound parameters such as amplitude, frequency, attack time, timbre, and spatial location are used to represent system variables (Bly et al. 1987). The goal is synthesising and translating data from one modality, perhaps a spatial or visual one, to the auditory modality. Sonification has been widely applied to a wealth of different domains: synthesised sound used as an aid to data visualisation, especially abstract quantitative data, for program comprehension, and monitoring performance of parallel programs.

2.6.4 Music synthesis

In synthesised music of non-speech audio, sounds are interpreted for consonance, rhythm, melodic content, and hence are able to present more advanced information such as emotional content. Schwanauer and Levitt (1993) review the history of automated music synthesis, from the stochastic music of Xenakis in the 1950s to modern recording and algorithmic composition. Computer-based music composition initiated in the mid 1950s when Lejaren Hillier and Leonard Isaacson conducted their first experiments with computer generated music on the ILLIAC computer at the University of Illinois. They employed both a rule-based system utilising strict counterpoint (a technique of combining two or more melodic lines in such a way that they establish a harmonic relationship while retaining their linear individuality), and a probabilistic method based on Markoff chains which was also employed by Xenakis. These procedures were applied with variation to pitch and rhythm resulting in *the ILLIAC Suite* a series of four pieces for string quartet. The recent history of automated music and computers is densely populated with examples based on various theoretical rules from music theory and mathematics. While the ILLIAC Suite used known examples of these, developments in such theories have added to the repertoire of intellectual technologies applicable to the computer. Amongst these are serial music techniques, application of music grammars, sonification of fractals, and chaos equations, and connectionist pattern recognition techniques based on work in neuro-psychology and artificial intelligence. Arguably the most comprehensive of the automated computer music programs is Cope's experiments in Music Intelligence, which performs a feature analysis on a database of coded musical examples presented to it, and can then create a new piece which is a pastiche of those features.

Figure 2.16 illustrates the four types of non-speech audio described above and their common features. Auditory icons and earcons are small pieces of audio clips (audio icons); sonification and synthesised music can generate audio from other modal data; and earcons and synthesised music are melodic sound.

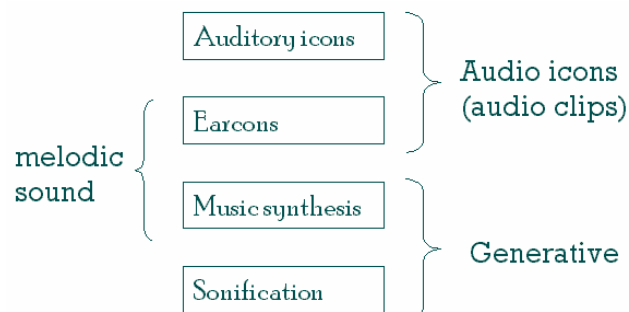


Figure 2.16: Four types of non-speech audio

2.7 Mental imagery in cognitive science

Mental imagery is defined as the human ability to visualise (or construct mental pictures) of various concepts, where the concept can be a simple object (e.g. a noun) or as complex as an entire sentence or paragraph. Although there is agreement among philosophers and cognitive scientists regarding the existence of mental imagery, controversy remains with regard to the mechanisms in the brain that support this function. The work in mental imagery provides some cognitive basis for language visualisation. Figure 2.17 illustrates the mental architecture and meaning processing widely accepted in cognitive science (Tye 1995, 2000). Ellipses denote meaning processing and rectangles denote results of each level.

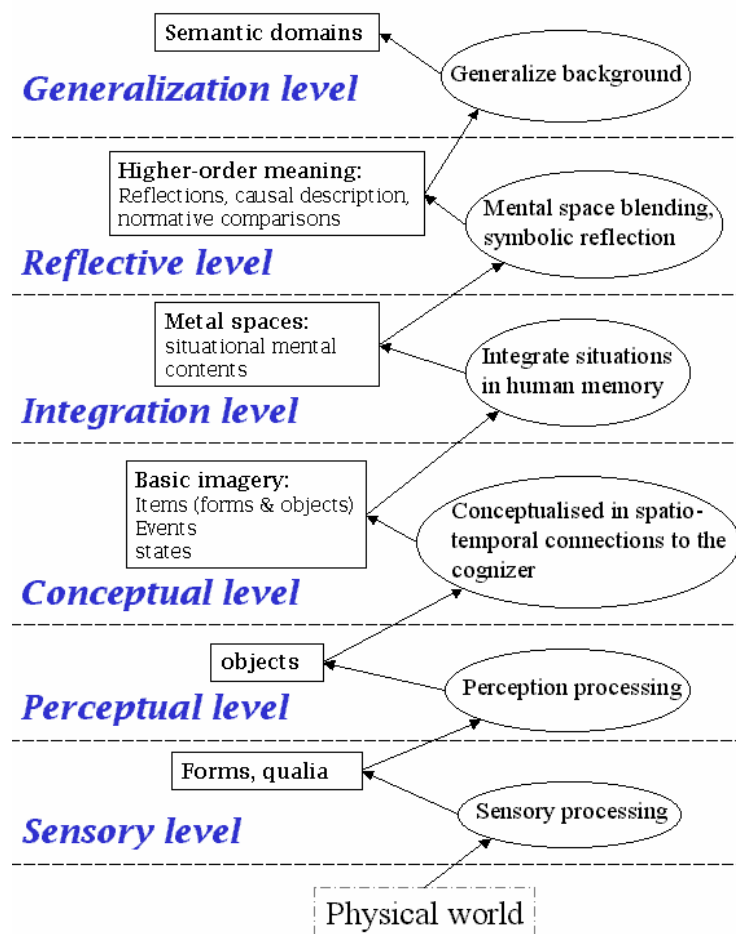


Figure 2.17: Mental architecture and meaning processing

In the theoretical perspective of an exploration of cognition and meaning, a mental space is a real semantic unit that on a specific level of mental processing significantly integrates other important semantic units of that same level or of underlying levels. Sensory processing, on the base level, lets people perceive forms or qualia; perceptual processing lets people perceive objects; configurations of objects are further conceptualised in such spatio-temporal connections to the

cognizer that they are experienced as existing in situations relevant to this cognizer. These units constitute the basic imagery that makes it possible for us to represent items: forms and objects, events and states, instead of just experiencing them and *presenting* them to others. They are universally shaped as finite or local spatial and temporal wholes, and they can additionally be compared to scenes performed on the stage of a theatre. These theatrical wholes are *mental spaces*. Neither a list of objects, a color, a sound, a feeling, nor the contour of a body is per se a mental space; they are preparatory perception integrations, but the situated wholes are. Human memory is theatrical in the sense that it predominantly operates on information from this level of integration. Mental spaces further integrate when real higher-order meanings are built, beyond these situational mental contents, through processes involving blending; reflections, notional meanings, such as those appearing in causal descriptions of events and changes, narrative accounts of intentional doings, normative comparisons and judgments. Beyond the reflective level of mental space blending, as its generic background, are the larger units called semantic domains — a level of “regions in being”.

The finite mental spatiality of mental spaces allows the individual to interact not only with the surrounding physical spatiality but also with other individuals, and to hold other mental spaces present in consciousness in addition to the one representing the present, then to let out-of-presence mental spaces generate meaning relevant for the present. This is also the cognitive foundation of Schank’s scripts. Beyond the level of represented situations in the architecture of human mind is abstract thinking such as discourse-based or symbolic reflection.

Linguistic meaning may use all levels of mental architecture and thus express their transversal coherence. Jackendoff (1987) outlines conceptual semantics, an intermediate representation between the perceptual level and the conceptual level, providing a link between language and Marr’s (1982) computational theory of vision. Marr suggests that the human ability to categorise objects and recognise individuals is due to the *conceptual primitives* TOKEN and TYPE, where the former is used to label an individual object and the latter is used to label categories of objects.

2.8 Summary

This chapter has reviewed and compared a range of systems, from multimodal storytelling systems, automatic text-to-graphics systems, to embodied agents and virtual humans. We also discussed multimodal allocation and four types of non-speech audio in which auditory icons are an ideal medium to supplement animation. In addition, related work on mental imagery in cognitive science was investigated. The next chapter discusses previous work on natural language and multimodal semantic representations.

Chapter 3

Natural Language Semantics

In this chapter, we discuss various natural language and multimodal semantic representations, Allen's interval-based temporal relations, existing computational lexicons, and language ontology.

3.1 Natural language semantic representations

This section discusses natural language semantic representations including semantic networks, CD and scripts, Lexical Conceptual Structure (LCS), event-logic truth conditions, and X-schemas and f-structs.

3.1.1 Semantic networks

A semantic network, as defined in Quillian (1968), is a graph structure in which nodes represent concepts, while the arcs between these nodes represent relations among concepts. From this perspective, concepts have no meaning in isolation, and only exhibit meaning when viewed relative to the other concepts to which they are connected by relational arcs. In semantic networks then, structure is everything. Taken alone, the node *Scientist* is merely an alphanumeric string from a computer's perspective, but taken collectively, the nodes *Scientist*, *Laboratory*, *Experiment*, *Method*, *Research*, *Funding* and so on exhibit a complex inter-relational structure that can be seen as meaningful, inasmuch as it supports inferences that allow us to conclude additional facts about the *Scientist* domain. Semantic networks are widely used in natural language processing, especially in representing lexical semantics such as WordNet (Beckwith et al. 1991), a lexicon in which English vocabulary is organised into semantic networks.

3.1.2 Conceptual Dependency theory and scripts

Natural language processing systems store the ideas and concepts of input language in memory which is termed *conceptual representation*. Conceptual representation is significant for interpreting a story in intelligent storytelling. It may help find how information from texts is encoded and recalled, improve machine understanding and present stories more precisely. Conceptual Dependency, introduced by Schank (1972), was developed to represent concepts acquired from natural language input. The theory provides eleven primitive actions and six

primitive conceptual categories (Figure 3.1). These primitives can be connected together by relation and tense modifiers to describe concepts and draw inferences from sentences.

ATRANS - Transfer of an abstract relationship. E.g. give.
 PTRANS - Transfer of the physical location of an object. E.g. go.
 PROPEL - Application of a physical force to an object. E.g. push.
 MTRANS - Transfer of mental information. E.g. tell.
 MBUILD - Construct new information from old. E.g. decide.
 SPEAK - Utter a sound. E.g. say.
 ATTEND - Focus a sense on a stimulus. E.g. listen, watch.
 MOVE - Movement of a body part by owner. E.g. punch, kick.
 GRASP - Actor grasping an object. E.g. clutch.
 INGEST - Actor ingesting an object. E.g. eat.
 EXPEL - Actor getting rid of an object from body.

A. Primitive actions in CD

PP -- Real world objects.
 ACT -- Real world actions.
 PA -- Attributes of objects.
 AA -- Attributes of actions.
 T -- Times.
 LOC -- Locations.

B. Primitive conceptual categories in CD

Figure 3.1: Conceptual Dependency primitives

For example, the sentence “I gave John a book” can be depicted in CD theory as shown in Figure 3.2. The double arrow indicates a two-way link between actor and action. The letter ‘P’ over the double arrow indicates past tense. The single-line arrow indicates the direction of dependency. ‘o’ over the arrow indicates the object case relation. The forficate arrows describe the relationship between the action (ATRANS), the source (from) and the recipient (to) of the action. The ‘R’ over the arrow indicates the recipient case relation.

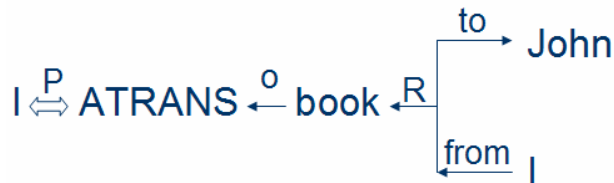


Figure 3.2: Conceptual Dependency representation of “I gave John a book.”

CD theory makes it possible to represent sentences as a series of diagrams depicting actions using both abstract and real physical situations. The agents and objects in the sentences are represented. The process of splitting the knowledge into small sets of low-level primitives makes the problem solving process easier, because the number of inference rules needed is reduced. Therefore CD theory could reduce inference rules since many inference rules are already represented in CD structure itself.

However, knowledge in sentences must be decomposed into fairly low level primitives in CD, therefore representations can be complex even for relatively simple actions. In addition, sometimes it is difficult to find the correct set of primitives, and even if a proper set of

primitives are found to represent the concepts in a sentence, much inference is still required. Narayanan et al.'s (1995) text-to-animation system, discussed in Chapter 2, section 2.1.3, shows another limitation of CD. The graphic display in the system is iconic, without body movement details because CD theory focuses on the inferences of verbs and relations rather than the visual information of the primitive actions.

Additionally, since people have routines, for example, routine ways of responding to greetings, to go to work every morning, as should an intelligent knowbot, Schank introduced *scripts*, expected primitive actions under certain situations, to characterize the sort of stereotypical action sequences of prior experience knowledge within a human being's *common sense* which computers lack, such as going to a restaurant or travelling by train. A script could be considered to consist of a number of slots or frames but with more specialised roles. The components of a script include:

entry conditions — these must be satisfied before events in the script can occur.

results — conditions that will be true after events in script occur.

props — slots representing objects involved in events.

roles — persons involved in the events.

track — variations on the script. Different tracks may share components of the same script.

scenes — the sequence of *events* that occur. *Events* are represented in CD form.

For example, to describe a situation *robbing a bank*. The *Props* might be:

- Gun, *G*.
- Loot, *L*.
- Bag, *B*
- Get away car, *C*.

The *Roles* might be:

- Robber, *R*.
- Cashier, *M*.
- Bank Manager, *O*.
- Policeman, *P*.

The *Entry Conditions* might be:

- *R* is poor.
- *R* is destitute.

The *Results* might be:

- *R* has more money.
- *O* is angry.
- *M* is shocked.
- *P* is shot.

There are 3 scenes:

- obtaining the gun

- ❑ robbing the bank
- ❑ escape with the money (if they succeed).

The scene robbing the bank can be represented in CD form as the following:

```

R PTRANS R into bank
R ATTEND eyes M, O and P
R MOVE R to M position
R GRASP G
R MOVE G to point to M
R MTRANS 'Give me the money or ...' to M
P MTRANS 'Hold it. Hand up.' to R
R PROPEL shoots G
P INGEST bullet from G
M ATRANS L to R
R ATRANS L puts in B
R PTRANS exit
O ATRANS raises the alarm

```

Therefore, provided events follow a known trail we can use scripts to represent the actions involved and use them to answer detailed questions. Different trails may be allowed for different outcomes of scripts (e.g. the bank robbery goes wrong). The disadvantage of scripts is that they may not be suitable for representing all kinds of knowledge.

Schank and his colleagues developed applications based on his CD theory. SAM (Script Applier Mechanism) is a representative system. It reads short stories that follow basic scripts, then outputs summaries in several languages and answers questions about the stories to test its comprehension. SAM had four basic modules: (1) a parser and generator based on a previous program, (2) the main module - the Script Applier, (3) the question-answer module, and (4) the Russian and Spanish generators. SAM had deficiencies when a story digresses from a script. In 1980, another system called IPP (Integrated Partial Parser) (Schank et al. 1980) was developed. It used more advanced techniques than SAM, in addition to Concept Representation primitives and scripts it used plans and goals too. IPP was built to read newspaper articles of a specific domain, and to make generalizations about the information it read and remembered. An important feature of IPP is that it could update and expand its own memory structures. Moreover, another script-based story understanding system called PAM (Plan Applier Mechanism) was developed later by Wilensky (1981). PAM's understanding focuses on plans and goals rather than scripts.

3.1.3 Lexical Conceptual Structure (LCS)

Lexical Conceptual Structure (LCS) is a semantic representation proposed by Jackendoff (1990). It takes the view that there should be a method of mapping syntax onto semantics (and

vice versa) and provides a sound foundation of semantics onto which a strict set of rules for communication between syntax and semantics could be built.

Based on the linguistic minimalism principle, Jackendoff explains that conceptual structure is made up of a set of entities (conceptual primitives or ontological categories) that combine to perform a number of meaning functions. The list of entities (e.g. THING, EVENT, STATE, PLACE, PATH) is not intended to be exhaustive, a case may present itself which requires a further entity to be added to the list or to replace one with something more general. Keeping minimalism in mind, it should be clear that the number of distinct categories should remain as low as possible. A formalisation for forming conceptual structures from the ontological categories is shown in Figure 3.3.

1. [PLACE X] -> [PLACE placePredicate [THING Y]]
2. [PATH X] -> [PATH pathPredicate [THING Y]]
[PATH pathPredicate [PLACE Y]]
3. [EVENT X] -> [EVENT go [THING Y], [PATH Z]]
[EVENT stay [THING Y], [PLACE Z]]
[EVENT cause [THING Y], [EVENT Z]]
[EVENT inchoate [STATE]]
4. [STATE X] -> [STATE be [THING Y], [PLACE Z]]
[STATE orient [THING Y], [PATH Z]]
[STATE extension [THING Y], [PATH Z]]

Figure 3.3: Conceptual structures from ontological categories in LCS

Using this formalism, LCS can deal extremely well with sentences involving purely spatial relations, for instance, the semantic structure of “John went towards the house” is represented as:

[EVENT go [THING JOHN], [PATH towards [THING HOUSE]]]

“John put the cup on the table” is represented as:

[EVENT cause [THING JOHN], [EVENT go [THING CUP], [PATH to [PLACE on [THING TABLE]]]]]

However, the EVENT predicates that LCS defines are far too coarse for the diversity of human actions, and hence not suitable for language animation. For instance, the EVENT predicate “cause” in LCS is overloaded by including both phrasal causations (e.g. “cause”, “force”, “prevent”, “impede”) and lexical causatives (e.g. “put”, “push”, “break” (vt.), “open”, “kill”).

We discuss now two other semantic representations of simple action verbs linking vision and language: event-logic truth conditions and f-structs. Both of them are mainly designed for verb labelling (visual recognition), whereas the focus of our work is the reverse process — language visualisation. A common problem in the tasks of both visual recognition

and language visualisation is to represent visual semantics of motion events, which happen both in the *space* and *time* continuum.

3.1.4 Event-logic truth conditions

Traditional methods in visual recognition segment a static image into distinct objects and classify those objects into distinct object types. Siskind (1995) describes the ABIGAIL system which focuses on segmenting continuous motion pictures into distinct events and classifying those events into event types. He proposed event-logic truth conditions for simple spatial motion verbs' definition used in a vision recognition system. The truth conditions are based on the spatial relationship between objects such as *support*, *contact*, and *attachment*, which are crucial to recognize simple spatial motion verbs. According to the truth condition of the verbs' definition, the system recognizes motions in a 2D line-drawing movie. He proposed a set of perceptual primitives that denote primitive event types and a set of combining symbols to aggregate primitive events into complex events. The primitives are composed of three classes: time independent primitives, primitives determined from an individual frame in isolation, and primitives determined on a frame-by-frame basis. Using these primitives and their combinations, he gives definitions of some simple motion verbs and verifies them in his motion recognition program ABIGAIL.

Siskind's event-logic definition has two deficiencies: (1) lack of conditional selection, i.e. this framework does not provide a mechanism for selection restrictions of the arguments, and (2) overlapping between primitive relations. So some definitions are arbitrary in some degree. They do not give a necessary and sufficient truth-condition definition for a verb. For example: the definitions for 'jump' and 'step' are the following.¹

$$\begin{aligned} \text{jump}(x) &= \text{supported}(x); (\neg \diamond \text{supported}(x) \wedge \text{translatingUp}(x)) \\ \text{step}(x) &= \exists y (\text{part}(y, x) \wedge [\text{contacts}(y, \text{ground}); \neg \diamond \text{contacts}(y, \text{ground}); \\ &\quad \text{contacts}(y, \text{ground})]) \end{aligned}$$

The definition of "jump" means *x* is supported, and then not supported AND moves up in the immediate subsequent interval. The definition of 'step' can be interpreted that there exists *y*, could be a foot, which is part of the *x*, AND *y* first contacts ground, then does not contact, and finally contacts ground again. From the two definitions, we see that the definition of 'step' can also define the motion of 'jump' or 'stamp (a foot)'. Hence, the definition of one verb can also be used to define other verbs. Also, an alternative definition of 'step' based on Siskind's methodology could be:

$$\begin{aligned} \text{step}(x) &= \exists y_1, y_2 (\text{part}(y_1, x) \wedge \text{part}(y_2, x) \wedge \\ &\quad [(\text{contacts}(y_1, \text{ground}) \wedge \neg \diamond \text{contacts}(y_2, \text{ground}))]; \end{aligned}$$

¹ a;b means event b occurs immediately after event a finishes. $\diamond a@i$ means a happens during *i* or a subset of *i*, so $\neg \diamond \text{supported}(x)@i$ means 'x is not supported in any time during *i*.'

$$(\neg \diamond \text{contacts}(y1, \text{ground}) \wedge \text{contacts}(y2, \text{ground})) ;$$

$$\text{contacts}(y1, \text{ground})])$$

The definition describes the alternate movement of two feet $y1$ and $y2$ contacting the ground in a step. Hence, one verb can be defined by many definitions.

Siskind's visual semantic representation method is subject to ambiguity, i.e. a single verb can legitimately have different representations such as "step", and a single representation can correspond to different events such as the first definition of 'step' can define "jump" and "stamp" also. This arbitrariness in the event definition causes some false positives and false negatives when ABIGAIL recognizes motions in animation. The deficiency of conditional selection causes some loose definitions, admitting many false positives, e.g. the definition of "jump" admits unsupported upward movement of some inanimate objects like ball or balloon, because it does not have any semantic constraints on the fillers of argument x , indicating that x should be an animate creature (non-metaphor usage).

The arbitrariness of verb definition might arise from two problems in his primitives. One is the overlapping between some primitives in an individual frame class, such as $\text{contacts}()$, $\text{supports}()$, and $\text{attached}()$. For instance, when one object is supported by another, it usually contacts the supporting object. The other problem is that some primitives in a frame-by-frame class are not atomic, i.e. could be described by combinations of others, such as $\text{slideAgainst}(x, y)$ might be performed by $\text{translatingTowards}() \wedge \text{supports}(y, x)$.

In his methodology, Siskind does not consider internal states of motions (e.g. motor commands), relying instead on visual features alone, such as support, contact, and attachment. *Event-logic truth condition* works in vision recognition programs such as ABIGAIL. However, for vision generation applications internal states of motions (e.g. intentions, motor commands) are required. X-schemas (eXecuting-schema) and f-structs (Feature-structures) (Bailey et al. 1997) examine internal states of motor actions.

3.1.5 X-schemas and f-structs

Bailey et al.'s (1997) x-schemas (eXecuting schemas) and f-structs (Feature-STRUCTures) representation combines schemata representation with fuzzy set theory. It uses a formalism of Petri nets to represent x-schemas as a stable state of a system that consists of small elements which interact with each other when the system is moving from state to state (see Figure 3.4). A Petri net is a bipartite graph containing *places* (drawn as circles) and *transitions* (rectangles). Places hold *tokens* and represent predicates about the world state or internal state. Transitions are the active component. When all of the places pointing into 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 token in its output place. As a side effect a firing transition triggers an external action. From these constructs, a wide variety of control structures can be built.

Each sense of a verb is represented in the model by a feature-structure (f-struct) whose values for each feature are probability distributions. Table 3.1 shows the f-structure of one word-sense of *push*, using the *slide* x-schema (Figure 3.4). It consists of two parts, *motor parameter features* and *world state features*. Motor parameter features concern the hand motion features of the action *push*, which invoke an x-schema with corresponding parameters, such as force, elbow joint motion, and hand posture. World state features concern the object that the action is performed on, such as object shape, weight, and position.

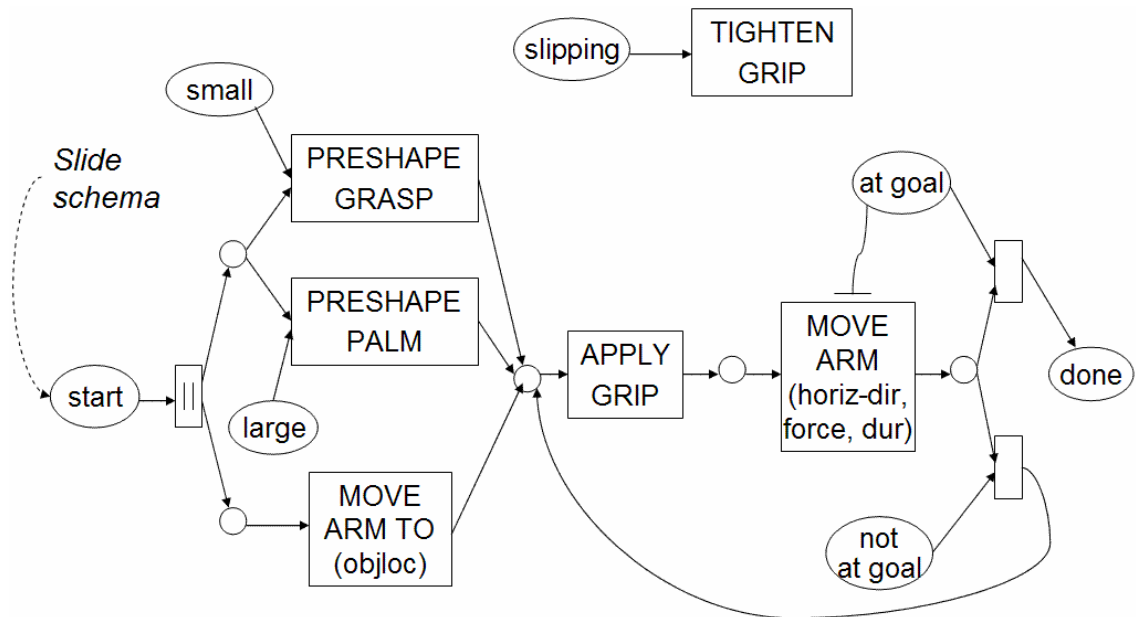


Figure 3.4: *slide* x-schema (Bailey et al. 1997)

<i>Motor parameter features</i>						<i>World state features</i>		
x-schema	posture	elbow joint	direction	aspect	acceleration	object	weight	position
slide	palm	Flex extend	left right	once	Low med high	cube	2.5lbs	(100,0,300)

Table 3.1: F-struct of one verb sense of *push* using slide x-schema

The probabilistic feature values in this structure are learned from training data. The application based on this representation is a system trained by labelled hand motions and learns to both label and carry out similar actions by a simulated agent. It can be used in both verb recognition and performing the verbs it has learned. However, the model requires training data to create the f-structs of verbs before it can recognize and carry them out. The x-schema model is a procedural model of semantics because the meanings of most action verbs are procedures of performing the actions. The intuition of this model is that various parts of the semantics of events, including the aspectual factors, are based on schematised descriptions of sensory-motor processes like inception, iteration, enabling, completion, force, and effort.

Traditional sentence/phrase level semantic representations include First Order Predicate Calculus (FOPC) and predicate-argument models, e.g. predicate-argument models list as many arguments as are needed to incorporate all the entities associated with a motion, such as

`give(sub, indirectObj, directObj), cut(sub, obj, tool)`. Event-logic and x-schemas work on the word level (action verbs), and Schank's CD theory also provides fourteen primitive actions to represent and infer verb semantics, i.e. at the word level. However, there is a dearth of movement details in Schank's CD theory which may result in lack of adequate image quality of visualisation based on it. These semantic representations are suited for certain purposes. FOPC suits query-answering, especially a true or false judgement; event-logic truth conditions are suitable for motion recognition; x-schemas with f-structs suit both verb recognition and performing the action but require training.

3.2 Multimodal semantic representations

Multimodal interpretation, realisation and integration in intelligent multimedia systems have general requirements for multimodal semantic representations: they should support both interpretation and generation, support many kinds of multimodal input and output, and support a variety of semantic theories. A multimodal representation may contain architectural, environmental, and interaction information. Architectural representation indicates producer/consumer of the information, confidence, and devices. Environmental representation indicates timestamps, spatial information (e.g. speaker's position or graphical configurations). Interaction representation indicates speaker/user's state or other addressees.

Frame-based and XML representations are the most common multimodal semantic representations. They are commonly used in previous intelligent multimedia applications to represent multimodal semantics, such as CHAMELEON (Brøndsted et al. 2001), AESOPWORLD (Okada 1996), REA (Cassell et al. 2000), and WordsEye (Coyne and Sproat 2001) based on frame representations to represent semantic structure. XML (eXtensible Markup Language) as a mark-up language is also used to represent *general* semantic structure in recent multimodal systems, such as in BEAT (Cassell et al. 2001) and a derivative M3L (MultiModal Markup Language) in SmartKom (Wahlster et al. 2001).

3.2.1 Frame representation and frame-based systems

Frames were introduced by Minsky (1975) in order to represent *situations*. Frames are based on a psychological view of human memory and the basic idea is that on meeting a new problem humans select an existing frame (a remembered framework) to be adapted to fit new situations by changing appropriate details. Much like a semantic network except each node represents prototypical concepts and/or situations, in frame representation, each node has several property *slots* whose values may be specified or inherited by default. Frames are typically arranged in a taxonomic hierarchy in which each frame is linked to one parent frame. A collection of frames in one or more inheritance hierarchies is a *knowledge base*. Frame representation is *object-oriented*: all the information about a specific concept is stored with that concept, as opposed, for example, to rule-based systems where information about one concept may be scattered

throughout the rule base. Frame representation provides a natural way to group concepts in hierarchies in which higher level concepts represent more general, shared attributes of the concepts below. Frames also provide a convenient method for *reasoning*, i.e. the ability to state in a formal way that the existence of some piece of knowledge implies the existence of some other, previously unknown piece of knowledge, and for *classification*, i.e. given an abstract description of a concept, determine if a concept fits that description, which is actually a common special form of reasoning.

Many knowledge representation languages have been developed based on frames. The KL-ONE (Brachman and Schmolze 1985) and KRL (Bobrow and Winograd 1985) languages were influential efforts representing knowledge for natural language processing purposes. Recent intelligent multimodal systems which use frame representations for multimodal interaction are CHAMELEON (Brøndsted et al. 2001), WordsEye (Coyne and Sproat 2001), AESOPWORLD (Okada 1996), and REA (Cassell et al. 2000).

However, frame-based systems are limited when dealing with *procedural knowledge*. An example of procedural knowledge would be calculating gravitation, i.e. the attraction between two masses is inversely proportional to the square of their distances from each other. Given two frames representing the two bodies, with slots holding their positions and mass, the value of the gravitational attraction between them cannot be inferred declaratively using the standard reasoning mechanisms available in frame-based languages, though a function or procedure in any programming language can represent the mechanism for performing this inference. Frame-based systems that can deal with this kind of knowledge by adding a procedural language to the representation, and this knowledge is not represented in a frame-based way.

3.2.2 XML representations

XML (eXtensible Markup Language) specification was published as a W3C (World Wide Web Consortium) recommendation (W3C 2002). As a restricted form of SGML (the Standard Generalized Markup Language), XML meets the requirements of large-scale web content providers for industry-specific markup, data exchange, media-independent publishing, workflow management in collaborative authoring environments, and the processing of web documents by intelligent clients. XML documents are made up of *entities* which contain either parsed or unparsed data. Parsed data is either *markup* or *character data* bracketed in a pair of start and end markups. Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure.

Unlike HTML, XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the *XML processor* or *parser* that reads it. It is assumed that an XML processor is doing its work on behalf of another module, called the *application*.

Any programming language such as Java can be used to output data from any source in XML format. There is a large body of middleware written in Java and other languages for managing data either in XML or with XML output.

There is an emerging interest in combining multimodal interaction with simple natural language processing for Internet access. One approach to implementing this is to combine XHTML (eXtensible HTML, a reformulation of HTML 4.01 as an XML 1.0 application) with markup for prompts, grammars and the means to bind results to actions. XHTML defines various kinds of events, for example, when the document is loaded or unloaded, when a form field gets or loses the input focus, and when a field's value is changed. These events can be used to trigger aural prompts, and to activate recognition grammars. This would allow a welcome message to start playing when the page is loaded. When you set the focus to a given field, a prompt could be played to encourage the user to respond via speech rather than via keystrokes. There are some specific standards of XML specially designed for the purpose of multimodal access to the Internet, such as SMIL (SMIL 2005), VoiceXML (VoiceXML 2004), X+V — XHTML plus Voice (W3C Voice Architecture 2003), SALT (SALT 2002), MultiModal Interface Language (MMIL) (Romary and Bunt 2002), Extensible MultiModal Annotation markup language (EMMA) (EMMA W3C Working Draft 2005), Java Speech API Markup Language (JSML 2005), and VHML (Gustavsson et al. 2001). Appendix A shows an example of VHML (XML-based language) tagged text of a virtual storyteller. JSML provides a speech synthesiser with detailed information on how to speak text and thus enable improvements in the quality, naturalness and understandability of synthesised speech output. JSML defines elements that indicate phrasing, emphasis, pitch and speaking rate, and control other important speech characteristics.

Due to its advantages of being media-independent, understandable and with wide coverage, XML-based representation is becoming more popular in multimodal systems. SmartKom (Wahlster et al. 2001) is a multimodal communication kiosk for airports, train stations, or other public places where people may seek tourist information. It can understand speech combined with video-based recognition of gestures and facial expressions. SmartKom develops an XML-based mark-up language called M3L (MultiModal Markup Language) for the representation of all of the information that flows between the various processing components. BEAT (Cassell et al. 2000, 2001) also uses XML for its knowledge representation. Besides multimodal presentation systems, XML representation is common in natural language processing applications and annotated corpora, such as the Gate natural language processing platform (Cunningham et al. 2002), Connexor Machine parser (Connexor 2003), and SemCor (Mihalcea 2003).

3.2.3 Summary of knowledge representations

There are several general knowledge representation languages which have been implemented in artificial intelligence applications: rule-based representation, First Order Predicate Calculus (FOPC), semantic networks, CD, and frames. FOPC and frames have historically been the principal methods used to investigate semantic issues. After first order logic and frame representation, artificial intelligence generally breaks down common sense knowledge representation and reasoning into the two broad categories of physics, including spatial and temporal reasoning, and psychology, including knowledge, belief, and planning, although the two are not completely independent. Planning intended actions, for example, requires the ability to reason about time and space. We are interested here in the physical aspects of knowledge representing and reasoning.

Recent semantic representation and reasoning on physical aspects such as representation of simple action verbs (e.g. push, drop) includes event-logic (Siskind 1995) and x-schemas with f-structs (Bailey et al. 1997). Many natural language and vision processing integration applications are developed based on the physical semantic representations which focus most on visual semantic representation of verbs — the most important category for dynamic visualisation. Narayanan's language visualisation system (Narayanan et al. 1995) is based on CD, ABIGAIL (Siskind 1995) is based on event-logic truth conditions, and L_0 (Bailey et al. 1997) is based on x-schemas and f-structures.

Table 3.2 shows categories of major knowledge representations we have discussed and their typical suitable applications. We distinguish general and physical knowledge representations. General knowledge representations include rule-based representation, FOPC, semantic networks, frames and XML. Typically, rule-based representation and FOPC are used in expert systems; semantic networks are used to represent lexical semantics; frames and XML are commonly used to represent multimodal semantics in intelligent multimedia systems. Physical knowledge representation and reasoning includes Schank's CD, event-logic truth conditions, and x-schemas. All of them could be used to represent visual semantics in movement recognition or generation applications.

Figure 3.5 illustrates the relationship between multimodal semantic representations and visual semantic representations. Multimodal semantic representations are media-independent and are usually used for media fusion and coordination; visual semantic representations are media-dependent (visual) and are typically used for media realisation.

3.3 Temporal relations

A common problem in the tasks of both language visualisation and vision recognition is to represent visual semantics of actions and events, which happen in both space and *time continuum*. It requires a facility to represent temporal relationships in visual semantics of events. Allen (1983) introduced thirteen basic interval-based temporal relations as listed in

Table 3.3. The thirteen binary relations in the table represent the relationship of “before”, “after”, “meets”, “met by”, “overlaps”, “overlapped by”, “during”, “contains”, “starts”, “started by”, “finishes”, “finished by” and “equals”. In Table 3.3, subscript “e” denotes “end point”, and “s” denotes “start point”.

<i>Categories</i>	<i>Knowledge representations</i>	<i>Typical applications</i>
(1) general knowledge representation & reasoning	rule-based representation	expert systems
	FOPC (First Order Predicate Calculus)	sentence representation, expert systems
	semantic networks	lexical semantics
	Schank’s scripts	story understanding
	frame-based representations XML-based representations	multimodal semantics
(2) physical knowledge representation & reasoning (including spatial/temporal reasoning)	Conceptual Dependency (CD)	dynamic vision (movement) recognition & generation
	event-logic truth conditions	
	x-schema and f-structure	
	Lexical-Conceptual Structure (LCS)	

Table 3.2: Categories of knowledge representations

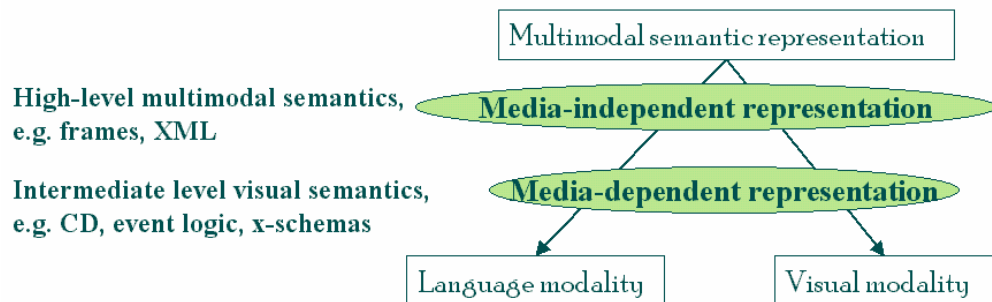


Figure 3.5: Multimodal semantic representations and visual semantic representations

Allen’s interval relations have been employed in story-based interactive systems (Pinhanez et al. 1997) to express progression of time in virtual characters and handling linear/parallel events in story scripts and user interactions. In their stories, interval logic is used to describe the relationships between the time intervals which command actuators or gather information from sensors, which in turn decides the storyline. There are three types of interaction pattern in their interactive systems: linear, reactive, and tree-like. In reactive patterns, a story unfolds as a result of the firing of behaviours as a response to users’ actions; in tree-like patterns, the user chooses between different paths in the story through some selective action. Linear, reactive, and tree-like interaction patterns can be modelled with interval logic.

The x-schema model (Bailey et al. 1997) represents the aspectual semantics of events via a kind of probabilistic automaton called *Petri Nets*. The nets used in the model have states like *ready*, *process*, *finish*, *suspend*, and *result*. For example, the meaning representation of “Jack is walking to the store” activates the *process* state of the walking event. An

accomplishment event like “Jack walked to the store” activates the *result* state. An iterative activity like “Jack walked to the store every week” is simulated in the model by an iterative activation of the *process* and *result* nodes.

<i>Basic relations</i>		<i>Example</i>	<i>Endpoints</i>	<i>Sentences</i>
precede	$x p y$	xxxx	$x_e < y_s$	John left before Mary arrived.
inverse precede	$y p^{-1} x$	yyyy		
meet	$x m y$	xxxxx	$x_e = y_s$	All passengers died when the plane crashed into the mountain.
inverse meet	$y m^{-1} x$	yyyyy		
overlap	$x o y$	xxxxx	$x_s < y_s < x_e \cap$	Mary got up. She felt very ill.
inverse overlap	$y o^{-1} x$	yyyyy	$x_e < y_e$	
during	$x d y$	xxxx	$x_s > y_s \cap$	John arrived in Boston last Thursday.
inverse during (include)	$y d^{-1} x$	yyyyyyyyy	$x_e < y_e$	
start	$x s y$	xxxxx	$x_s = y_s \cap$	John has lived in Boston since 2000.
inverse start	$y s^{-1} x$	yyyyyyyyy	$x_e < y_e$	
finish	$x f y$	xxx	$x_e = y_e \cap$	John stayed in Boston till 2000.
inverse finish	$y f^{-1} x$	yyyyyyyyy	$x_s > y_s$	
equal	$x \equiv y$	xxxxx	$x_s = y_s \cap$	John drove to London. During his drive he listened Classic FM.
	$y \equiv x$	yyyyy	$x_e = y_e$	

Table 3.3: Allen’s thirteen interval relations

On sentence level (or post-lexical level) temporal analysis within natural language understanding, there are extensive discussions on tense, aspect, duration and iteration, involving *event time*, *speech time*, and *reference time* (Reichenbach 1947). To represent the relations among them, some use point-based metric formalisms (e.g. van Benthem 1983), some use interval-based logic (e.g. Halpern and Shoham 1991), others integrate interval-based and point-based temporal logic (Kautz and Ladkin 1991) because of the complexity of temporal relations in various situations, for example, the distinction between punctual events and protracted events, achievements and accomplishments (Smith 1991, Vendler 1967), stative verbs and eventive verbs, states, events and activities (Allen and Ferguson 1994). However, few of these are concerned with the temporal relations at the lexical level, e.g. between or within verbs. In lexical semantics, extensive studies have been conducted on the semantic relationships of verbs (Fellbaum 1998), but few temporal relations have been considered. The only work that considers temporal relations on the lexical level was conducted by Badler et al. (1997). They generalized five possible temporal relationships between two actions in the technical orders (instruction manuals) domain. The five temporal constraints are *sequential*, *parallel*, *jointly parallel* (the actions are performed in parallel and no other actions are performed until after both have finished), *independently parallel* (the actions are performed in parallel but once one of the actions is finished, the other one is stopped), and *while parallel* (the subordinate action is

performed while the dominant action is performed; once the dominant action finishes, the subordinate action is stopped).

3.3.1 Punctual events

Verbs can describe states and events. Events can be punctual or last for a period of time. Vendler (1967) categorises verbs according to aspectual classes. Vendler's verb classes (1-4) emerge from an attempt to characterize a number of patterns in aspectual data:

1. activities: run, swim, think, sleep, cry
2. statives: love, hate, know
3. achievements: arrive, win, find
4. accomplishments: build (a house), write (a book)

Following Vendler, Stede (1996) presents the ontology of his machine translation system MOOSE as Figure 3.6. 5-10 list examples of each category. Stede's EVENTS have internal structure, i.e. their results are included in them (8-10), and therefore involve change of state.

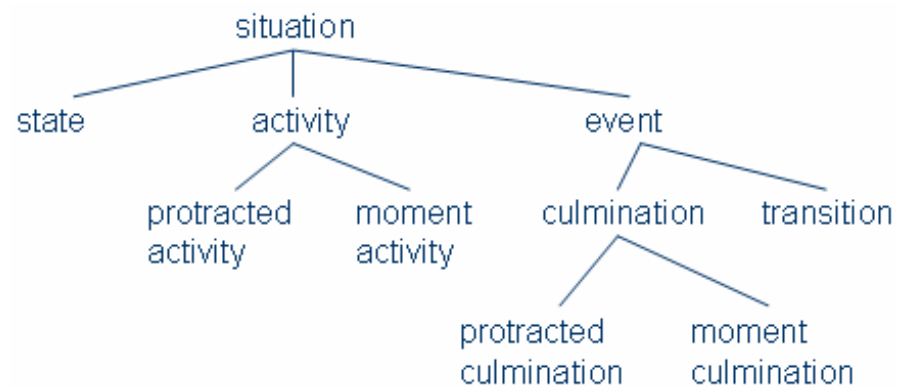


Figure 3.6: The ontology of MOOSE

5. state: love, hate, know
6. protracted activities: run, sleep, read
7. moment activities: knock (the door)
8. protracted culmination: build (a house), write (a book)
9. moment culmination: arrive, win, find
10. transition: (the room) lit up

There is a group of verbs indicating punctual events which never hold over overlapping intervals, or two intervals one of which is a subinterval of the other, such as “find”, “arrive”, and “die”. Vendler (1967) classified them as *achievement* events (distinct from *accomplishment* events), which occur at a single moment and involve unique and definite time instants. Dowty (1979) draws attention to a major difference between achievements and accomplishments: accomplishment verbs are telic, describing activities that normally lead to a result. Smith (1991) similarly proposes that achievements are *instantaneous events* that result in a *change of state*. It

seems that point-based relations are more appropriate for these verbs. However, pragmatic, ontological, and practical cases for interval relations have been advocated. Some pragmatic approaches (Verkuyl 1993) deny the semantic distinction between accomplishments and achievements. They hold that the length of the event is not a linguistic matter. Jackendoff (1991) and Pinon (1997) introduce the concept of *boundaries* into a temporal ontology for aspectual semantics to analogise achievement events. Boundaries are *ontologically dependent* objects: they require the existence of that to which they are bound.

3.3.2 Verb entailment and troponymy

Verb entailment is a fixed truth relation between verbs where entailment is given by part of the lexical meaning, i.e. entailed meaning is in some sense contained in the entailing meaning. Verb entailment indicates an *implication* logic relationship: “if x then y” ($x \Rightarrow y$). Take the two pairs *snore-sleep* and *buy-pay* as example, we can infer $\text{snore} \Rightarrow \text{sleep}$ and $\text{buy} \Rightarrow \text{pay}$ since when one is snoring (s)he must be sleeping, and if somebody wants to buy something (s)he must pay for it, whilst we cannot infer in the reverse direction because one may not snore when (s)he is sleeping, and one might pay for nothing (not buying, such as donation). In these two examples, the entailing activity could temporally *include* (i.e. d^{-1}) or *be included in* (i.e. d) the entailed activity. Fellbaum (1998) classifies verb entailment relations into four kinds, based on temporal inclusion, backward presupposition (e.g. the activity *hit/miss* supposes the activity *aim* occurring in a previous time interval) and causal structure (Figure 3.7).

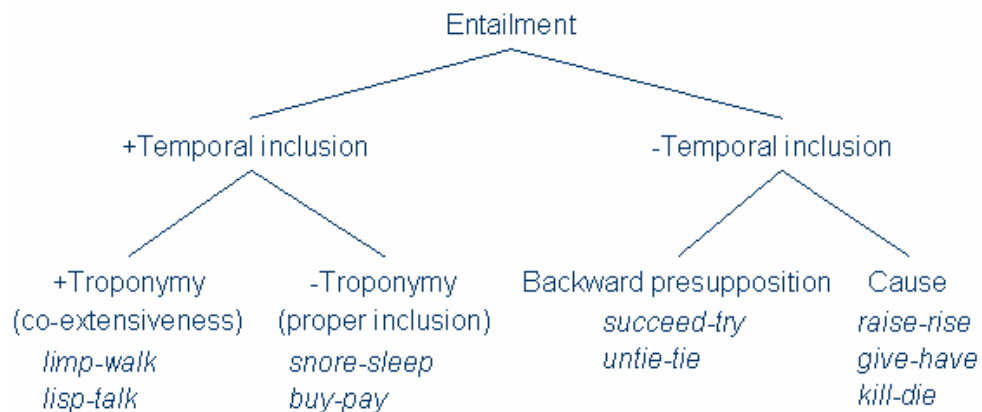


Figure 3.7: Fellbaum’s classification of verb entailment

Troponymy is an important semantic relation in verb entailment which typically holds between manner elaboration verbs and their corresponding base verbs, i.e. two verbs have the troponym relation if one verb elaborates the manner of another (base) verb. For instance, *mumble-talk* indistinctly, *trot-walk* fast, *stroll-walk* leisurely, *stumble-walk* unsteadily, *gulp-eat* quickly, the relation between *mumble* and *talk*, *trot/stroll/stumble* and *walk*, *gulp* and *eat* is troponymy.

3.4 Computational lexicons

Many problems in Natural Language Processing (NLP), especially disambiguation, resort to lexical resources. In the last decade, there have been advances in lexical knowledge on how to create, represent, organise, categorise, and access large computational lexicons, such as WordNet (Fellbaum 1998), FrameNet (Baker et al. 1998), LCS database (Dorr & Jones 1999), and VerbNet (Kipper et al. 2000), especially for verbs, and the relation between the syntactic realisation of a verb's arguments and its meaning has been extensively studied in Levin's (1993) classes. In this section, existing computational lexicons are analysed and compared.

3.4.1 WordNet

WordNet (Fellbaum 1998), one of the most widely used lexical resources, is a relational and taxonomic semantic network modelling the lexical knowledge of English. It incorporates information on lexicalisation patterns, semantic components and conceptual inferences. WordNet divides the lexicon into five categories: nouns, verbs, adjectives, adverbs, and function words. Lexical information is organised in terms of semantic relations between words. WordNet uses semantic networks (*synsets*²) representing some major semantic clusters per part-of-speech. The relations used in WordNet include synonymy, autonymy, hyperonymy, hyponymy, holonymy, meronymy, troponymy (entailment), cause, value_of, attributes (has_value), and derivationally related form. Figure 3.8 lists the semantic relations distinguished between synsets.

Hypernyms are synsets which are the more general class of a synset, e.g. {noun.artifact} => {noun.object}. Hyponyms are synsets which are particular kinds of a synset, e.g. {weather, atmospheric condition} => {sunshine}, {noun.object} => {noun.artifact}. Using these two relations one can trace the word 'person' along the edges between nodes in the semantic network:

```

person => human being => hominid => primate => placental =>
mammal => vertebrate => chordate => animal => organism =>
animate thing => object => entity

```

up to a noun top *entity*, one root of the major semantic clusters of nouns in WordNet. There are some *hypernym* or *hyponym* relations between some synsets which result in categories overlapping.

In terms of adjective synsets, Gross and Miller (1990) proposed that adjective synsets be regarded as clusters of adjectives associated by semantic similarity to a focal adjective that relates the cluster to a contrasting cluster at the opposite pole of the attribute as shown in Figure 3.9. Through a synonymic adjective, adjectives which seem to have no appropriate antonyms

² Set of synonymous word meanings (synset members).

can find their antonyms. For instance, “soggy” is similar to “wet” and “wet” is the antonym of “dry”, so a conceptual opposition of “soggy”-“dry” is mediated by “wet”.

Synonyms: members of the synset which are equal or very close in meaning.
Antonyms: synsets which are opposite in meaning
Hypernyms: synsets which are the more general class of a synset, e.g. {glass (sense 2)} ==> {container}
Hyponyms: synsets which are particular kinds of a synset, e.g. {weather} ==> {fair, sunshine, temperateness}
Holonyms: synsets which are the whole of which a synset is a part. [Part of] e.g., {flower, bloom, blossom} PART OF {angiosperm, flowering plant}
 [Member of] e.g., {homo, man, human being, human} MEMBER OF {genus Homo}
 [Substance of] e.g., {glass} SUBSTANCE OF {glassware, glasswork}
Meronyms: synsets which are the parts of a synset.
 [Has Part] e.g. {flower, bloom, blossom} HAS PART {stamen}, {pistil}, {carpel}, {ovary}, {floral leaf}
 [Has Member] e.g. {womankind} HAS MEMBER {womanhood, woman}
 [Has Substance] {glassware, glasswork} HAS SUBSTANCE {glass}
Entailments: synsets which are entailed by the synset, e.g. {walk, go on foot, foot, leg it, hoof} ==> {step, take a step}
Causes: synsets which are caused by the synset, e.g. {kill} ==> {die, pip out, decease, perish, go, exit, pass away, expire}
Value of: (adjectival) synsets which represent a value for a (nominal) target concept. e.g. poor VALUE OF {financial condition, economic condition}
Has Value: (nominal) synsets which have (adjectival) concept as values, e.g. {size} ==> {large, big}
Similar to: Peripheral or Satellite adjective synset linked to the most central (adjectival) synset, e.g. {damp, dampish, moist} SIMILAR TO {wet}
Derived from: Morphological derivation relation with a synset, e.g. {coldly, in cold blood, without emotion} Derived from adj ==> {cold}

Figure 3.8: Basic relations between synsets in WordNet

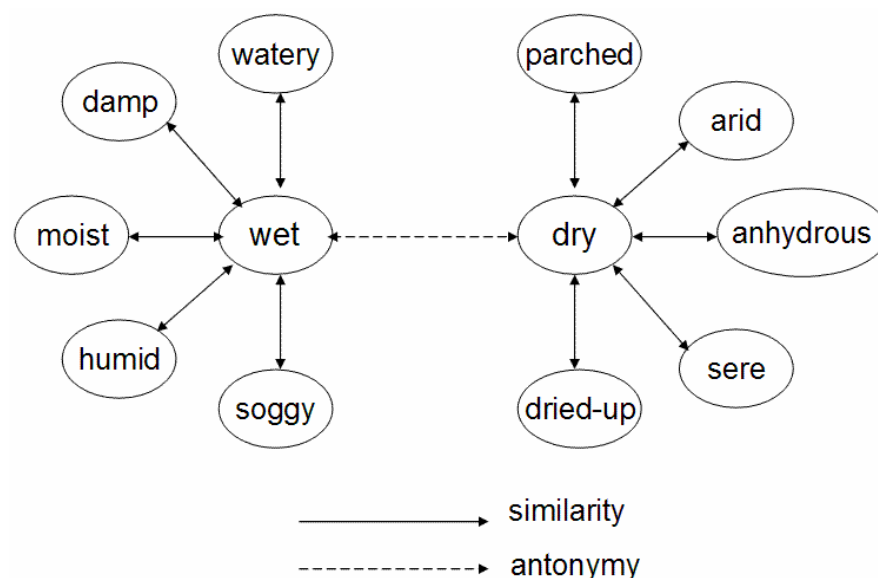


Figure 3.9: Bipolar adjective structure in WordNet (Gross and Miller 1990)

3.4.2 FrameNet

FrameNet (Baker et al. 1998) is a corpus-based computational lexicon based on the British National Corpus (BNC). It contains descriptions of the semantic frames underlying the meanings of words and the representation of the valences of words in which the semantic portion makes use of frame semantics.

Unlike WordNet which provides a framework to organise all of the concepts we use to describe the world, aiming to cover every possible subject area with at least a low level of detail, the semantic domains covered in FrameNet are limited: health care, chance, perception, communication, transaction, time, space, body, motion, life stages, social context, emotion and cognition.

FrameNet is somehow similar to efforts to describe the argument structures of lexical items in terms of case roles or theta roles, but the definition of frame in FrameNet is different from others, to wit, FrameNet's frames are rather semantic categories. In FrameNet, the role names, called Frame Elements (FEs), are local to particular conceptual structures (called frames in FrameNet); some FEs are general, while others are specific to a small family of lexical items, for instance, the motion frame has theme, path, source, goal, area FEs, the activity frame has the agent FE, whereas the experience frame has experiencer and content FEs.

Default arguments and shadow arguments such as instrument, means, and purpose are peripheral FEs, and not specified in FrameNet, e.g. there are three word senses of "drive" in FrameNet's semantic domains as listed in the Table 3.4. The verb "drive" implies that the value of the argument instrument/means is a hyponym of vehicle. Although the knowledge is listed as a FE in the frame operate_vehicle, it is hard to access this information since a way to distinguish between these three frames is not provided.

<i>Entry</i>	<i>Frame</i>	<i>FEs</i>
drive.v.	Operate_vehicle	Area, Driver, Path Goal, Source, <i>Vehicle</i>
drive.v.	Self_motion	Area, Goal, Source, Path, Self_mover
drive.v.	Carrying	Agent, Area, Carrier Path, Theme, Path_end, Path_start

Table 3.4: Frames and FEs of "drive" in FrameNet

Therefore, FrameNet has two limitations for language-to-vision applications: (1) its semantic domains are limited, (2) default arguments are either not contained or inaccessible.

3.4.3 The LCS database and VerbNet

LCS database (Dorr & Jones 1999) and VerbNet (Kipper et al. 2000) are verb lexicons. In LCS database, verbs (approximately 9000) are organised into semantic classes and each class is represented with Jackendoff's LCS. LCS database defines the relationship between semantic

classes (based on Levin’s (1993) verb classes) and LCS meaning components. The LCS database contains verb semantic classification, WordNet’s mapping, theta roles, and LCS representation of verbs. In a typical verb entry of the LCS database shown in Figure 3.10, “:” is the delimiter of fields, CLASS refers to Levin’s verb classes, and WN_SENSE is WordNet verb sense. Besides LCS representation and variables’ specification (VAR_SPEC), a verb entry also comprises PropBank (Kingsbury & Palmer 2002) argument frames and theta roles. An underscore before a role name in the THETA_ROLES field means this role is obligatory, and comma before a role name means this role is optional. The preposition in parentheses following a role name means that this role must follow the specified preposition.

LCS representation is composed of logical arguments, including AG (agent), EXP (experiencer), TH (theme), SRC (source), GOAL, INFO, PERC (perception), PRED (predicate), LOC (location), POSS (possession), TIME, and PROP (proposition), and logical modifiers including MOD-POSS (possessional modifier), BEN (beneficiary), INSTR (instrument), PURP (purpose), MOD-LOC, MANNER, and MOD-PROP. Comparing to the above comprehensive lexicons (not only verb lexicons), LCS database does contain lexical knowledge in its selectional restrictions (variables specification), e.g. the agent of cut is specified as an animate being (VAR_SPEC ((1 (animate +)))) in Figure 3.10.

```
(
:DEF_WORD "cut"
:CLASS "21.1.c"
:WN_SENSE (("1.5" 00894185) ("1.6" 01069335))
:PROPBANK ("arg0 arg1 argm-LOC(in/on-up.) arg2(with)")
:THETA_ROLES ((1 "_ag_th,mod-loc(), instr(with)")
: LCS (act_on loc (* thing 1) (* thing 2)
      ((* [on] 23) loc (*head*) (thing 24))
      ((* with 19) instr (*head*) (thing 20)) (cut+ingly 26))
:VAR_SPEC ((1 (animate +)))
)
```

Figure 3.10: A verb entry of “cut” in LCS database

VerbNet is also a class-based verb lexicon based on Levin’s classes. It has explicitly stated syntactic and semantic information. The syntactic frames for the verb classes are represented by a Lexicalised Tree Adjoining Grammar augmented with semantic predicates, which allows for a compositional interpretation. In the verb entry of “cut” shown in Figure 3.11, thematic roles specify the selectional restrictions for each role like the VAR_SPEC in LCS database, e.g. [+concrete] for the instrument of cut. Some verb senses may have more specific selectional restrictions, the verb “kick” (in the verb class hit-18.1) has the following specification:

```
Instrument[+body_part OR +refl]
Instrument[+concrete]
```

It states the instrument of kicking should be either a concrete thing or a body part.

Both LCS database and VerbNet have some form of selectional restrictions which contain lexical knowledge such as default arguments. Nevertheless, these specifications are still not enough for the language visualisation task.

3.4.4 Comparison of lexicons

Table 3.5 presents a comparison showing features of lexical knowledge contained in the above-mentioned computational lexicons. WordNet does not have enough knowledge for compositional information of verbs, default instrument and functional information, which could be complemented by LCS database and VerbNet. However, as was mentioned earlier the selection restrictions of the instrument argument in both lexicons are insufficient for language-to-vision applications. We have to look for other sources for this knowledge.

Verb Class: cut-21.1-1
WordNet Senses: cut(1 24 25 31 33)
Thematic Roles:
Agent[+int_control]
Instrument[+concrete]
Patient[+body_part OR +refl]
Patient[+concrete]
Frames:
Basic Transitive
"Carol cut the bread"
Agent V Patient
cause(Agent,E) manner(during(E),Motion,Agent) contact (during (E), ?Instrument, Patient) degradation_material_integrity (result (E), Patient)
(other frames)...
Verbs in same (sub)class:
[chip, clip, cut, hack, hew, saw, scrape, scratch, slash, snip]

Figure 3.11: A verb entry of “cut” in VerbNet

<i>Lexicons</i>	<i>WordNet</i>	<i>FrameNet</i>	<i>LCS database</i>	<i>VerbNet</i>
Semantic domains	all	limited	all	all
POS	all	all	verb	verb
Hypernymy (is_a)	+	+	+	+
Hyponymy (n.) troponymy (v.)	+	+	-	-
Metonymy constructive (n.) compositional (v.)	+ (n.) - cause (v.)	-	+ Conceptual structure	+ decompose with time functions
Instrument	-	-	? Selection restrictions	? Selection restrictions
Functional information (telic role)	-	+ used_by	n/a	n/a

Table 3.5: Comparison of verb lexicons

3.4.5 Generative lexicon

The generative lexicon presented by Pustejovsky (1995) contains a considerable amount of information that is sometimes regarded as common sense knowledge. A generative lexicon has four levels of semantic representations: *argument structure*, *event structure*, *qualia structure*, and *lexical inheritance* from the global lexical structure.

The argument structure includes *true arguments* (obligatory parameters expressed as syntax), *default arguments* (parameters which are necessary for the logical well-formedness of a sentence but may not be expressed in the surface syntax), *shadow arguments* (semantic content which is not necessarily expressed in syntax and can only be expressed under specific conditions, e.g. “Mary buttered her toast *with butter”), and *adjuncts*. Qualia structure represents the different modes of predication possible with a lexical item. It is made up of *formal*, *constitutive*, *telic* and *agentive* roles. Telic roles are the function of an object or aim of an activity.

The default/shadow arguments and telic roles in a generative lexicon can complement WordNet with regard to instrument and functional information (see Table 3.5). Previous research in language-to-vision also proves the necessity of such information in the lexicon. In PAR (Badler 1997), to animate a virtual human to “walk to the door and turn the handle slowly”, the representation of the “handle” object lists the actions that the object can perform, which are called telic roles in the generative lexicon theory. WordsEye (Coyne & Sproat 2001) relies on the telic roles (functional properties) of objects to make semantic interpretations, e.g. implicit instruments, as well. To visually depict the action “ride”, it looks for objects whose functional properties are compatible with the verb to find an implied instrument “bicycle”. The theory of generative lexicon shows its adequacy to fill underspecified roles. Hence, using a generative lexicon to make inferences on given sentences is potentially useful for language-to-vision applications where it is necessary to infer as much as possible from the given sentences.

3.5 Language ontology

Language ontology is conceptual modelling in linguistics. In this section we investigate previous language ontologies and discuss how the lexicon is reflected in top-level distinctions and how functional/grammatical categories are reflected in the ontology.

3.5.1 Top concepts

The hierarchy shown in Figure 3.12 lists the top concepts in the EuroWordNet project (Vossen et al. 1998). The first level of the Top Ontology is divided into three types:

- 1stOrderEntities roughly correspond to concrete, observable physical objects, persons, animals and physical substances. They can be located at any point in time and in a 3D space.

- 2ndOrderEntities are processes, states, situations and events that can be located in time. Whereas 1stOrderEntities *exist* in time and space 2ndOrderEntities *occur* or *take place*, rather than exist.
- 3rdOrderEntities are mental entities such as ideas, concepts and thoughts that exist outside space/time dimension and are unobservable. Furthermore, they can be predicated as true or false rather than real, they can be asserted or denied, remembered or forgotten.

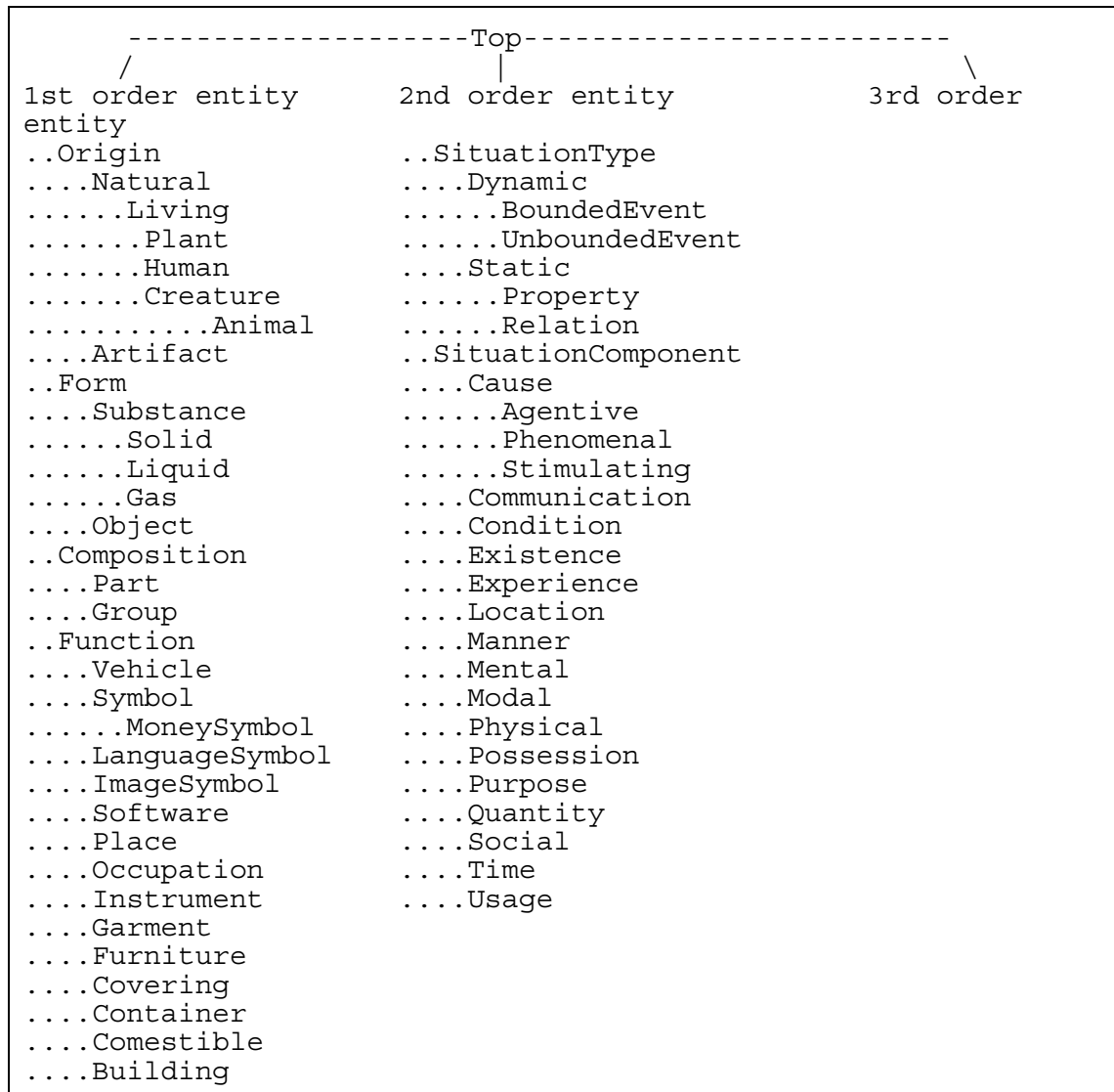


Figure 3.12: Hierarchy of top concepts in EuroWordNet (Vossen et al. 1998)

3.5.2 Ontological categories of nouns

We interpret WordNet hypernym/hyponym relationships among the noun synsets as specialization relations between conceptual categories and use it as a lexical ontology. Figure 3.13 shows the major semantic clusters of nouns in WordNet.

There may be one or more synsets in Figure 3.13 which have no hypernym and therefore represent the top of the semantic network. In the case of nouns there are only 11 tops or unique-beginners, in the case of verbs 573 tops in WordNet. Figure 3.14 lists the eleven top

noun categories. However, such an ontology should normally be corrected before being used since it contains many basic semantic inconsistencies such as the existence of common specializations for exclusive categories and redundancies in the specialization hierarchy.

```
noun.act - nouns denoting acts or actions
noun.animal - nouns denoting animals
noun.artifact - nouns denoting man-made objects
noun.attribute - nouns denoting attributes of people and objects
noun.body - nouns denoting body parts
noun.cognition - nouns denoting cognitive processes and contents
noun.communication - nouns denoting communicative processes and
    contents
noun.event - nouns denoting natural events
noun.feeling - nouns denoting feelings and emotions
noun.food - nouns denoting foods and drinks
noun.group - nouns denoting groupings of people or objects
noun.location - nouns denoting spatial position
noun.motive - nouns denoting goals
noun.object - nouns denoting natural objects (not man-made)
noun.person - nouns denoting people
noun.phenomenon - nouns denoting natural phenomena
noun.plant - nouns denoting plants
noun.possession - nouns denoting possession and transfer of possession
noun.process - nouns denoting natural processes
noun.quantity - nouns denoting quantities and units of measure
noun.relation - nouns denoting relations between people or things or
    ideas
noun.shape - nouns denoting two and three dimensional shapes
noun.state - nouns denoting stable states of affairs
noun.substance - nouns denoting substances
noun.time - nouns denoting time and temporal relations
```

Figure 3.13: Major semantic clusters of nouns in WordNet

1. entity - something having concrete existence; living or nonliving
2. psychological feature - a feature of the mental life of a living organism
3. abstraction - a concept formed by extracting common features from examples
4. location, space - a point or extent in space
5. shape, form - the spatial arrangement of something as distinct from its substance
6. state - the way something is with respect to its main attributes; 'the current state of knowledge'; 'his state of health'; 'in a weak financial state'
7. event - something that happens at a given place and time
8. act, humanaction, humanactivity - something that people do or cause to happen
9. group, grouping - any number of entities (members) considered as a unit
10. possession - anything owned or possessed
11. phenomenon - any state or process known through the senses rather than by intuition or reasoning

Figure 3.14: Noun tops in WordNet

3.5.3 Ontological categories of verbs

Since verbs are core to events, verb subcategories are significant for visualisation of events. The classification of verbs and their semantic properties has been the topic of numerous philosophical and linguistic studies. Both traditional grammars subcategorising verbs into transitive and intransitive, and modern grammars distinguishing as many as 100 subcategories — tagsets such as the COMLEX tagset (Macleod et al. 1998) and the ACQUILEX tagset (Sanfilippo 1993), classify verbs according to *subcategorisation frame*, i.e. possible sets of complements the verbs expect (see Table 3.6). For instance, a verb like “find” subcategorises for an NP, whereas a verb like “want” subcategorises for either an NP or a non-finite VP. These possible sets of complements of a verb are also called the *subcategorisation frame* for the verb.

In 1980s, the Longman Dictionary of Contemporary English (LDOCE) was the most comprehensive computational lexicon with a description of grammatical properties of words. It had a very detailed word-class categorisation scheme, particularly for verbs. In addition to part-of-speech information LDOCE specifies a subcategorisation description in terms of types and numbers of complements for each entry. In LDOCE grammar codes separate verbs into the categories: e.g. D (ditransitive), I (intransitive), L (linking verb with complement), T1 (transitive verb with an NP object), T3 (transitive verb with an infinitival clause as object). These grammar codes implicitly express verb subcategorisation information including specifications on the syntactic realisation of verb complements and argument functional roles.

<i>Sub-categorisation</i>	<i>Verb</i>	<i>Examples</i>
∅	eat, sleep	I want to eat
NP	prefer, find, leave, want	find [_{NP} the flight from New York to Boston]
NP NP	show, give	show [_{NP} me] [_{NP} airlines with flights from New York]
PP _{from} PP _{to}	fly, travel	I would like to fly [_{PP} from New York] [_{PP} to Boston].
VP _{to}	prefer, want, need	I want [_{VPto} to have a pint of beer].
VP _{bareStem}	can, would, might	I can [_{VPbareStem} swim]
S	mean, say, think, believe	He said [_S the Government disagreed with her account].

Table 3.6: Some linguistic subcategorisation frames and example verbs

The notion of valency is borrowed from chemistry to describe a verb’s property of requiring certain arguments in a sentence. Valency fillers can be both obligatory (*complements*) and optional (*adjuncts*): the former are central participants in the process denoted by the verb, the latter express the associated temporal, locational, and other circumstances. Verbs can be divided into classes based on their valency.

There are different opinions on the type of a verb’s valency fillers. Leech (1981) raises the idea of *semantic valency* to operate on a level different from surface syntax. Semantic valency was further developed into the theory of thematic roles in terms of which semantic role

each complement in a verb's argument structure plays, ranging from Fillmore's (1968) case grammar to Jackendoff's (1990) Lexical Conceptual Structure (LCS). The term *thematic role*, also known as theta-role, case role, deep grammatical function, transitivity role, and valency role, covers a layer in linguistic analysis. The idea is to extend syntactic analysis beyond surface case (nominative, accusative) and surface function (subject, object) into the semantic domain in order to capture the roles of participants. The classic roles are *agent*, *patient (theme)*, *instrument*, and a set of locational and temporal roles like *source*, *goal* and *place*.

Having a set of thematic roles for each verb type, Dixon (1991) classifies verbs into 50 verb types, each of which has one to five thematic roles that are distinct to that verb type. Systemic Functional Grammar (Halliday 1985) works with 14 thematic roles divided over 5 *process types* (verb types). Some linguists work out a minimal thematic role system of three highly abstract roles (for valency-governed arguments) on the grounds that the valency of verbs never exceeds 3. Dowty (1991) assumes that there are only two *thematic proto-roles* for verbal predicates: the *proto-agent* and *proto-patient*. Proto-roles are conceived of as *cluster-concepts* which are determined for each choice of predicate with respect to a given set of semantic properties. Proto-agent involves properties of volition, sentience/perception, causes event, and movement; proto-patient involves change of state, incremental theme, causally affected by event, and stationary (relative to movement of proto-agent).

The ontological categories proposed by Vendler (1967) are dependent on aspectual classes. Vendler's verb classes (activities, statives, achievements, and accomplishments) emerge from an attempt to characterize a number of patterns in aspectual data. Formal ontologies such as DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) (Oltramari et al. 2002), SUMO (Suggested Upper Merged Ontology) (Pease et al. 2002) and CYC (Lenat 1995) all assume the traditional aspectual (temporal) classification for their events (processes).

Semantic Perspective: WordNet and Dimension of Causation

The verb hierarchical tree in WordNet (Fellbaum 1998) represents another taxonomic approach based on pure lexical semantics. It reveals the semantic organisation of the lexicon in terms of lexical and semantic relations. Table 3.7 lists the lexicographer files of verbs in WordNet, which shows the top nodes of the verb trees.

Asher and Lascarides (1995) put forward another lexical classification based on the dimension of causal structure. They assume that both causation and change can be specified along the following four dimensions so as to yield a thematic hierarchy such as the one described in the lattice structure in Figure 3.15.

- *locative*: specifying the causation of motion, e.g. "put"
- *formal*: specifying the creation and destruction of objects, e.g. "build"
- *matter*: specifying the causation of changes in shape, size, matter and colour of an object, e.g. "paint"

- *intentional*: specifying causation and change of the propositional attitudes of individuals, e.g. “amuse”, “persuade”

<i>Lexicographer files</i>	<i>Contents</i>
verb.body	grooming, dressing, bodily care
verb.change	size, temperature change, intensifying
verb.cognition	thinking, judging, analyzing, doubting
verb.communication	telling, asking, ordering, singing
verb.competition	fighting, athletic activities
verb.consumption	eating and drinking
verb.contact	touching, hitting, tying, digging
verb.creation	sewing, baking, painting, performing
verb.emotion	feeling
verb.motion	walking, flying, swimming
verb.perception	seeing, hearing, feeling
verb.possession	buying, selling, owning
verb.social	political/social activities & events
verb.stative	being, having, spatial relations
verb.weather	raining, snowing, thawing, thundering

Table 3.7: WordNet verb files

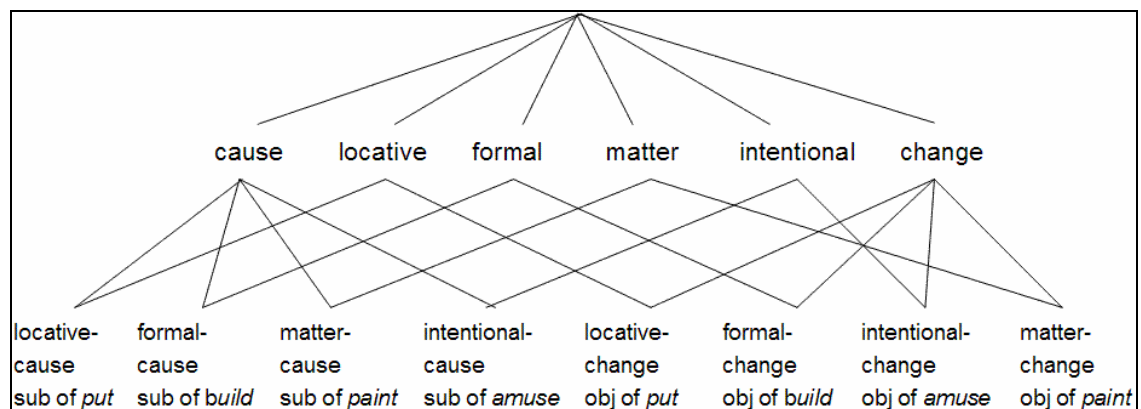


Figure 3.15: Dimension of causation-change

Semantic-Syntactic Correlations: Levin’s Verb Classes

Besides purely syntactic and purely semantic methodologies, parallel syntactic-semantic patterns in the English verb lexicon have been explored as well since it is discovered that words with similar meaning, i.e. whose LCSs (Jackendoff 1990) are identical in terms of specific meaning components, show some tendency toward displaying the same syntactic behaviours. Levin’s (1993) verb classes represent the most comprehensive description in this area. She examines a large number of verbs, classifies them according to their semantic/syntactic correlations, and shows how syntactic patterns systematically accompany the semantic classification. Levin’s verb classes are important to the compatibility of a visual semantic based verb taxonomy.

Processes classification of human activities

Halliday (1985) distinguishes six types of processes involved in natural languages which describe human actors' performance: *doing* (actor and goal, implying a change of state), *sensing* (mental processes like perception, cognition and affection), *being* (attributes like "be happy", "be sad"), *behaving* (physiological behaviours), *saying* (addresser, addressee, and verbiage), and *existing* (existent and circumstance). Halliday's classification can be instrumental for generating humanoid animation from natural language because it considers human activities rather than syntax or verb semantics.

3.6 Summary

This chapter has reviewed previous work on language and multimodal semantic representations. Knowledge representations are classified to general knowledge representations, which are usually used in expert systems and multimodal systems, and physical knowledge representations, which suit for language visualisation. Allen's interval temporal relations and their application in representing verb entailment were discussed. Computational lexicons and language ontology were also reviewed. The next chapter proposes a natural language semantic representation called Lexical Visual Semantic Representation.

Chapter 4

Lexical Visual Semantic Representation

This chapter analyses the visual ontology of concepts, especially verbs, proposes Lexical Visual Semantic Representation (LVSR), and investigates challenges which LVSR may encounter when certain linguistic phenomena are concerned.

4.1 Multimodal representation

We have discussed high-level multimodal representations such as frames and XML-based languages and low-level media dependent representations such as Virtual Reality Modelling Language (VRML) in Chapter 3, section 3.2. It is necessary to have an intermediate level semantic representation which is capable of connecting meanings across modalities. Such an intermediate level meaning representation, which links language modalities to visual modalities, is proposed in this chapter. Figure 4.1 illustrates our multimodal semantic representation. It is composed of language, visual and audio modalities. Between the multimodal semantics and each specific modality there are three levels of representation: a high-level *multimodal* semantic representation (VHML, as described in Chapter 2, section 2.2.1), an intermediate level representation which links visual and language modalities (LVSR), and low-level media-dependent representations (VRML, as described in Chapter 2, section 2.2.1, and JSML, as described in Chapter 3, section 3.2.2).

There is no dearth of language semantic representations for linguistic information. As we discussed in Chapter 3, section 3.1, common *natural language semantic representation*¹ includes FOPC, semantic networks and frames. Most semantic representations in natural language processing represent meaning at the sentence or phrase levels, and are used for purposes like question-answering, information retrieval, and image recognition. The LVSR proposed in this chapter can represent visual semantics² on the word level (e.g. action verbs) and is suited for computer graphics generation from natural language input, i.e. the intermediate level representation in Figure 4.1. LVSR can be translated to low-level representations such as

¹ Here we use “natural language semantic representation” to differentiate from “multimodal semantic representation”. Typically, without any modifier “semantic representation” means semantics for natural language processing.

² By saying “visual semantics” we don’t mean image processing of vision input. We use this term to refer to meaning of visual information in language visualisation.

VRML and JSML. The audio modality consists of speech and nonspeech auditory icons. Speech is specified by JSML that is a low-level media-dependent representation. The following sections focus on the visual semantic representation of different parts-of-speech, particularly the visual semantic representation of verbs.

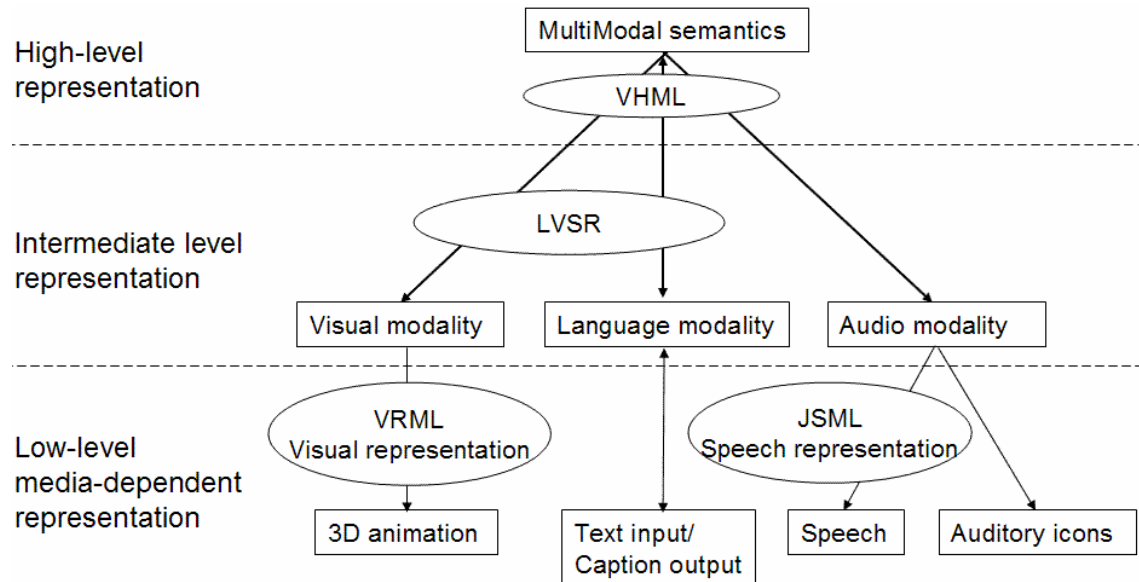


Figure 4.1: Multimodal semantic representation

4.2 Ontological categories of concepts (conceptual “parts of speech”)

Basic ontological categories of concepts are concepts on the top levels in semantic networks which make up the core of the networks. They can be grouped into coherent semantic clusters, called *top-ontology*, which is typically used to extract semantic distinctions applying to situations cutting across parts of speech, i.e. they apply to nouns, verbs and adjectives. Hence these ontological categories of concepts are so called conceptual “parts of speech”. They include THING, EVENT³, STATE, PLACE, PATH, PROPERTY and AMOUNT. Conceptual “parts of speech” are semantic categories rather than syntactic ones, though most semantic constituents correspond to syntactic constituents (e.g. THING-NP, EVENT-VP, PROPERTY-AP, PLACE/PATH-PP⁴). In addition, the matching here is by *constituents*, not by *categories* (e.g. THING matches NP instead of noun, and EVENT matches VP instead of verb), because the mapping between semantic and syntactic categories is many-to-many. An NP can express a THING (e.g. the dog), an EVENT (e.g. the war), or a PROPERTY (e.g. paleness); a PP can express a PLACE (e.g. in the house), a PATH (e.g. towards the house), or even a PROPERTY (e.g. in luck); and an S can express a STATE (e.g. John is sick) or an EVENT (e.g. John ran

³ Jackendoff’s (1990) “semantic parts of speech” include ACTION besides these seven categories. Our category EVENT here subsumes ACTION.

⁴ S = sentence, NP = noun phrase, VP = verb phrase, AP = adjective phrase, PP = prepositional phrase

away). These realisations are subject to marked conditions, in the unmarked case, NP expresses THING, S or VP express EVENT, and AP expresses PROPERTY.

There are many equivalences across parts of speech (called XPOS⁵ relationship in the WordNet lexical semantic networks, Beckwith et al. 1991) such as *beautiful* (adj.) – *beauty* (abs n.), *change* (v.) – *changing* (event n.), *adorn* (v.) – *adornment* (n.), *design* (v.) – *designer* (agent n.), *respect* (v.) – *respectful* (adj.). They play nearly the same roles in visualisation because a property or feature needs a bearer that has the property, in the *beautiful-beauty* case; and an action needs an agent who performs the action, in the *design – designer* case. This taxonomy is necessary because words from different parts of speech can be related in the semantic networks via a XPOS_SYNONYM relationship, and the entries in a graphic library can be related to any part-of-speech. Therefore, action verbs and action nouns are treated as events, and descriptive adjectives and their corresponding abstraction nouns are treated as properties, etc.

Consider the top concepts of the EuroWordNet ontology discussed in Chapter 3, section 3.5.1 from the prospective of multimodal presentation, 1stOrderEntities are suitable for presentation in static visual modalities (still pictures), 2ndOrderEntities are suitable for display in dynamic visual modalities (animation or video, accompanied with nonspeech audio as a supplement), and 3rdOrderEntities are suitable for expression in language (text/speech) since they are unobservable by visual sensors. This classification has a mapping to the linguistic concept *part-of-speech*: the 1stOrderEntities cover concrete nouns; *static situations* in the 2ndOrderEntities concern either properties of entities or relationships between entities in a 3D space, i.e. adjectives and prepositions; *dynamic situations* in the 2ndOrderEntities cover either events or their action manners, i.e. verbs; and the 3rdOrderEntities are *non-action verbs*.

4.3 Lexical Visual Semantic Representation (LVSR)

To link language with dynamic visual information, a conceptual semantic representation is required. Here, we propose a semantic representation, Lexical Visual Semantic Representation (LVSR), based on Jackendoff's (1990) LCS (see Chapter 2, section 2.5.3). LVSR is a necessary semantic representation between 3D visual information and syntactic/semantic information because 3D model differences, although crucial in distinguishing word meanings, are invisible to syntax and semantics. First, we identify nine ontological categories of concept: OBJ, HUMAN, EVENT, STATE, PLACE, PATH, PROPERTY, TIME, and AMOUNT.

LVSR distinguishes OBJ (non-animated Thing) and HUMAN (animated, articulated Thing), both of which are belong to *Thing* in LCS classification. OBJ can be props or places

⁵ For instance, in 'adorn v. XPOS_NEAR_SYNONYM adornment n.', 'adorn' is a verb and 'adornment' is a noun but they are synonyms of the same concept. The relationship between them is called XPOS_NEAR_SYNONYM (XPOS means 'across parts of speech'). XPOS relations can also be in antonyms across part-of-speech, e.g. dead n. XPOS_NEAR_ANTONYM live v.

(e.g. buildings). HUMAN can be either human being or any other articulated, animated character (e.g. animals or plants) as long as its skeleton hierarchy is defined in the graphic library. EVENTS are animation keyframes defined in the graphic library. A STATE is a static situation, which does not involve changes, and it usually refers to a fact. PLACE and PATH involve spatial relationships. A PLACE is a location, and a PATH describes the direction or course of movements. PROPERTYs are attributes of OBJs or HUMANs. AMOUNT specifies the quantity of OBJs. Moreover, LVSR adds the category TIME which is solved by adding a temporal feature to PLACE predicates in LCS. The ontological categories here are primarily for the purpose of generating humanoid character animation, and they provide a finer selection restriction facility, for example, distinguishing HUMAN from OBJ provides a finer selection restriction for verbs requiring a human agent.

Each of these categories can be elaborated into a predicate-argument form given in Figure 4.2. `placePredicate` in Figure 4.2A is usually prepositions which express spatial information, e.g. “on”, “in”, “under”. Most frequently used `movingEvent` predicates in Figure 4.2C are “go” (for both OBJ and HUMAN), “walk”, “run”, and “jump” (only for HUMAN). The `orient` predicate in Figure 4.2D can represent verbs like “point” and “face”, and the `extension` predicate represents verbs like “reach” in “the railway reaches the shore”.

```

A. [PLACE] -> [PLACE placePredicate ([OBJ])]
B. [PATH] -> [PATH pathPredicate ([OBJ/PLACE])]
C. [EVENT] -> [EVENT movingEvent ([OBJ/HUMAN], [PATH])]
               [EVENT stay ([OBJ/HUMAN], [PLACE])]
               [EVENT cause ([OBJ/EVENT], [EVENT])]
D. [STATE] -> [STATE be ([OBJ/HUMAN], [PLACE])]
               [STATE movingEvent ([OBJ/HUMAN], [PLACE])]
               [STATE orient ([OBJ/HUMAN], [PATH])]
               [STATE extension ([OBJ], [PATH])]

```

Figure 4.2: Predicate-argument forms of some conceptual categories

We analysed 62 common English prepositions and defined 7 PATH predicates and 12 PLACE predicates, as listed in Table 4.1, for interpreting spatial movement events of OBJ/HUMAN. They are in conformity with Jackendoff’s classification (see Chapter 3, section 3.1.3), and can replace the `placePredicate` or `pathPredicate` in Figure 4.2A and B. A typical example of LVSR using EVENT and PATH/PLACE predicates is the following:

Nancy ran across the field.

```
LVSR: [EVENT run [HUMAN nancy] [PATH via [PLACE on [OBJ field]]]]
```

For those verbs with both STATE reading and EVENT reading such as “point”, “face”, “stand”, “sit”, “surround”, “cover”, “hide”, “shelter”, “block”, and “support”, which are known as *inchoative verbs*, the EVENT reading describes a change taking place whose final state is the STATE reading. Figure 4.3 shows examples of three inchoative verbs “point”, “sit”, and “cover”. Jackendoff formalizes the inchoative verbs as `[EVENT inch ([STATE])]`. Hence the two readings of Figure 4.3 (1) can be expressed by Figure 4.3 (1)A and (1)B.

<i>PATH predicates</i>	<i>Direction feature</i>	<i>Termination feature</i>	<i>PLACE predicates</i>	<i>contact/attach feature</i>
to	approaching	+	around	unmarked
from	leaving	+	at	unmarked
toward	approaching	-	behind	unmarked
away_from	leaving	-	end_of	n/a
via	n/a	-	in	unmarked
across	n/a	n/a	in_front_of	unmarked
along	n/a	n/a	near	<-contact>
			on	<+contact>
			out	unmarked
			over	<-contact>
			top_of	n/a
			under	unmarked

Table 4.1: The definition of PATH and PLACE predicates

The primary significance of LVSR is relating arguments in conceptual structure to arguments in syntax. Each lexical item in the sentence specifies how its conceptual arguments are linked to syntactic positions in the phrase it heads. The mapping between syntax analysis and LVSR depends on well-defined lexical entries, in particular, lexical entries for verbs and prepositions. Figure 4.4 illustrates the lexical entries for “into” and “run”. The first line of every entry is the entry word, the second line is its part-of-speech, the third line is its subcategorisation feature, and the fourth line is the lexical conceptual structure. Some words may have a fifth line for modifications. “into” subcategorises for an NP object, which is coindexed with the THING argument *j* in conceptual structure. “run” expects two arguments: the HUMAN in motion and the PATH that specifies the trajectory of motion. The first is indexed *i*, which indicate subject position or *external argument*. The second argument is filled in with a PP, with which it is coindexed in the subcategorisation feature.

1. The weathervane pointed north.

- A. [STATE orient ([OBJ weathervane], [PATH north])]
 B. [EVENT inch ([STATE orient ([OBJ weathervane], [PATH north])])]

2. John sat on the chair.

- A. [STATE be ([HUMAN john], [PLACE on [OBJ chair]])]
 B. [EVENT inch ([STATE be ([HUMAN john], [PLACE on [OBJ chair]])])]

3. Snow covered the mountains.

- A. [STATE be ([OBJ snow], [PLACE on [OBJ mountains]])]
 B. [EVENT inch ([STATE be ([OBJ snow], [PLACE on [OBJ mountains]])])]

Figure 4.3: Alternate readings between STATE and EVENT

Additionally, LVSR can also encode *shadow arguments* (Pustejovsky, 1995), such as the instrument, the goal, the theme and the path, of verbs like “butter”, “pocket”, “cut”, and “hammer”. Figure 4.5 shows that the verb “butter” incorporates information of the theme “butter”, the instrument “a knife” (the default value), and the path “onto the object”, and that “pocket” specifies the path “into a pocket”.

1. $\left[\begin{array}{l} \text{into} \\ P \\ \sim NP_j \\ [\text{PATH to } ([\text{PLACE in } ([\text{THING } j])]]] \end{array} \right]$
2. $\left[\begin{array}{l} \text{run} \\ V \\ \sim \langle PP_j \rangle \\ [\text{EVENT run } ([\text{HUMAN } i], [\text{PATH } j])] \end{array} \right]$

Figure 4.4: Examples of lexical entries

1. **John buttered the bread.**
 $\left[\begin{array}{l} [\text{EVENT put } ([\text{HUMAN john}], [\text{OBJ butter}])] \\ [\text{EVENT go } ([\text{OBJ butter}], [\text{PATH to } ([\text{PLACE on } ([\text{OBJ bread}])])])]) \\ [\text{with } [\text{OBJ knife}]] \end{array} \right]$
2. **Nancy pocketed the money.**
 $\left[\begin{array}{l} [\text{EVENT put } ([\text{HUMAN nancy}], [\text{OBJ money}])] \\ [\text{EVENT go } ([\text{OBJ money}], [\text{PATH to } ([\text{PLACE in } ([\text{OBJ pocket}])])])]) \end{array} \right]$
3. **John hammered a nail into the wall.**
 $\left[\begin{array}{l} [\text{EVENT hit } ([\text{HUMAN john}], [\text{OBJ nail}])] \\ [\text{EVENT go } ([\text{OBJ nail}], [\text{PATH to } ([\text{PLACE in } ([\text{OBJ wall}])])])]) \\ [\text{with } [\text{OBJ hammer}]] \end{array} \right]$
4. $\left[\begin{array}{l} \text{butter} \\ V \\ \sim NP_j \\ [\text{EVENT put } ([\text{HUMAN } i], [\text{OBJ butter}])] \\ [\text{EVENT go } ([\text{OBJ butter}], [\text{PATH to } ([\text{PLACE on } ([\text{OBJ } j])])])]) \\ [\text{with } [\text{OBJ cutlery, eating utensil}]] \end{array} \right]$
 $\left[\begin{array}{l} \text{pocket} \\ V \\ \sim NP_j \\ [\text{EVENT put } ([\text{HUMAN } i], [\text{OBJ } j])] \\ [\text{EVENT go } ([\text{OBJ } j], [\text{PATH to } ([\text{PLACE in } ([\text{OBJ pocket}])])])]) \end{array} \right]$
 $\left[\begin{array}{l} \text{hammer} \\ V \\ \sim NP_j \langle PP_k \rangle \\ [\text{EVENT hit } ([\text{HUMAN } i], [\text{OBJ } j])] \\ [\text{EVENT go } ([\text{OBJ } j], [\text{PATH } k])] \\ [\text{with } [\text{OBJ hammer}]] \end{array} \right]$

Figure 4.5: Incorporated arguments of verb entries

4.3.1 Finer EVENT predicates

Although LCS provides a mapping between syntax and concept representation and is helpful especially in spatial movement of simple objects (atomic entities), it is not possible to represent articulated objects, i.e. human actions and poses. Let's look at the examples in Figure 4.3(2) again. Figure 4.6(1) and (2) shows LCS representation of its STATE reading and EVENT reading which do not specify the pose of the [STATE be] because one may *stand* on the chair instead of *sit* on the chair and both poses are "be on the chair".

Since most animation concerns humanoid characters, Jackendoff’s original LCS is inadequate for the diversity of human actions. For instance, the `EVENT cause` in LCS is overloaded by including both *phrasal causations* (e.g. “cause”, “force”, “prevent”, “impede”) and *lexical causatives* (e.g. “push”, “break” (vt.), “open”, “kill”). Figure 4.7 illustrates the LCS of some phrasal causations and lexical causatives. Figure 4.7(1) is an example of phrasal causation where the effect or potential effect appears as an infinitival or gerundive complement, e.g. cause somebody/something to do something. Figure 4.7(2) shows an example of lexical causation where the cause-effect relation is implicit in the verb. A direct consequence of generalising lexical causatives to `[EVENT cause]` is indistinctness between action verbs. The semantic representation in 4.7(3) indicates that the verb “drink” means a HUMAN causes a liquid OBJ to go into his mouth; and 4.7(2) shows the similar solution to treat the action verb “push” as “cause something away from the agent”.

1. `[STATE be ([HUMAN john], [PLACE on [OBJ chair]])]`
2. `[EVENT inch ([STATE be ([HUMAN john], [PLACE on [OBJ chair]])])]`

Figure 4.6: LCS representation of “John sat on the chair”

1. John forced Bob to go away.

`[EVENT cause ([HUMAN john], [EVENT go ([HUMAN bob], [PATH away]])])]`

2. John pushed the door.

`[EVENT cause ([HUMAN john],
[EVENT go ([OBJ door], [PATH away_from ([PLACE at ([HUMAN
john])])])])])]`

3.

drink
V
~ <NP _j >
[EVENT cause ([HUMAN i],
[EVENT go ([OBJ LIQUID _j],
[PATH to ([PLACE in([THING i.mouth])])])])]

Figure 4.7: LCS examples of phrasal causations and lexical causatives

However, to adapt the semantic representation for humanoid animation generation, we consider that each distinct human action should be regarded as an `EVENT`, and hence has a distinct animation model in the graphic library. The LVS_R representation of two lexical causatives is shown in Figure 4.8. It allows mapping the first `EVENT` to animation models and explicitly codes the `PATH` information of the words in the second `EVENT`.

Now let’s review the inchoative example “John sat on the chair” in Figure 4.9. Figure 4.9 (1) and (2) are LCS representations of two readings, and (3) and (4) are their LVS_R representations. In (4), the `[EVENT sit]` uses the complete animation information (key frames) of “sit”, from a *stand* pose to a *sit* pose, and the second argument specifies its path as `[PATH to [PLACE on [OBJ chair]]]`. In (3), the `[STATE sit]` only uses the final key frame of the *sit* animation, i.e. the *sit* pose solely, and the second argument specifies a `PLACE` rather than a

PATH. Therefore, the [STATE sit] could be visualised as a static scene. Similarly, the resolution of the inchoative verb “point” is presented in Figure 4.10 where the [EVENT orient] indicates “turning to north”.

1. John pushed the door.

```
[EVENT push ([HUMAN john], [OBJ door])]
[EVENT go ([OBJ door], [PATH away_from ([PLACE at ([HUMAN john])])])]
```

2. John lifted his hat.

```
[EVENT lift ([HUMAN john], [OBJ hat])]
[EVENT go ([OBJ hat], [PATH from [PLACE on [OBJ john.head]])]
```

Figure 4.8: LVSR examples of lexical causatives

```
1. [STATE be ([HUMAN john], [PLACE on [OBJ chair]])]
2. [EVENT inchoate ([STATE be ([HUMAN john], [PLACE on [OBJ
chair]])])]
```

LCS representation

```
3. [STATE sit ([HUMAN john], [PLACE on [OBJ chair]])]
4. [EVENT sit ([HUMAN john], [PATH to [PLACE on [OBJ chair]])]
```

LVSR representation

Figure 4.9: LCS and LVSR of “John sat on the chair”

```
1. [STATE orient ([OBJ weathervane], [PATH north])]
2. [EVENT inchoate ([STATE orient ([OBJ weathervane], [PATH north])])]
```

LCS representation

```
3. [STATE orient ([OBJ weathervane], [PATH north])]
4. [EVENT orient ([OBJ weathervane], [PATH north])]
```

LVSR representation

Figure 4.10: LCS and LVSR of “The weathervane pointed north”

Here, we have proposed LVSR as a visual semantic representation between 3D animation and syntactic information. It is based on Jackendoff’s LCS and is adapted to the task of language visualisation. LVSR overcomes the disadvantages of LCS by first introducing finer ontological categories of concepts and adding basic human actions as EVENT predicates since LCS’ event predicates are too coarse for the purpose of language visualisation.

4.3.2 Under-specification and selection restrictions

LVSR helps to resolve the under-specification problem, i.e. under-specified (or unspecified) semantic representations for ambiguous or vague linguistic expressions, and selection restrictions. For example, if no PP is syntactically present in Figure 4.4(2), the PATH is simply unspecified. “Nancy ran” means in part “Nancy traversed some unspecified trajectory”. LVSR requires the PATH argument to be present in conceptual structure even if it is not expressed syntactically, to wit, it is an *implicit argument*. Therefore, one conceptual structure can be expressed in different syntactic forms, for example, both “John entered the house” and “John

entered” can be represented by Figure 4.11(1). “enter” incorporates into its meaning the PATH and PLACE functions expressed separately by the preposition “into” in Figure 4.4(1). Its second argument is a THING rather than a PATH and must be expressed by an NP-complement. The intransitive “enter” in “John entered” means not only “John traversed some path” but also “John went into something”. The sense of “into” appears even when the second argument is unspecified.

Figure 4.11(1) also shows a facility of selection restrictions. The LVSR of “run” indicates that its first argument must be a HUMAN. Figure 4.11(1) reveals the manner of motion—*walk*, when its first argument is a HUMAN, and when the theme is a THING a simple spatial movement *go* is exerted as illustrated in Figure 4.11(2). These facilities of mapping between syntactic and conceptual arguments are significant for language visualisation. The examples in Figure 4.12 show how the selection restrictions provided by ontology categories can also be used to indicate the difference of transitives and intransitives. When the semantic analyser detects that the subject in the input syntax tree is HUMAN, it uses the lexical entry in Figure 4.12(1), otherwise it uses that in 4.12(2).

1.

enter V ~ <NP _j > [EVENT walk ([HUMAN i], [PATH to ([PLACE in ([OBJ j])])])]
--
2.

enter V ~ <NP _j > [EVENT go ([THING i], [PATH to ([PLACE in ([OBJ j])])])]
--

Figure 4.11: Lexical entries for “enter”

1. Nancy broke the bowl into pieces.

[EVENT break ([HUMAN nancy], [OBJ bowl])] [EVENT inchoate [STATE be_comp+ ([OBJ bowl], [PLACE at [OBJ pieces]])]]]

Lexical entry of transitive “break”

break VT ~ NP _j <PP _k > [EVENT break ([HUMAN i], [OBJ j])] [EVENT inchoate [STATE be_comp+ ([OBJ j], [PLACE k])]]]
--

2. The bowl broke into pieces.

[EVENT inchoate [STATE be_comp+ ([OBJ bowl], [PLACE at [OBJ pieces]])]]]
--

Lexical entry of intransitive “break”

break VI ~ <PP _k > [EVENT inchoate [STATE be_comp+ ([OBJ i], [PLACE k])]]]
--

Figure 4.12: Using ontology categories to differentiate transitive and intransitive verbs

4.4 Visual semantics of events

Here we introduce the idea of *visual definition* which defines visual semantics of a word. The methods of visual definition are different according to part of speech. For example, 3D models are used to define nouns, animations based on key-frame are used to define event verbs, property-value pairs are used to define visually presentable adjectives. Visual definition of event verbs can also be a VHML specified human action or a decomposed action specification.

Every visual definition is for one *word sense* rather than one *word*. According to Beckwith's statistics (Fellbaum 1998), one verb has 2.11 senses on average in Collins English dictionary. For instance, "beat" may have two definitions for the sense of "strike, hit" and the sense of "stir, whisk (cooking verb)". Vice versa, synonyms like "shut" and "close" can share one definition. Disambiguation is a task of language parsing, probably solved in language modality by selection restrictions.

However, one word sense of a verb may have more than one visual definition because word sense is a minimal complete unit⁶ of conception in the language modality perspective whilst visual definition is a minimal complete unit of visual representation in the vision modality. Figure 4.13 shows the relationship between visual definition and word sense. Take the word sense "close" as an example again, there could be three visual definitions for a closing of a normal door (rotation on y axis), a closing of a sliding door (moving on x axis), or a closing of a rolling shutter door (a combination of rotation on x axis and moving on y axis).

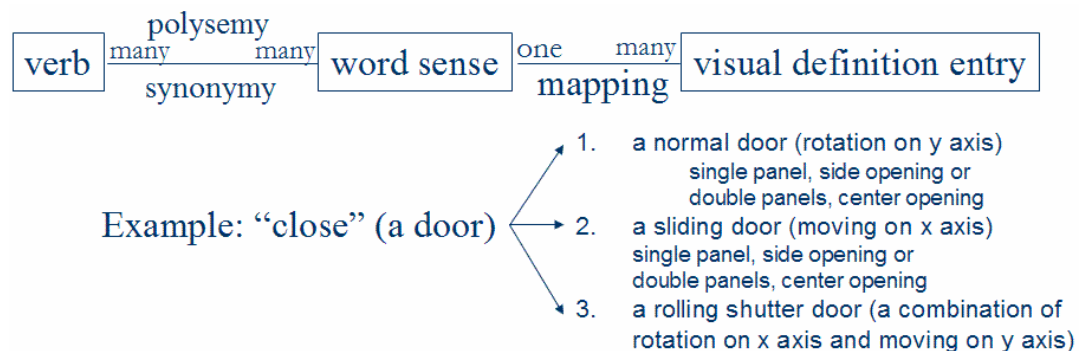


Figure 4.13: Visual definition and word sense

4.4.1 Action decomposition

Although semantic decomposition of lexical meanings is not a new idea in verb semantic analysis (e.g. Schank's Conceptual Dependency analysis), pros and cons are within the language modality only, and little consideration of visual presentation of verb semantics is given. Jackendoff advocates semantic decomposition for the generative facilities it provides. He thinks that the decomposition of word meaning into smaller semantic elements allows specification of

⁶ Strictly speaking, the smallest meaningful unit in the grammar of a language is morpheme. But morpheme is not a *complete* unit since it often concerns prefix and suffix.

a generative, compositional system which constrains the way such elements can be related and thereby constrains the ways in which sentences can be constructed, i.e. to prevent semantically anomalous sentences. Opponents of semantic decomposition argue that it is inadequate because a list of necessary and sufficient conditions of a word meaning does not adequately capture the creative aspect of meaning. Linguists have attempted to set forth the *full and complete* semantic structure of some particular lexical items, and there is always some residue of unexpressed meaning left.

Some verb semantic predicates such as “move”, “go”, or “change” are argued to be the basic components of most verbs from a wide variety of different semantic fields (Jackendoff 1976, Dowty 1991). The decompositional methodology that we use differentiates from the previous semantic decomposition theories on two points: firstly, it aims for the presentation purpose on vision modality rather than the generative or interpretative purposes in language modality; and secondly it does not emphasize *atomic* predicates. It is free to choose predicates in any level to construct a new verb definition.

We propose an action-decomposition structure for presenting visual semantics of action verbs, in which composite actions are defined with a set of more specific, partially-ordered sub-actions as illustrated in Figure 4.14. The practical purpose of introducing this facility is to blend available animations in a graphic library to build new animations. This structure hides geometrical details, so that non-expert users can use it to add new event/action models in a natural way. It can also be used to expand FOPC representations to make them workable on lower levels of linguistic input.

```
[EVENT x]:
  [EVENT x1] {temporal relations}
  [EVENT x2] {temporal relations}
  ...
  [EVENT xi].
```

Figure 4.14: The action decomposition structure

This approach allows high-level representations to control virtual humans’ behaviour. These representations serve as the interface between the animation control mechanism and the language understanding component since translating a complex behaviour into elementary steps needs to be available in the animation library in order to use language understanding to control virtual humans. This approach is useful to extend an animation library by simply giving definitions of complex behaviours which reuses defined elementary actions, rather than loading key frames of a new behaviour.

4.5 Temporal relationships in language

Numerous temporal relations of verbal actions have been analysed in terms of various grammatical means of expressing verbal temporalisation such as tense, aspect, duration and

iteration. In this section, we investigate temporal relationships on the sentence level and lexical level. Specifically, four basic problems in event temporal relationships are addressed:

- A. Ordering events with respect to one another
- B. Anchoring of events (e.g. “John left on Monday.”)
- C. Reasoning about the persistence of events (i.e. how long does an event or the outcome of an event last)
- D. Sense of iteration

There are two main kinds of temporal reasoning formalism in artificial intelligence systems: point-based formalisms to encode relationships between time points (moments), and interval-based temporal calculus to encode qualitative relationships between time intervals (Allen 1983). Point-based linear formalisms are suitable for representing moments, durations, and other quantitative information, whilst interval-based temporal logic is useful for treating actual intervals and expressing qualitative information, i.e. relations between intervals. In the interval temporal logic, temporal intervals can always be subdivided into subintervals, with the exception of moments which are non-zero length intervals without internal structure. Allen argues that “the formal notion of a time point, which would not be decomposable, is not useful” (Allen 1983, p. 834) and the difference between interval-based and point-based temporal structures is motivated by different sources for intuitions: interval logic is meant to model time as used in natural language, whereas point-based formalisms are used in classical physics.

A common problem in the tasks of both visual recognition (image processing and computer vision) and language visualisation (text-to-graphics) is to represent visual semantics of actions and events, which happen in both the space and *time continuum*. We use an interval-based formalism in the compositional predicate-argument representation discussed here to represent temporal relationships in the visual semantics of event verbs. Our choice of temporal structure is motivated by our desire to analyse the composition of actions/events and temporal relationships between ordered pairs of verb entailment based on visual semantics. Since states and events are two general types of verbs and events often occur over some time interval and involve internal causal structure (i.e. change of state), it is convenient to integrate a notion of event with the interval logic’s temporal structure, and event occurrences coinciding, overlapping, or preceding one another, may easily be represented in interval temporal logic.

4.5.1 Sentence level temporal relationship

Representing tense and aspect

The issue of representing temporal information of events is addressed in this subsection. Temporal information that English verb tenses and aspect can convey is listed below.

Tenses: past/present/future

Aspects: progressive/perfective/perfective_progressive

Verb tense and aspect of the language input can be interpreted by a temporal reasoning component during natural language processing. In such a temporal reasoning component, the temporal relationships between events are represented by Allen's interval algebra that was discussed in Chapter 3, section 3.3. For a stand-alone event, the visualisation is identical no matter what tense/aspect the input text is. Consider the following sentences:

John reads the book (every year).

John is reading the book.

John has read the book.

John has been reading the book.

John read the book.

John was reading the book.

John had read the book.

John had been reading the book.

John will read the book.

John will be reading the book.

John will have read the book.

John will have been reading the book.

When appearing alone as single sentences, they would have the same visualisation. Tenses and aspects matter only in successive events (or states) linked by temporal connectives (e.g. "before", "after", "while", "when") or in discourse. For example, in the following sequence:

(1) Mary got up. She brushed her teeth.

The most natural interpretation is one where Mary's teeth brushing follows her getting up. If, however, we replace the second sentence by a state:

(2) Mary got up. She was very hungry.

then a second interpretation is available, where Mary's feeling hungry begins before she gets up and continues afterwards. Another possible reading is that she begins to feel hungry just as, or shortly after, she gets up.

Explicit temporal expressions, such as times, dates, and durations, which are usually expressed by temporal prepositions (e.g. "for", "during", "on", "at"), can be represented by the presentation of a clock or a calendar in storytelling. There are three major types of explicit temporal expression: (a) fully specified temporal expressions (e.g. "1 October 2004"); (b) underspecified temporal expressions (e.g. "Monday", "next month", "last year", "two days ago"); (c) durations (e.g. "nine months", "two days"). Underspecified temporal expressions are solved by the speech time (current time); and durations are presented by animation of clock hands or calendar changes.

Possible aspectual relations expressed by aspectual verbs such as "begin", "start", "finish", "stop", "continue", "keep", "give up" are shown below:

1. Initiation: "John started to read."

2. Culmination: “John finished writing the novel.”
3. Termination: “John stopped talking.”
4. Continuation: “John kept talking.”

Events following aspectual verbs must be protractable processes (either (a) protracted activities or (b) protracted culmination).

- a. protracted activities: run, sleep, read, talk
- b. protracted culmination: build (a house), write (a book), assemble (a table)

We treat initiation and continuation as the progressive aspect, i.e. visualises “John started to read”/“John kept reading” as “John is (was) reading”, and treat culmination and termination as the perfective aspect (i.e. to show the finish or result state of an event).

Sense of iteration

The phenomena discussed so far in this section involve areas where the syntactic category and the semantic category match up such as parts of speech, voice, and tense. However, sense of iteration is not encoded in English syntax through it may be added by some prepositional phrases like “for hours”, “until midnight”, or a temporal quantifier such as “twice”, “three times”, “every”, and so forth. Consider, for example, the difference between the two sentences below:

John taught two hours every Monday. (iteration)

John taught two hours on Monday.

Sense of iteration is closely related to the notation of *temporal boundedness*. Jackendoff (1991) introduced the bounded/unbounded distinction in the temporal dimension. A bounded predicate reaches an actual temporal boundary. Table 4.2 shows some examples of temporal boundedness of events. Temporal bounded events (e.g. Table 4.2: 1, 3) are also called *punctual events* or *achievement events* (distinct from *accomplishment events*, Vendler 1967, Smith 1991). The prepositional phrases “for hours” and “until midnight” can follow temporally unbounded processes, and place either a measure or a boundary on them. “John slept”, for instance, expresses an unbounded event, so it can be felicitously prefixed with these prepositional phrases. But “John waked” expresses a temporally bounded event, so it cannot be further measured or bounded by these prepositional phrases.

Some verbs have the sense of repetition included/hinted in their lexical semantics, e.g. Table 4.2: 5 and 6. Prefixing “for hours” or “until midnight” adds or enhances the sense of repetition to them. However, there is a nuance between 5 and 6. Without those prepositional phrases, “the light flashed” means it flashed once, whereas “Jane hammered the door” suggests she hammered the door repeatedly. Therefore, “for hours” *adds* the sense of repetition in 5, and *enhances* it in 6. Example 1 and 3 are bounded but unrepeatable, so they cannot give grammatical productions when prefixing “for hours” or “until midnight”.

<i>Examples</i>	<i>Temporal boundedness</i>	<i>Prefixing “for hours” or “until midnight”</i>
1) John woke.	bounded	not acceptable
2) John slept.	unbounded	acceptable
3) John entered the house.	bounded	not acceptable
4) John walked toward the house.	unbounded	acceptable
5) The light flashed.	bounded (repeatable)	acceptable, add the sense of repetition
6) Jane hammered the door.	bounded (repeatable)	acceptable, enhance the sense of repetition

Table 4.2: Temporal boundedness of events

The bounded/unbounded distinction in events is strongly parallel to the count/mass distinction in NPs (Jackendoff 1990). The criterion for the boundedness and countableness distinction has to do with the description of parts of an entity. For instance, a part of “an apple” (countable) cannot itself be described as “an apple”, but any part of a body of “water” (mass) can itself be described as “water”; a part of the event “John entered the house” (bounded) cannot itself be described as “John entered the house”, but any part of “John walked toward the house” (unbounded) can be described as “John walked toward the house”. Therefore, a static graphic scene can only represent unbounded events such as “John walked toward the house” properly, by selecting a representative part of the event; while bounded events are better presented by animation.

Distinction for sense of iteration is very important for visualising events since the animation generator needs to know whether it’s necessary to repeat an action loop, and whether it’s necessary to animate the complete process of an event (a bounded event) or just a part of it (an unbounded event).

Modal verbs and subjunctive mood

Modal verbs (e.g. “should”, “could”, “will/would”, “able to”, “may/might”), the subjunctive mood and some cognition verbs introduce events in a possible/mental world (see the examples below).

John should have bought some wine.

Mary wanted John to buy some wine.

If she was/were here now, . . . (present)

If she had been here then, . . . (past)

Montage techniques are applied to present this world in language visualisation. Subjective flashback and subliminal flash shots are adopted to show the thoughts, imagery and memory of a character. We open a second window and present the allusion in it. This new channel which is impossible in conventional film-editing allows direct communication of characters’ thoughts and mental-related activities. *Voiceovers* (narrator’s or characters’ speech) are also a useful modality to express subjunctive mood and cognition.

Factive/counterfactive verbs

There are several verbs which can be followed by either a gerund or infinitive with different temporal meanings (e.g. remember, forget, regret, try). When these verbs are used with a gerund or a *that* clause they refer to something that happened before a certain time. When they are used with an infinitive they refer to something that happens after the speech time. Consider the examples of “forget” below:

1. I'll never forget going to Japan. (refer to a previous action)
2. John forgot that he was in London three years ago. (refer to a previous action)
3. Don't forget to meet me at 5.00. (refer to something happening after the speech time)

We use speech modality to present the negative meaning of “forget”, and for other verbs in this small group both visual, such as the montage techniques mentioned above, and speech modalities are considered for presentation.

Events and causation

Event causation involves more than temporal precedence of events. However, we only discuss temporal ordering of event causation here. Four distinct cases of event causal relations are identified here:

1. Event cause event: The rains caused the flooding.
2. Entity cause event: The technician's inadvertence caused the accident.
3. Event discourse marker event: He kicked the door, and it opened.
4. Implicit lexical causation: John killed the young man. (The young man died.)

We establish the precedence relation between result events and caused events for explicit causation (case 1 and 2) which are expressed by verbs such as the following, in their causative senses: “cause”, “stem from”, “lead to”, “breed”, “engender”, “hatch”, “induce”, “occasion”, “produce”, “bring about”, and “produce”. It is the same for causation expressed by the discourse marker “and” in case 3. We will now discuss case (4) in section 4.5.2 in detail since it belongs to the lexical level.

4.5.2 Temporal relations in lexical semantics of verbs

In this section the temporal relations within verb semantics, particularly ordered pairs of verb entailment, are studied using Allen's interval-based temporal formalism. Their application to the action decomposition of visual definitions is presented, including the representation of procedural events, achievement events and lexical causatives. In applying these methods we consider both language modalities and visual modalities.

Temporal Relations in Verb Entailments

Various temporal relations between ordered pairs of verbs in which one entails the other are studied and their usage in visualisation is discussed here. Verb entailment and troponymy, a special semantic relation in verb entailment, were discussed in Chapter 2, section 2.8.2. Figure 4.15 shows a tree of troponyms, where children nodes are troponyms of their parent node (e.g. the bolded route *limp/stride/trot-walk-go*). We use the method of base verb + adverb to present manner elaboration verbs, that is, to present the base verb first and then, to modify the manner (speed, the agent’s state, duration, and iteration) of the activity. To visually present “trot”, we create a loop of walking movement, and then modify a cycle interval to a smaller value to present fast walking.

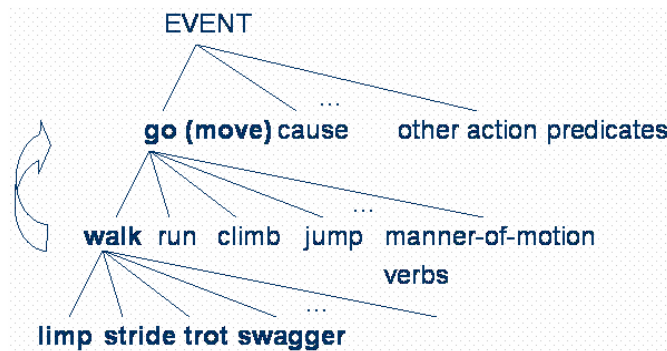


Figure 4.15: A troponymy tree

In Table 4.3 we analyze the possible temporal relations between these verb entailments using Allen’s interval algebra (see Table 2.4) and give some examples. We note that the interval relation between a troponym pair of verbs is $\{\equiv\}$, e.g. *limp* \equiv *walk*. The relation set of $\{p,m,o,s,f^{-1},\equiv\}$ may hold in any pair with causal structure (e.g. lexical causatives), between the eventive verb and its result state (either stative verb or adjective), such as *give-have*, *eat-full*, *work-getPaid*, *heat-hot*. Thanks to the productive morphological rules in English deriving verbs from adjectives via affixes such as *-en* and *-ify*, these deadjectival verbs, e.g. “whiten”, “shorten”, “strengthen”, “soften”, often refer to a change of state or property and have the meaning (*make/become/cause* + corresponding adjective or its comparative form). The temporal relation between the pair of deadjectival verbs and the state of their corresponding adjectives is also $\{p,m,o,s,f^{-1},\equiv\}$. For instance, the possible interval relations set between *shorten-short/shorter* could be *shorten* $\{p,m,o,s,f^{-1},\equiv\}$ *short/shorter*. Similarly, the relation set $\{p,m,o,s,f^{-1},\equiv\}$ is also applicable to cognate verbs and adjectives (or their comparative forms) such as *beautify-beautiful* and *clarify-clear/clearer*.

Propagation within Interval Logic

Allen’s interval algebra is convenient for describing the propagation algorithm used for logical inference through a collection of intervals, determining the most constrained disjunction of relations for each pair of intervals which satisfies the given relations. Reversibility and

transitivity are important elements of temporal reasoning, and they provide a facility for propagating temporal relationships through a collection of intervals, determining the most constrained disjunction of relations for each pair of intervals which satisfies the given relations and is consistent in time. By adding directions to interval relations we may denote the implication logic relationship between two events.

Verb entailment relations	Temporal relations	Examples
Troponym	$\{\equiv\}$	limp \equiv walk
non-troponym (proper temporal inclusion)	$\{d, d^{-1}\}$	snore d sleep, buy d^{-1} pay
backward presupposition	$\{p^{-1}, m^{-1}\}$	untie p^{-1} tie \cup untie m^{-1} tie
Cause	$\{p, m, o, s, f^{-1}, \equiv\}$	eat p fullUp \cup eat o fullUp, give m have, build o exist

Table 4.3: Temporal relations in verb entailments

The basic algebra of temporal relations includes reversibility and transitivity. The reversibility of an interval relation is:

$$\forall R: \text{act1 } R \text{ act2}, R \in \{p, p^{-1}, m, m^{-1}, o, o^{-1}, d, d^{-1}, s, s^{-1}, f, f^{-1}\} \Leftrightarrow \text{act2 } R^{-1} \text{ act1}$$

For instance, untie p^{-1} tie \Leftrightarrow tie p untie. All interval relations are reversible, and the relation \equiv is reflexive, symmetric, and transitive.

Transitivity of one interval relation is defined as:

$$\text{if } \exists R: (\text{act1 } R \text{ act2}) \cap (\text{act2 } R \text{ act3}), R \in \{p, p^{-1}, d, d^{-1}, s, s^{-1}, f, f^{-1}, \equiv\} \Rightarrow \text{act1 } R \text{ act3}$$

then this temporal relation R is transitive, to wit, the temporal relations between the pairs of intervals can be propagated through the collection of all intervals. For instance, born p age, age p die \Rightarrow born p die. Notice that m and m^{-1} are not in the set of possible transitive relations, because the nature of these two relations is not transitive, i.e. $(\text{act1 } m \text{ act2}) \cap (\text{act2 } m \text{ act3}) \Rightarrow \sim(\text{act1 } m \text{ act3})$. All the other temporal relations $\{p, p^{-1}, d, d^{-1}, s, s^{-1}, f, f^{-1}, \equiv\}$ must be transitive except o and o^{-1} since $(\text{act1 } o \text{ act3})$ cannot be inferred from $(\text{act1 } o \text{ act2}) \cap (\text{act2 } o \text{ act3})$, though it might be true.

The temporal reasoning of the interval relations can be obtained by computing the possible relations between any two time intervals. For instance, $(x \ d^{-1} \ y) \cap (y \ p \ z) \Rightarrow x \ R \ z$, $R \in \{p, o, d^{-1}, f^{-1}, m\}$. In this case, x could be the activity ‘‘buy’’, y could be the activity ‘‘pay’’ and z could be ‘‘consume’’.

We revise Allen’s interval logic by adding directions of *implication* logic relationships to it, using $R>$, $<R$, or $<R>$, $R \in \{p, p^{-1}, m, m^{-1}, o, o^{-1}, d, d^{-1}, s, s^{-1}, f, f^{-1}, \equiv\}$. Hence, limp \Rightarrow walk indicates their troponymy relation, and \Leftrightarrow indicates synonym relations like speak \Leftrightarrow say, or same activity from different perspectives such as teach \Leftrightarrow learn, buy \Leftrightarrow sell. By this facility we may also use build $o>$ exist to indicate causal relationship (in prediction), and use tie $<p$ untie to indicate backward presupposition (in planning).

Application of Interval representation

The interval temporal logic discussed above can be combined with truth conditions of perceptual primitives such as support, contact, and attachment to represent simple spatial motion events for recognizing motion verbs in animation or vision input, such as Siskind's (1995) event logic. The drawback of Siskind's event logic is that it is limited to a reduced set of actions, such as *drop*, *place*, and *pick up*. Here we apply the interval logic to the compositional model of visual definition, to represent the temporal relationship between subactivities.

The relationship between the definiendum verb and the defining subactivities is temporal inclusion (whether proper inclusion or not), i.e. $act1 R act2$, $R \in \{d, s, f, \equiv\}$, $act1$ is part of, or a stage in, temporal realisation of $act2$, and hence it could be one sub-activity in $act2$'s visual definition⁷. \equiv is a special case. If there is only one subactivity in a definition and the relation of this subactivity and its defined verb is \equiv or $\equiv>$, the definition is rather an interpretation than a semantic decomposition, e.g. in the definition [EVENT slide]: [EVENT move], the temporal relation between the subactivity and definiendum is $slide \equiv move$. Because "slide" is a troponym of "move", i.e. $slide \equiv> move$, we use "move" to define "slide" but not "slide" to define "move". The relationship between any subactivity and the verb sense it defines are $\{d, s, f, \equiv\}$, i.e.

```
[EVENT x]8:
    [EVENT x1] {temporal relations}
    [EVENT x2] {temporal relations}
    ...
    [EVENT xi].
xi R x, i ∈ N, R ∈ {d, s, f, ≡}
```

The temporal relations between two neighbouring subactivities are $\{p, m, o, f^1, d^1, \equiv\}$, i.e. $x_i R x_{i+1}$, $R \in \{p, m, o, f^1, d^1, \equiv\}$. \equiv is used to indicate the temporal relation between two simultaneous activities. [EVENT x] \equiv [EVENT y] means that x and y start and finish at the same time. This temporal relation is usable for defining verbs such as rolling of a wheel:

```
[EVENT roll]:
    [EVENT move] ≡
    [EVENT rotate].
```

Figure 4.16 shows the decomposition of "turn" in "turn a vehicle". The activity of *slowDown* can *overlap/include/be finished by* *changeGear* besides *preceding* or *meeting* *changeGear*, i.e. $slowDown \{p, m, o, f^{-1}, d^{-1}\} changeGear$. It is necessary to distinguish the relation between *slowDown* and *changeGear* with the relation between *steer* and

⁷ $act2$ is the definiendum verb, and $act1$ is one of its defining subactivities.

⁸ Parameters of these EVENTS are omitted because the number and type of parameters depend on the EVENT.

straight, because the latter relation is just a simple *precede* or *meet* relation⁹ $\{p, m\}$ whilst the former relation could be any of $\{p, m, o, f^1, d^{-1}\}$.

```
[EVENT turn] :
...
[EVENT slowDown] {p, m, o, f-1, d-1}
[EVENT changeGear] {p, m}
...
[EVENT steer] {p, m}
[EVENT straight].
```

Figure 4.16: Decomposition of “turn” using interval algebra

Interval algebra can define multiple temporal relationships (even in reverse order) in one definition. For instance, one may argue the *eatOut* definition in Figure 4.17A that in fast food shops people pay first and then get the food they order. Figure 4.17B includes this circumstance by adding p^{-1} in the relation set between “eat” and “pay”, as opposed to defining another event describing *eatOut* in fast food shops. The distinction between A and B shows the nonlinear advantage on efficiency and flexibility, which is similar to partial-order planning¹⁰ vs. total-order planning. The decomposition structure using interval algebra can also represent optional subactivities. In the *eatOut* example, *bookSeat()* is optional. We use $\langle \rangle$ to indicate optional subactivities (Figure 4.17C).

In terms of punctual events discussed in Chapter 3, section 3.3.1, interval algebra can also represent punctual events. Previous considerations of punctual events are in respect of language modalities. When multimodal representation is concerned, we take visual representation into account, punctual events could also be represented using interval-based relations. As stated in Pinon’s boundaries analogy the existence of achievement events depends on the existence of their corresponding accomplishments, and in visual representation we cannot separate these events from their context, e.g. to separate “find” from “search”, and “arrive” from “go”. In computer games and dynamic visual arts like movies, for example, the event “die” is usually associated with a “falling” movement. When we include context in their visual definitions (Figure 4.18), these events become intervals rather than moments. Therefore we can declare that all verbs are in time intervals, whether they indicate states, processes, or punctual events. Strictly speaking, the relationships between these punctual events and the subactivities in their visual definitions cover all five possible relations between a point and an interval: *starts*, *before*, *during*, *finishes*, and *after* since these are also the relations between punctual events and their contexts.

⁹ Because the activity “straight” must happen after “steering” finishes.

¹⁰ Partial-order planning focuses on relaxing the temporal order of actions. Plans can be *totally ordered* if every action is ordered with respect to every other action, or *partially ordered* if actions can be unordered with respect to each other.

```
[EVENT eatOut]:
  [EVENT bookSeat] {p}
  [EVENT go[HUMAN i], [PATH to [PLACE in [OBJ restaurant]]]] {p,m}
  [EVENT orderDishes] {p}
  [EVENT eat] {p,m}
  [EVENT pay] {p,m}
  [EVENT go[HUMAN i], [PATH from [OBJ restaurant]]].
```

A. “eatOut” in a restaurant

```
[EVENT eatOut]:
  [EVENT bookSeat] {p}
  [EVENT go[HUMAN i], [PATH to [PLACE in [OBJ restaurant]]]] {p,m}
  [EVENT orderDishes] {p}
  [EVENT eat] {p,p-1,m}
  [EVENT pay] {p,m}
  [EVENT go[HUMAN i], [PATH from [OBJ restaurant]]].
```

B. “eatOut” in a restaurant/fast food shop

```
[EVENT eatOut]:
  <[EVENT bookSeat] {p}>
  [EVENT go[HUMAN i], [PATH to [PLACE in [OBJ restaurant]]]] {p,m}
  [EVENT orderDishes] {p}
  [EVENT eat] {p,p-1,m}
  [EVENT pay] {p,m}
  [EVENT go[HUMAN i], [PATH from [OBJ restaurant]]].
```

C. Optional subactivities

Figure 4.17: Visual definitions of “eatOut”

<pre>[EVENT die]: [EVENT fall].</pre>	<pre>[EVENT find]: [EVENT search]{m} [EVENT eyesFixedOn].</pre>
<pre>[EVENT arrive]: [EVENT go [HUMAN] [PATH to [PLACE]]].</pre>	

Figure 4.18: Examples of punctual events’ visual definitions

Temporal relationships in lexical causatives

Visual definition should also include causative information to determine the result state following a particular action, i.e. the effects of actions. Hence the visual definitions of causative verbs like “kill” must subsume their result states (stative verbs) like “die” as the following:

```
[EVENT kill]:
  [EVENT hit] {p,m,o,f-1}
  [STATE die [HUMAN victim]].
```

Moreover, interval relations can represent the distinction between *launching* and *entraining* causation. In Table 4.4, the sentences (1-4) describe causation of the *inception* of motion (launching causative), whereas (5) describes *continuous* causation of motion (entraining causative). A disjunction set of interval relations between the cause and the effect is adequate to define the difference: {p,m,o,s} for launching causative verbs (1-4), and {≡,f⁻¹} for entraining causatives (5).

Examples	Temporal relation between cause-effect
1. John threw the ball into the field.	{s}
2. John released the bird from the cage.	{p}
3. John gave the book to me.	{m}
4. John opened the door.	{o}
5. John pushed the car down the road.	{ \equiv, f^{-1} }

Table 4.4: Launching and entraining causation

Representing actions consisting of repeatable periods

We introduce a facility to represent repeatable periods of subactivities since many actions may be sustained for a while and consist of a group of repeatable subactivities. We use $\langle \rangle$ and a subscript R to indicate the repetition constructs in examples of Figure 4.19, which can also be captured by Kleene iteration in finite state descriptions for temporal semantics (Fernando 2003). The activities bracketed by $\langle \rangle_R$ are repeatable. Besides periodical repetition of subactivities, it can represent morphological prefix "re-" as well, as the "recalculate" example in Figure 4.19, substituting the number of iterations (which is 2 in this case) for R . This facility of representing iteration may be used for post-lexical level repetition as well, e.g. events marked by "again", "continues to", or "a second time".

```
[EVENT hammer [HUMAN] [OBJ]] :
  <[EVENT hit [HUMAN] [OBJ] [INSTRUMENT hammer]>_R.
[EVENT recalculate] :
  <[EVENT calculate]>_2.
```

Figure 4.19: Verbs defined by repeatable subactivities

Due to the advantages in representing temporal relations between entailed verb pairs, punctual events and their contexts, lexical causatives, and iteration of sub-activities, we adopt interval logic rather than point-based logic. Though some may argue that in some cases like $\text{kill } f^{-1} \text{ die}$, the quantitative factor is critical. Fernando (2003) introduces a temporal granularity δ which is a non-empty observation interval $\delta \in \mathbb{R}$ greater than 0. An event with end-points p, q :

$$(p,q) = \{r \in \mathbb{R} \mid p < r < q\}$$

p, q are real numbers indicating temporal instants of the event's end-points. \mathbb{R} denotes the set of real numbers. $p, q \in \mathbb{R}$ and $q - p > \delta$. The requirement $q - p > \delta$ bounds the precision of a δ -observation and ensures that δ covers part of the event. Next, given O for a set of events, let Succ be the binary relation on O defined by

$$(p,q) \text{ Succ}(p',q') \text{ iff } 0 \leq p' - q \leq \delta$$

for all $(p,q), (p',q') \in O$. The requirement $p' - q \geq 0$ ensures that each point in (p,q) is less than each point in (p',q') ; while $p' - q \leq \delta$ precludes a gap between (p,q) and (p',q') large enough to squeeze in an intervening δ -observation, and hence ensures δ covering both events

(at least part of each). Therefore, succ is exactly Allen's relation $\{p\}$ with an observation seeing the end point of the former event and the start point of the latter event.

Let's analyse the intrinsic causal structure of the event *kill*. There is a nuance in meaning between *kill* and *cause to become not alive* (Arnold et al. 1994) in virtue of the quantitative factor of temporal relation, in particular, where a *killing* is a single event, a *causing to become not alive* involves two events: a *causing* and a *dying*. If the causal chain that links a particular event to dying is long enough, one may admit that the event caused the dying, but not want to say there has been a 'killing'. For the situation where the causing event is a *shooting*, i.e. shoot succ die constructs the event *kill*. $p' - q \leq \delta$ makes sure that the time difference between p' (the start point of *die*) and q (the end point of *shoot*) is less than or equal to the observation interval δ , i.e. only when the both events are observed the event *kill* is fulfilled, which requires the visualisation of *killing* to subsume both the cause and the *dying* events (or part of each, at least). How long should the state of *wounded but not dead* ($p' - q$) last that we can say that it is one event *kill* rather than two events *cause* and *die*? We can see the importance of the quantitative factor for language generation, especially in the stage of sentence planning and surface realisation. However, it might not be critical for language understanding (or visualisation), e.g. the interpretation of *kill* includes shoot $\{p, m, o, f^1\}$ die, and if the relation is shoot p die (or shoot succ die), we can give a reasonable value to the $p' - q$, say, several minutes, before the victim finally dies. The value could even be 0 or negative in the case of shoot $\{m, o, f^1\}$ die, i.e. the killer kept shooting until (or after) the victim died.

Temporal relation is a crucial issue in modelling action verbs, their procedures, contexts, presupposed and result states. We have discussed temporal relations between multiple events on sentence level and within verb semantics such as various temporal interrelations between ordered pairs of verb entailment, accomplishment and achievement events, and lexical causatives. We propose an enhanced compositional visual definition of verbs based on Allen's interval logic and apply it to animation generation.

4.6 Categories of nouns for visualisation

In this section we classify nouns from the visual semantic perspective. We have investigated sub-categories of nouns in other semantic ontology in Chapter 2, section 2.10.2. Here we give our visual semantic based noun categorisation. Having analysed the eleven noun tops of WordNet (see section 2.9.1), we classify concrete nouns into four top categories (see Figure 4.20) to suit visualisation purposes. The figure shows that visual semantic representation of concrete nouns concerns four issues: (1) the existence of the entity (including OBJ and HUMAN), i.e. its physical features like 3D size and color, (2) its position in a three dimensional space (i.e. PLACE and PATH), (3) mass nouns and grouping (AMOUNT), and (4) TIME.

There are two other groups of semantic components with the form of noun while expressing concepts of verb or adjective, e.g. *jumping*, *falling*, *happiness*. One concept group is

event, i.e. the category 7 (event) and 8 (act, humanaction, humanactivity) in Figure 2.29. The other group concerns *properties, features, or states* of entities, which cannot be seen in the same way as concrete nouns, i.e. category 2 (psychological feature), 3 (abstraction), 5 (shape, form), 6 (state), and 11 (phenomenon) in Figure 2.29.

1. entity (OBJ, HUMAN)
2. location, space (PLACE, PATH)
3. group, grouping (AMOUNT)
4. TIME

Figure 4.20: Concrete noun categories

For the part-whole semantic relation, also called meronymy in WordNet (Miller 1994), in nouns, LVSR introduces a dot operator to denote a part of an OBJ or HUMAN, simulating the convention in Java and C/C++. Figure 4.8(2) shows this usage. One main task of semantic analysis is to segregate OBJs and HUMANS since they are two ontological categories of LVSR. We use WordNet's hypernym tree to assist this task.

For objects, we add three types of information besides those required in VRML representation (e.g. position, orientation, size, color). They are *grasp sites*, *spatial tags*, and *the longest axis*. Grasp sites specify the particular place when the object is grasped, for instance, the grasp site of *a cup* is its handle which is different from grasp sites of *a bottle* (cylinder) or *a piece of paper* (sheet). Spatial tags were introduced by Badler et al. (1997) as *directions*. We define the spatial tag set as

```
spatial tag set of OBJs = {front, back, in, on, under, left, right}
```

to depict spatial relations (cf. PLACE predicates in Table 4.1). We add these spatial tags with all objects in their geometry files in VRML format.

4.7 Visual semantic representation of adjectives

All languages provide some means of modifying and elaborating the qualification of nouns. Noun modification is primarily associated with the syntactic category *adjective* whose function is modifying nouns, though modifiers could also be prepositional phrases, noun phrases, or even entire clauses. In this section, the subcategories of adjectives from the visualisation perspective and their visual semantic representation are discussed.

4.7.1 Categories of adjectives for visualisation

Conventional classification of adjectives (Gross and Miller 1990) divides them into two major classes: descriptive adjectives and relational adjectives. Descriptive adjectives (such as “large”/“small”, “interesting”/“boring”) ascribe to their head nouns values of bipolar attributes and consequently are organised in terms of binary oppositions (antonymy) and similarity of meaning (synonymy). Relational adjectives, such as “nuclear” and “royal”, are assumed to be stylistic variants of modifying nouns and can be cross-referenced to the nouns.

In Figure 4.21 we classify the category of adjectives according to the perceiving senses they require. The first level is distinguished by the standard as to whether they can be perceived through visual sense as vision is a main input modality of human perception. Visually observable adjectives are adjectives whose meaning can be directly observed by human eyes. They consist of adjectives describing objects' attributes or states, e.g. dark/light, large/small, white/black (and other color adjectives), long/short, new/old, high/low, full/empty, open/closed, observable human attributes, and relational adjectives. Observable human attributes include human emotions, such as happy/sad, angry, excited, surprised, terrified, and other non-emotional features such as old/young, beautiful/ugly, strong/weak, poor/rich, fat/thin. Human feelings are usually expressed by facial expression and body posture, while non-emotional features are represented by some body features or costumes. This convention is also used in performance art.

The third kind of *visually observed adjectives* is a large and open class — *relational adjectives*. They usually mean something like “of, relating/pertaining to, or associated with” some noun instead of relating to some attribute, and play a role similar to that of a modifying noun. For example, “nasal”, as in “a nasal voice” relates to “nose”, “mural”, as in “mural painting”, relates to “wall”, and “royal” relates to “king” or “queen”. Some head nouns can be modified by both the relational adjective and the noun from which it is derived: both “atomic bomb” and “atom bomb” are acceptable. So the relational adjective and its related noun refer to the same concept, but they differ morphologically. Moreover, relational adjectives have features like nouns and unlike descriptive adjectives: they do not refer to a property of their head nouns; they are not gradable; they do not have antonyms; and the most important, their visual semantics are the same as their corresponding nouns. Therefore we treat this subcategory of adjectives as nouns, and represents the appropriate nouns that they point to in WordNet.



Figure 4.21: Categories of adjectives

There are four types in the unobservable class. The first type is adjectives that can be perceived by haptic sensors, e.g. wet/dry, warm/cold, coarse/smooth, hard/soft, heavy/light. The

second type of adjectives is perceivable by the auditory modality, e.g. quiet/noisy, loud/soft, cacophonous/euphonious. The third type of visually unobservable adjectives is abstract attributes, either unobservable human attributes concerning virtue (e.g. good/evil, kind, mean, ambitious) or non-human attributes (e.g. easy/difficult, real, important, particular, right/wrong, early/late). The last type is the closed class of *reference-modifying adjectives*. They are a relatively small number of adjectives including “former”, “last”, and “present”. Many refer to the temporal status of the noun (e.g. “former”, “present”, “last”, “past”, “late”, “recent”, “occasional”); some have an epistemological flavour (“potential”, “reputed”, “alleged”); others are intensifying (“mere”, “sheer”, “virtual”, “actual”) or *degree of certainty* (“likely”, “possible/impossible”). The reference-modifying adjectives often function like adverbs: “the former Prime Minister” means “he was formerly Prime Minister” and “the alleged killer” states that “she allegedly killed”.

We represent unobservable adjectives in language and audio modalities. Here we shall distinguish narrator’s language with character’s language. If the adjective appears in a character’s dialogue it should be synthesized and presented in speech modality; if it appears in the narration part, a natural language processing component can identify which adjective category it belongs to and which modality should be used to present the concept. Visually unobservable adjectives may be presented by a narrator’s voiceover (speech) or nonspeech sounds such as auditory icons, nonverbal expressions, or music.

4.7.2 Semantic features of adjectives relating to visualisation

Most of the observable adjectives (except relational adjectives) are descriptive. There are some semantic features of descriptive adjectives that relate to visualisation. One basic semantic relation among these adjectives is antonymy. The importance of antonymy first became obvious from results obtained with word association tests (Fellbaum 1998): when the probe is a familiar adjective, the response commonly given by native speakers is its antonym. For example, for the word “good”, the common response is “bad”; for “bad”, the response is “good”. This mutuality of association is a salient feature of the data for adjectives. The importance of antonymy in the organisation of descriptive adjectives is understandable when it is recognized that the function of these adjectives is to express values of attributes, and that nearly all attributes are bipolar. Antonymous adjectives express opposing values of an attribute. For example, the antonym of “large” is “small”, which expresses a value at the opposite pole of the SIZE attribute.

However, besides a handful of frequently used adjectives which have indisputable antonyms, and a number of antonyms that are formed by morphological negative prefix (e.g. un-, in-, il-, im-, ir-), there are numerous adjectives which seem to have no appropriate antonyms or whose antonyms are disputable, e.g. “soggy”, “ponderous”. A simple solution is to introduce a similarity relation and use it to indicate that the adjectives lacking antonyms are similar in meaning to adjectives that have antonyms. This strategy in lexical semantics greatly reduces the

graphic representation of synonyms. For the example in Figure 3.9, only two graphic representations are enough to express the meaning group “wet”-“dry”, no matter if the actual word from input is “soggy” or “sere”.

Gradation is another feature of some descriptive adjectives. For some attributes gradation can be expressed by ordered strings of adjectives, all of which pertain to the same attribute noun. Table 4.5 illustrates lexicalized gradations for SIZE, WHITENESS, AGE, VIRTUE, VALUE, WARMTH and ANGER. In Table 4.5 adjectives with bipolar values are shown in italics which are antonyms in the given column.

<i>SIZE</i>	<i>WHITENESS</i>	<i>AGE</i>	<i>VIRTUE</i>	<i>VALUE</i>	<i>WARMTH</i>	<i>ANGER</i>
astronomical	snowy	Ancient	saintly	superb	torrid	furious
huge	<i>white</i>	<i>Old</i>	<i>good</i>	great	<i>hot</i>	wrathful
<i>large</i>	ash-gray	middle-aged	worthy	<i>good</i>	warm	enraged
standard	gray	Mature	ordinary	mediocre	tepid	angry
<i>small</i>	charcoal	adolescent	unworthy	<i>bad</i>	cool	Irate
tiny	<i>black</i>	<i>Young</i>	<i>evil</i>	awful	<i>cold</i>	Incensed
infinitesimal	pitch-black	Infantile	fiendish	atrocious	frigid	Annoyed

Table 4.5: Examples of graded adjectives

Generally, graded adjectives represent a range of points along a linear state machine. Figure 4.22 shows some examples of LVSR adjective entries. The meaning of all of the adjectives of size, for instance, is described as a value on the size-attribute scale, and many of them differ from each other only in the numerical value, while all of the rest of the constraints in the semantic part of their lexical entries remain the same as those in a sample entry, say, that for “big”. Thus, the entries for “enormous” and “tiny” differ from that for “big” in the way shown in Figure 4.22. One sense of “fat” (given in Figure 4.22), as in “fat man”, has a similar entry with a different scale, MASS, substituted for SIZE, and modifies the LVSR category HUMAN instead of physical objects (i.e. LVSR category OBJ). The values in the entries are relative to 1, the normal (original) value of the modified object or human model.

enormous	big	Tiny	fat
ADJ	ADJ	ADJ	ADJ
attribute: SIZE	attribute: SIZE	attribute: SIZE	attribute: MASS
category: OBJ	category: OBJ,HUMAN	category: OBJ	category: HUMAN
value: 2	value: 1.3	value: 0.2	value: 1.5

Figure 4.22: Example LVSR adjective entries

Other types of adjectival senses based on numerical scales: quantity-related (e.g. “abundant”, “scarce”, “plentiful”), price-related (e.g. “affordable”, “cheap”, “expensive”), human-height-related (e.g. “tall”, “short”), human mass-related (e.g. “fat”, “thin”, “emaciated”, “chubby”), container volume-related (e.g. “capacious”, “tight”, “spacious”) can be treated this way as well. It gives an automatic visualisation system more control over representing these attributes. Attributes like SIZE, LENGTH and WHITENESS are easy to present compared with other attributes such as *observable human attributes*.

Just like the fact that a single meaning can have many words (synonymy), a single word can have many meanings (*polysemy*). Polysemy and selectional preferences is an important issue of disambiguation of adjectives which have specific meanings when occurring with specific nouns, e.g. “old” in “my old friend” means long friendship and the friend is possibly young, while “old” in “an old man” means the man is old in terms of age. Justeson and Katz (1993) note that the noun context therefore often serves to disambiguate polysemous adjectives.

Even in a same word sense, the values of an adjective could be different, depending on the head nouns that they modify. For instance, “tall” denotes one range of heights for a person, another for a building, and still another for a tree. It appears that part of the meaning of each of the nouns “person”, “building”, and “tree” is a range of expected values for the attribute HEIGHT. “Tall” is interpreted relative to the expected height of objects of the kind denoted by the head noun. Therefore, in addition to containing a mere list of its attributes, a nominal concept is usually assumed to contain information about the default values of those attributes: for example, although both buildings and persons have the attribute of HEIGHT, the default height of a building is much greater than the default height of a person. The adjective modifies the attribute by multiplying the object’s default value by the value specified in its lexical entry.

Graded adjectives can also provide subjective descriptions of an object depending on the speaker’s individual background and experience. So a person from a remote town may describe a three-storey building as “tall”, whereas another person from a large city would describe it as “short”.

4.7.3 Entity properties for visual and audio display

WordNet, as discussed in Chapter 3, section 3.4.1, explicitly encodes ascriptive adjectives by describing the attribute to which the adjective ascribes a value. For example, the attribute for “loud” is “volume”. The complete list of attributes can be obtained by running every adjective in the WordNet adjective index asking for attributes. The result is a list of about 160 unique nouns (synsets) that are used as attributes. Based on the WordNet adjective attributes, we summarize the following list of “visually representable” and “audio representable” properties in Table 4.6 and 4.7 respectively.

<i>Classes of properties</i>	<i>Properties</i>
Space	size, length, width, thickness, height, depth, orientation (direction), shape (form), texture, speed (rate)
Matter	density, state (of matter), appearance, color, quantity (numerousness)
Time	Timing
Human attributes	gender, age
Affection	affection, sensitivity, emotion, personality, quality

Table 4.6: Visually representable properties

Table 4.7 lists the audio representable properties and types of auditory display which can be used to present these properties. Some of them are straightforward; the sound dimension

properties can be presented by all types of auditory display, for instance. Audio presentation of some properties is indirect, especially the matter properties and affections. For example, brittleness could be displayed through auditory icons of break/split sounds, and emotions can be conveyed through music.

<i>Classes of properties</i>	<i>Properties</i>	<i>Types of audio</i>
Sound dimension	frequency (pitch), amplitude (volume), timbre	Auditory icons, speech, music
Spatial relation	position, orientation (direction), speed (rate)	Auditory icons, speech, music
Time	duration, timing	Auditory icons, speech, music
Matter	density, state (of matter), quantity (numerousness), weight, brittleness	Auditory icons
Human attributes	gender, age	Speech, nonverbal expressions, music expressions
Affections	affection, sensitivity, emotion, personality, quality	Speech, music

Table 4.7: Audio representable properties

4.8 Visual semantic representation of prepositions

A significant portion of visual semantics involves the interpretation of spatial prepositions. Prepositions often bear spatial concepts that are crucial to decide the location of entities and movement paths of events in a language visualisation system. The location of an object is usually only relative to positions of other objects in the world, i.e. the spatial relation verbally described by prepositions or implicitly built-in common sense (e.g. *stand* means ‘stand *on the ground*’, the prepositional phrase *on the ground* is part of common sense), and the absolute coordinates of objects are usually irrelevant. The trick of visualising spatial prepositions is that the visualisation is not a one-to-one association, i.e. simply combining one percept with one image. Hence, one preposition may stand for different geometric configurations, and one geometric configuration could be described by different prepositions.

Figure 4.23A shows different prepositions based on the direction of observation of the two objects, and Figure 4.23B shows different geometric configurations based on the direction of observation of the two objects with the preposition “front”. Figure 4.23A is relevant for recognition processes of vision systems like VITRA (Schirra 1993), and Figure 4.23B is relevant for visualisation processes of intelligent presentation systems. Other information such as functional information of the objects involved, internal axes of rotation, direction of movement for moving objects, and surroundings etc. can be used to resolve visual ambiguity of prepositions. For example, to visualise “in front of the blackboard” properly, the functional information of “blackboard” and surrounding information “classroom/meeting room” and the “audience” need to be considered.

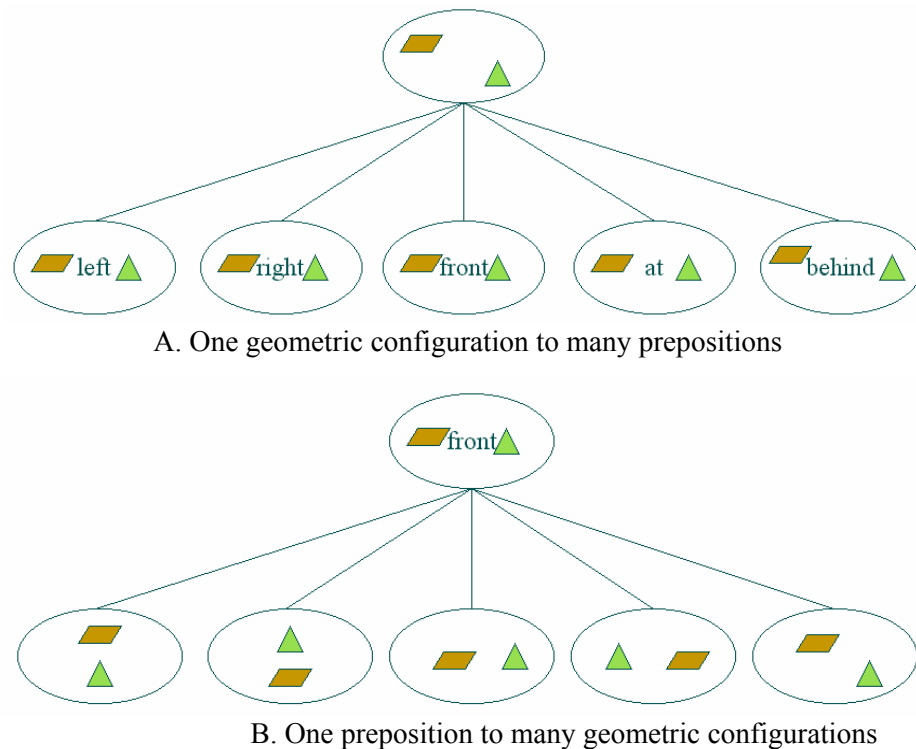


Figure 4.23: The relation between spatial prepositions and geometric configurations

4.8.1 LVSR definitions and semantic features of spatial prepositions

Table 4.8 lists the LVSR definitions of English prepositions. S stands for *spatial* feature, T for *temporal* feature, P for *possession*, R for *ascription of properties*, and U for *unrestricted*. We will explain features such as <contact>, <attach>, and <distributive>. <contact> expresses actual physical contact with another object (see the definitions of *on*, *over* and *against* in the table, and <attach> expresses its attachment to another object (see the definition of “off”). The unmarked value of the feature <contact> is undetermined, since prepositions such as *in*, and *next to* say nothing about contact; whereas the unmarked value of <attach> is <-attach>.

The semantic analysis of prepositions in the above table only shows one primary resolution of lexical conceptual structure. There may be other readings introduced by different conditions. The preposition *into* in the following (1) and (2) can be interpreted as (4), but in (3) it means “to collide with” and should be interpreted as (5).

- 1) John ran into the cave.
- 2) The cockroach ran into the wall.
- 3) John ran into the wall.
- 4) [PATH to [PLACE in<-contact> ([OBJ])]]
- 5) [PATH to [PLACE at<+contact> ([OBJ])]]

<contact> is a critical feature in the PATH of the movement predicate *go* in order to differentiate *impact verbs* (e.g. hit, strike) with *continuous contact movement verbs* (e.g. brush, rub, stroke, and scratch). (6) and (7) presents how to use this feature to represent those two verb types.

- 6) The ball hit the slope.

[EVENT go ([OBJ ball], [PATH to [PLACE on ([OBJ slope])]])]

7) The box slid on the slope.

[EVENT go ([OBJ box], [PATH via<+contact> [PLACE on ([OBJ slope])]])]

<p>about: ST, [PLACE near] above: S, [PLACE over] across: S, [PATH via [PLACE on]] after: ST, [PLACE behind] against: S, [PLACE at], <+contact> along: S, PATH predicate amid: S, PLACE among: S, PLACE around: ST, [PLACE near] as aside: S, PATH, PLACE at: ST, PLACE predicate because of before: T behind: S, PLACE predicate below: S, [PLACE under] beneath: S, [PLACE under] beside: S, [PLACE near] between: ST, PLACE beyond: ST, PLACE but by: ST, [PATH via [PLACE near]] despite down: S, [PATH toward [PLACE at_end_of]] during: T except: for: U from: U, PATH predicate in: ST, PLACE predicate in front of: S, PLACE predicate in terms of inside: S, [PATH to [PLACE in]]</p>	<p>into: S, [PATH to [PLACE in<-contact>]] like near: ST, PLACE predicate next to: S, [PLACE near] of: U off: S, [PATH from [PLACE at<+attach>]] on: ST, PLACE predicate, <+contact> onto: S, [PATH to [PLACE on]] out: S, PLACE predicate out of: S, [PATH from [PLACE in]] outside: S, [PATH to [PLACE out]] over: ST, PLACE predicate, <-contact> past: ST, [PATH via [PLACE near]] round: ST, [PLACE near] since: T, [PATH from [PLACE at]] through: ST, [PATH via [PLACE in]] throughout: ST, [PATH via [PLACE in]], [PLACE in<+dist>] till: T, [PATH to [PLACE at]] to: U, PATH predicate toward: ST, PATH predicate towards: ST, [PATH toward] under: S, PLACE predicate underneath: S, [PLACE under] unlike until: T, [PATH to [PLACE at]] up: S, [PATH toward [PLACE top_of]] upon: S, [PLACE on] via: S, PATH predicate with: SPR within: ST, [PLACE in] without: SPR</p>
---	---

Table 4.8: LVSR definitions of prepositions

4.8.2 Distributive feature for location and motion

Jackendoff (1991) introduced the distributive feature to distinguish distributive location/motion with ordinary location/motion, such as the pairs of “all over” – “on”, “all along” – “along”, “throughout” – “in” (see the definition of “throughout” in Table 4.8: [PLACE in<+dist>]), “all around” – “around”, “all across” – “across”, meaning the subject is distributed in every place in the proper relation to the reference object. Figure 4.24 (1) shows the LVSR definition of these prepositions (or adverbs), (2) and (3) give the comparison of <±dist> location and motion.

4.8.3 Semantic representation of prepositional phrases

This section will lay out some functions of prepositional phrases and clauses. Jackendoff (1990) generalised five basic relations as listed in Table 4.9. The prepositions in italics are used as the

function names for the relation which they indicate, for example, using “with” to indicate accompaniment relation. He uses the function “exchange” to denote the exchange relation.

<i>Relationships</i>	<i>Prepositions</i>	<i>Examples</i>
accompaniment	<i>with</i>	John entered the room <i>with a book under his arm.</i>
purpose & goal	<i>for, to, in order to</i>	The railings were built <i>for our protection.</i>
cause	<i>from, because (of), of</i>	John died <i>from cancer.</i>
means	<i>by, by means of, through</i>	John won the award <i>by performing a trick.</i>
exchange	<i>for</i>	John gave bob £5 <i>for mowing the lawn.</i>

Table 4.9: Jackendoff’s basic conceptual clause modification

The following are examples of some types of modification.

1. John entered the room with a book under his arm.

```
[EVENT go ([HUMAN john], [PATH to [PLACE in [OBJ room]]])
 [with [STATE be ([OBJ book], [PLACE under [OBJ john.arm]])]]]
```

2. John won the award because he worked very hard.

```
[EVENT go_POSS ([OBJ award], [PATH to [HUMAN john]])
 [from [EVENT work([HUMAN john])...]]]
```

3. John gave bob £5 for mowing the lawn.

```
[EVENT go_POSS ([OBJ £5], [PATH from [HUMAN john]
 to [HUMAN bob]])]
 [exch [EVENT mow([HUMAN bob], [OBJ lawn])]]]
```

For visualisation purposes, we classify these relations into three types according to the temporal relations between the main event and subordinate event. The classification illustrated in Table 4.10 simplifies Jackendoff’s modifying subordinate clause functions by means of their temporal relations, because the conceptual relations will be represented by temporal media (i.e. animation). In the table, x denotes the main event and y denotes the subordinate event. Some may argue that the purpose and goal relationship is just intentions and may not occur. We still characterize this relation to the temporal relations $x \{p, m, o, f^{-1}\} y$ since it is not uncommon in visual arts (e.g. movie and animation) to visualise people’s intentions, imaginations and even illusions no matter whether they actually happen or not.

<i>Subordinate relations</i>	<i>Temporal relations</i>
accompaniment, means	$x \{\equiv\} y$
cause, exchange	$y \{p, m, o, f^{-1}\} x$
purpose & goal	$x \{p, m, o, f^{-1}\} y$

Table 4.10: Temporal relations of clause modification

4.9 Semantic field of verbs and prepositions

Different usage of verbs and prepositions can be distinguished by the semantic field features they denote. For example, in the following sentences, the preposition “on” indicates spatial location in 1, and temporal information in 2; the verb “go” and preposition “to” indicate spatial location/motion in 3, possession in 4, and property in 5.

- 1) The vase is *on* the table.
- 2) John arrived *on* the third of April.
- 3) John *went* from London *to* Derry
- 4) The inheritance *went to* John.
- 5) The traffic light *went/changed* from green *to* red.

The precise values of the semantic field feature that a particular verb or preposition may carry are a lexical fact that should be included in its lexical entry. Thus “go” is marked for spatial motion, possession, or ascription of properties; some verbs such as “travel”, “donate”, and “become” are marked for only a single value of the field feature; whereas verbs such as “be” and “keep” are unrestricted. Similarly for prepositions, “on” is marked for both spatial and temporal, “out” is only spatial and “during” is only temporal, but “from” and “to” are unrestricted. Inheriting the unrestricted semantic feature from “from” and “to”, the prepositions “into” and “out of” (see their definition in Table 4.8) are unrestricted too. The following two sentences show their usage in the semantic feature of *material composition* and hence they are not restricted to spatial relations.

1. Nancy broke the bowl into pieces.
into: [PATH to [PLACE in]]
2. John built a house out of bricks.
out of: [PATH from [PLACE in]]

- | |
|--|
| <ol style="list-style-type: none"> 1) all over: on<+dist>
all along: along<+dist>
throughout: in<+dist>
all around: around<+dist>
all across: via on<+dist> 2) The beans were all over the floor.
[STATE be ([OBJ beans],[PLACE on<+dist> [OBJ floor]])]

The beans were on the floor.
[STATE be ([OBJ beans],[PLACE on<-dist> [OBJ floor]])] 3) Nancy ran all over the field.
[EVENT run ([HUMAN nancy],[PATH via [PLACE on<+dist>[OBJ field]])]

Nancy ran across the field.
[EVENT run ([HUMAN nancy],[PATH via [PLACE on<-dist>[OBJ field]])] |
|--|

Figure 4.24: The distributive feature of prepositions

The semantic field features of common English prepositions are listed in Table 4.8. LCS adds features to PLACE predicates to specify which semantic field it indicates in a specific situation like the temporal feature of “at” in Figure 4.25A. Although it could be reasoned from the *Temp* feature of the preposition, this representation does not specify which category [5 o’clock] is. Since we classify category TIME as a separate type of ontology concept in LVSR, the sentence in Figure 4.25 could be represented as B, which is clearer and consistent with the

PLACE format. Hence we may update Figure 4.2A to Figure 4.25C to include the temporal semantic field of prepositions.

Nancy went home at 5 o'clock.

```
[EVENT go ([HUMAN nancy], [PATH to [OBJ home]])
  [PLACE atTemp [5 o'clock]]]
```

A. Jackendoff's LCS representation

```
[EVENT go ([HUMAN nancy], [PATH to [OBJ home]])
  [PLACE at [TIME 5 o'clock]]]
```

B. LVSR representation

C. [PLACE] -> [PLACE placePredicate ([OBJ/TIME])]

Figure 4.25: Semantic representation of temporal information

4.10 Summary

In this chapter, Lexical Visual Semantic Representation (LVSR) is proposed. It consists of nine ontological categories, and is capable of representing the visual semantics of verbs, nouns, adjectives, and prepositions. It contains information about syntactic, semantic (both language and visual), and linking constraints. LVSR is a necessary semantic representation between 3D visual information and syntax because 3D model differences, although crucial in distinguishing word meanings, are invisible to syntax. Temporal information of various semantic relations is encoded in interval relations. Complex events are specified by means of compositional definitions. Adjectives are categorised according to the perceiving senses they require, and LVSR of graded adjectives is also discussed. Entity properties of ascriptive adjectives are studied for visual and auditory presentation. Finally, LVSR of spatial prepositions is given which aids in visualisation of spatial relations. As a consequence, LVSR captures a wide variety of linguistic phenomena, with different types of multimodal information, especially visual information, being localised in different hierarchies, in an encoding that is easy to maintain and extend.

Chapter 5

Natural Language Visualisation

This chapter discusses problems in Natural Language Processing (NLP) which are vital to successful language visualisation and proposes a visual/auditory semantic based verb ontology. Various lexicon-based approaches used for Word Sense Disambiguation (WSD) are addressed. A methodology to extract common sense knowledge on default arguments of action verbs from WordNet to solve the under-specification problem is described. A brief discussion of negation expressions and presentations of negation that are employed in this research is also included.

Natural language understanding usually results in transforming languages from one representation into another, for example, from one language into another in machine translation, or from language to action in natural language interfaces where natural language commands are performed by a machine, or from language to vision in language visualisation. A mapping is designed so that for each event an appropriate action will be performed. A language animation system understands natural language by transforming it from the language medium into visual and auditory media, and the NLP module of the system transforms natural language text into semantic representation.

5.1 Problems in language visualisation

Mapping natural language to semantic representation for visualisation purposes involves various issues such as disambiguation, under-specification, ontological semantics, coreference resolution, spatial relations, and negation. Ambiguity is omnipresent in NLP, and it occurs at various levels in NLP, from syntactic, semantic, lexical, to pragmatic. Typical ambiguity types include word sense, prepositional phrase attachment, parts of speech tagging, and scope of quantification. Lexical ambiguity resolution is the main concern for language visualisation and should combine several information resources (e.g. lexicons) and techniques. Word Sense Disambiguation (WSD) techniques are usually classified to unsupervised and supervised methods. Unsupervised methods use computational lexicons like WordNet and do not need sense annotated corpora for learning, whereas supervised methods disambiguate word senses using information gained from training corpora. We will focus on unsupervised WSD approaches in this chapter.

Under-specification, which is sometimes misleadingly referred to as *vagueness* or ambiguity, is another issue in language visualisation. Language under-specification means that no utterance is capable of containing all the details of the situation it intends to describe. For

example, the sentence “Jane goes shopping every Saturday” leaves an infinite number of questions unanswered. How does she go there, driving a car or walking? And where does she go? Humans intuitively use presuppositions, i.e. common sense knowledge, and inferences to answer these questions in their minds. For computers, in order to fill underspecified roles of an action, a well-designed knowledge base or a lexicon from which the information can be inferred is required. In this chapter we will discuss an algorithm for extracting such common sense knowledge from WordNet.

Ontological semantics is a theory of language semantics and an approach to NLP which uses an ontology as the central resource for extracting and representing semantics, reasoning about knowledge derived from natural language as well as generating natural language texts based on semantic representations. The basic propositional component of meaning is represented by an ontology. The ontology knowledge reflects a person’s model of the world, and the atom of the meaning representation is a taxonomic organization of concepts. In this chapter, we will describe a verb ontology based on visual and auditory semantics which suits multimodal presentation of natural language.

Language animation requires accurate noun phrase coreference resolution to determine which entity a noun phrases in a text refers to. Given that coreference is resolved correctly, it can significantly augment the performance of language animation. We are interested in the narrow tasks of third-person pronoun and lexical anaphor resolution which identify intersentential and intrasentential antecedents of third person pronouns, including nominative (e.g. “he”), accusative (e.g. “him”), possessive case (e.g. “his”), reflexives (e.g. “himself”), and reciprocals (e.g. “each other”, “one another”). Many algorithms for coreference resolution combine syntactic and semantic cues via a set of hard-coded heuristics and filters. We use an anaphora resolution algorithm which applies a syntactic-semantic filter to rule out the noun phrases that are unlikely to be the antecedents. If more than one candidate remains, a salience measure is used and the candidate with the highest salience weight is selected.

Spatial relations between objects and characters in a virtual scene is an important issue in language visualisation. Spatial relations are expressed through prepositions or spatial events such as “surround”. Problems of visualising spatial relations described by natural language lie in three causes: (1) the many-to-many relation between prepositional expressions and spatial relations that we depicted in Chapter 4, Figure 4.23, i.e. one preposition may stand for different spatial relations, and one spatial relation could be described by different prepositions; (2) As natural languages often describe spatial relations at a high level, which means that more details that have to be specified and translated into a program language that computers can understand. This process requires other knowledge such as the referred object’s functionality, e.g. in “sit in front of the piano”, the specified position relates to pianos’ functionality; (3) The spatial relation assigned to virtual objects in a visualised scene can affect other non-spatial properties such as

size. Consider “the kitten in a box”, for instance, the spatial relation “in” determines that the size of the cat must be smaller than the size of the box.

Spatial prepositions were studied in Chapter 4, Table 4.1 and their LVSR definitions are given in Table 4.8. We have identified 12 place relations: *around*, *at*, *behind*, *end_of*, *in*, *in_front_of*, *near*, *on*, *out*, *over*, *top_of*, *under*, and 7 path relations: *to*, *from*, *toward*, *away_from*, *via*, *across*, *along*. We do not regard *left* and *right* as place relations because their positions are quite relative to many factors including the position of the referred object, the observer’s position, or the object functionality, and *left/right* can be defined via the relation *near*. Spatial events are verbs integrating some path or place information, e.g. “enter” integrates the place relation *in* and the path relation *to*, “surround” integrates the place relation *around*. The LVSR of spatial prepositions and events is used to visualise spatial relations.

Automatic detection of negation from natural language text is a task which is often ignored by language visualisation. The dependency grammar that we use gives an advantage to propagate the negation relation to parents of the negative words along the dependency tree. In addition, we use a list of negative particles, adverbs, and counterfactive verbs to aid negation detection.

5.2 Verb ontology and visual semantics

In order to identify the full set of meaning components that figure in the visual representation of verb meaning, the investigation of semantically relevant visual properties and ensuing clustering of verbs into classes needs to be carried out over a large number of verbs. Here we identify three visual factors concerning verb categorisation: (1) *visual valency*, the capacity of a verb to take a specific number and type of roles in language visualisation, (2) somatotopic effectors involved in action execution (visualisation) and perception, which are particular areas of the body, related to the motor area of the brain that controls the movement of different parts of the body and is centred in specific regions of the cortex, and (3) level-of-detail of visual information. Event verbs are categorised according to involved somatotopic effectors, visual semantic roles (e.g. obligatory argument number and classes, humanoid vs. non-humanoid roles), and the level-of-detail they indicate.

Verbs belonging to the same class in our classification are visual “synonyms”, i.e. they should be substitutable in the same set of animation keyframes, through not necessarily in exactly the same visualisation. Visualisation of action verbs could be an effective evaluation of the taxonomy.

5.2.1 Visual valency

Visual valency refers to the capacity of a verb to take a specific number and type of *visual arguments* in language visualisation (3D animation). We call a valency filler a *visual role* and distinguish two types of visual roles: human (biped articulated animate entity) and object

(inanimate entity), since they require different processes in animation generation and are consistent with the HUMAN-OBJ distinction in the LVSR ontological categories discussed in Chapter 4, section 4.3. Visual valency sometimes overlaps with syntactic and semantic valency. The following three examples show the difference among syntactic, semantic, and visual valencies. The number of obligatory roles varies from case to case. Valency fillers in the examples are in italics. It is obvious that visual modalities require more obligatory roles than surface grammar or semantics. What is optional in syntax and semantics is obligatory for visual valency.

1) *Neo pushed the button.*

syntactic valency 2, subject (Neo) and object (the button)

semantic valency 2, agent (Neo) and theme (the button)

visual valency 2, human (Neo) and object (the button)

2) *Jane cut the cloth (with scissors).*

syntactic valency 2, subject (Jane), object (the cloth), optional PP adjunct (with scissors)

semantic valency 2, agent (Jane), theme (the cloth), optional instrument (scissors)

visual valency 3, 1 human (Jane) and 2 objects (the cloth and scissors), all obligatory

3) *Neo is reading (newspaper).*

syntactic valency 1, subject (Neo)

semantic valency 1, agent (Neo), and optional source (newspaper)

visual valency 2, 1 human (Neo) and 1 object (newspaper), all obligatory

Three visual valency verbs subsume both syntactic trivalency verbs such as “give” and syntactic bivalency verbs such as “put” (with goal), “cut” (with instrument), “butter” (with theme, in “butter the toast”) and, an intransitive verb may turn up three visual valency, e.g. “dig” in “he is digging in his garden” involves one human role and two object roles, the instrument and the place.

5.2.2 Somatotopic factors in visualisation

The second visual factor considered in our verb ontology is somatotopic effectors. Psychology experiments prove that the execution, perception and visualisation of action verbs produced by different somatotopic effectors activate distinct parts of the cortex. Moreover, actions that share an effector are in general similar to each other in dimensions other than the identity of the effector. Recent studies (Bergen et al. 2003) investigate how action verbs are processed by language users in visualisation and perception, and prove that processing visual and linguistic inputs (i.e. action verbs) associated with particular body parts results in the activation of areas of the cortex involved in performing actions associated with those same effectors.

On these theoretical grounds, we take effectors into account. However, we only distinguish facial expression (including lip movement) and body posture (arm/leg/torso) in our

verb ontology. Further divisions like distinction between upper/lower arm, hands, and even fingers are possible, but we do not make the taxonomy too fine-grained and reflect every fine visual distinction. Figure 5.1 is an example of using somatopic effectors to classify action verbs “run”, “bow”, “kick”, “wave”, “sing”, and “put”.

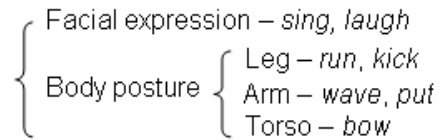


Figure 5.1: Somatopic effectors of some action verbs

5.2.3 Level-Of-Detail (LOD) — Basic-level verbs and their troponyms

Further fine-grained categories of verb ontology can be achieved based on *Level-of-Detail* (LOD). The term LOD has been widely used in relation to research on levels of detail in 3D geometric models. It means that one may switch between animation levels of varying computational complexity according to some set of predefined rules (e.g. viewer perception).

Let’s have a look at the *verbs of motion* in Levin’s verb classes discussed in Chapter 3, section 3.5.3. They subsume two subclasses: *verbs of inherently directed motion* (e.g. “arrive”, “come”, “go”) and *verbs of manner of motion* (e.g. “walk”, “jump”, “run”, “trot”). We find that there are actually three subclasses in *verbs of motion*, representing three LODs of visual information as shown in the tree in Figure 5.2. We call the top level *event level*, the intermediate level *manner level*, and the bottom level *troponym level*. The event level includes basic event predicates such as “go” (or “move”), which are *basic-level verbs* for atomic objects. The manner-of-motion level stores the visual information of the manner according to the verb’s visual role (either a human or a non-atomic object). Verbs on this level are basic-level verbs for human and non-atomic objects. The troponym level verbs can never be basic-level verbs because they always elaborate the manner of a base verb. Visualisation of the troponym level is achieved by modifying animation information (speed, the agent’s state, duration of the activity, iteration) of manner level verbs. The structure in Figure 5.2 is applicable to most troponyms, “cook” and “fry”/“broil”/“braise”/“micro-wave”/“grill”, for example, express different manners and instruments of cooking.

In the following examples, 4a is a LCS-like representation (Chapter 3, section 3.1.3) of “John went to the station”. The predicate “go” is at the event level. The means of going, e.g. by car or on foot, is not specified. Since the first argument of “go” is a HUMAN, we cannot just move John from one spot to another without any limb movement, the predicate “go” is not enough for visualising a human role. We need a lexical rule to change the high-level verb to a basic-level verb, i.e. change “go” to “walk”, when its visual role is human (4b), because walking is the default manner of movement for human beings. In 5 the predicate “run” is enough for visualising the action since it is a basic-level verb for human.

- 4) John *went* to the station.
 a) [EVENT go ([HUMAN john], [PATH to [OBJ station]])]
 b) [EVENT walk ([HUMAN john], [PATH to [OBJ station]])]
- 5) John *ran* to the station.
 [EVENT run ([HUMAN john], [PATH to [OBJ station]])]

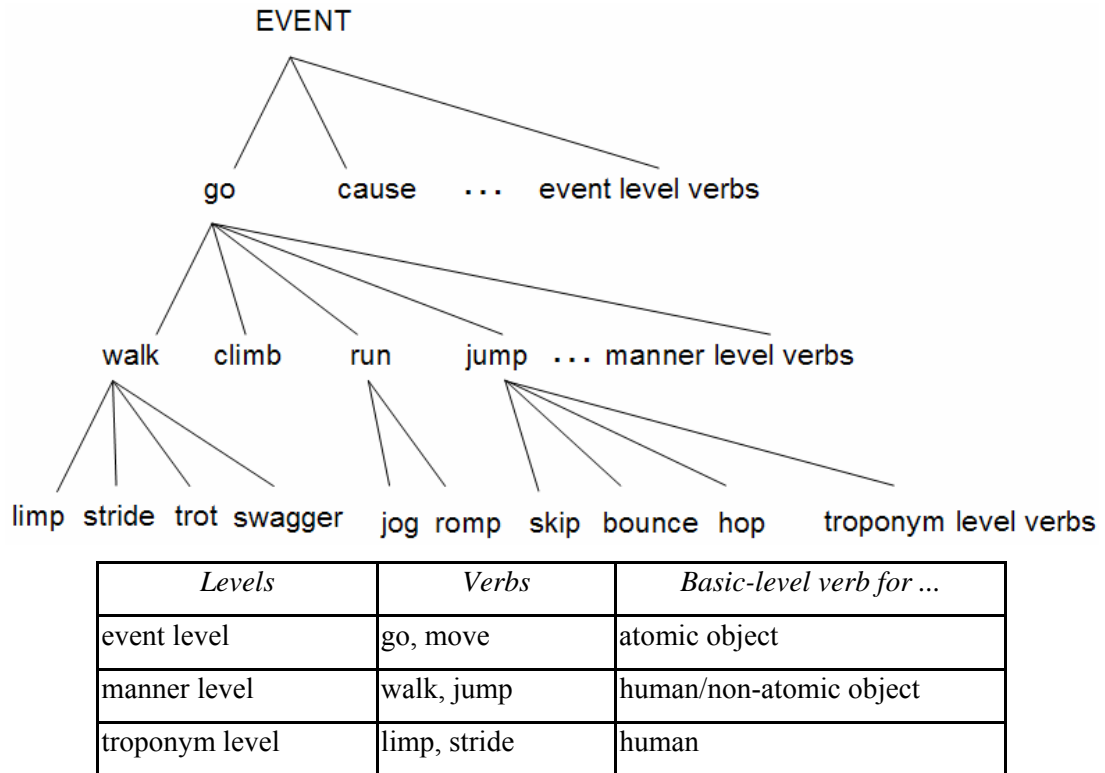


Figure 5.2: Hierarchical tree of verbs of motion

Visualisation processes should support the LOD approach by separating different levels of detail. For instance, the manner-of-motion verbs are stored as key frames of required joint rotations of human bodies in an animation library, without any displacement of the whole body. Therefore “run” is just *running in place*. The first phase of visualisation is finding the action in animation files and instantiating it on the human role in the LVSR representation. This phase corresponds to the manner level (run) in the tree in Figure 5.2. The next phase is to add position movement of the whole body according to the PATH argument. It makes the agent move forward and hence generates a *real* run. This phase corresponds to the event level (go) in the tree.

5.2.4 Visual semantic based verb ontology

The three visual factors discussed in sections 5.2.1-5.2.3 determine a verb ontology shown in Figure 5.3. First we divide all verbs into those occurring on atomic entities and those occurring on non-atomic entities based on whether the objects they act on have sub-components or not. Non-atomic entities are constructed out of collections of objects. In the atomic entities group, we classify the events to those changing objects’ physical location, those changing objects’ intrinsic attributes like shape, size, and colour, and those changing visually unobservable

1. On atomic entities
 - 1.1. Movement/rotation: change physical location (position or orientation), e.g. “bounce”, “turn”
 - 1.2. Change intrinsic attributes such as shape, size, color, texture, and even visibility, e.g. “bend”, “taper”, “(dis)appear”
 - 1.3. Visually unobserved change: temperature change, intensifying
 2. On non-atomic entities
 - 2.1. No human role involved
 - 2.1.1. Two or more individual objects fuse together, e.g. “melt (in)”
 - 2.1.2. One object divides into two or more individual parts
e.g. “break (into pieces)”, “(a piece of paper is) torn (up)”
 - 2.1.3. Change sub-components (their position, size, color, shape etc), e.g. “blossom”
 - 2.1.4. Environment events (weather verbs), e.g. “snow”, “rain”, “thunder”, “getting dark”
 - 2.2. Human role involved
 - 2.2.1. Action verbs
 - 2.2.1.1. One visual valency (the role is a human, (partial) movement)
 - 2.2.1.1.1. Biped kinematics, e.g. “go”, “walk”, “jump”, “swim”, “climb”
 - 2.2.1.1.1.1. Arm actions, e.g. “wave”, “scratch”
 - 2.2.1.1.1.2. Leg actions, e.g. “go”, “walk”, “jump”
 - 2.2.1.1.1.3. Torso actions, e.g. “bow”
 - 2.2.1.1.1.4. Combined actions
 - 2.2.1.1.2. Facial expressions and lip movement, e.g. “laugh”, “fear”, “say”, “sing”, “order”
 - 2.2.1.2. Two visual valency (at least one role is human)
 - 2.2.1.2.1. One human and one object (vt or vi+instrument/source/goal), e.g. “trolley” (lexicalized instrument)
 - 2.2.1.2.1.1. Arm actions, e.g. “throw”, “push”, “open”, “eat”
 - 2.2.1.2.1.2. Leg actions, e.g. “kick”
 - 2.2.1.2.1.3. Torso actions, e.g. “bow”
 - 2.2.1.2.1.4. Combined actions, e.g. “escape” (with source), “glide” (with location)
 - 2.2.1.2.2. Two humans, e.g. “fight”, “chase”, “guide”
 - 2.2.1.3. Visual valency ≥ 3 (at least one role is human)
 - 2.2.1.3.1. Two humans and one object, inc. ditransitive verbs, e.g. “give”, “buy”, “sell”, “show”, and vt. with an instrument, e.g. “beat (sb with sth)”
 - 2.2.1.3.2. One human and 2+ objects (vt + object + implicit instrument/goal/theme), e.g. “cut”, “write”, “butter”, “pocket”, “dig”, “cook”, “put”
 - 2.2.1.4. Verbs without distinct visualisation when out of context
 - 2.2.1.4.1. trying verbs: “try”, “attempt”, “succeed”, “manage”
 - 2.2.1.4.2. helping verbs: “help”, “assist”
 - 2.2.1.4.3. letting verbs: “allow”, “let”, “permit”
 - 2.2.1.4.4. create/destroy verbs: “build”, “create”, “assemble”, “construct”, “break”, “destroy”
 - 2.2.1.4.5. Verbs whose visualisation depends on their objects, e.g. “play” (harmonica/football), “make” (the bed/troubles/a phone call), “fix” (a drink/a lock)
 - 2.2.1.5. High level behaviours (routine events), political and social activities/events, e.g. “interview”, “eat out” (go to restaurant), “call” (make a telephone call), “go shopping”
 - 2.2.2. Non-action verbs
 - 2.2.2.1. stative verbs (change of state), e.g. “die”, “sleep”, “wake”, “become”, “stand”, “sit”
 - 2.2.2.2. emotion verbs, e.g. “like”, “disgust”, “feel”
 - 2.2.2.3. possession verbs, e.g. “have”, “belong”
 - 2.2.2.4. cognition, e.g. “decide”, “believe”, “doubt”, “think”, “remember”, “want”
 - 2.2.2.5. perception, e.g. “watch”, “hear”, “see”, “feel”
3. On either atomic or non-atomic entities, e.g. aspectual verbs: “begin”, “finish”, “stop”, “continue”, “keep”, “give up”

Figure 5.3: Verb ontology based on visual semantics

properties such as temperature change. Together with 2.1.3 and 2.2.1.1.2, the type listed in 1.2 concerns individual deformation (morphing) of an object or parts of an object. The non-atomic group is classified according to whether the respective events concern a character. Verbs happening on characters can next be divided based on whether or not they involve a physical action.

Action verbs are a major part of verbs involving humanoid performers (agent/experiencer) in animation. They can be classified into five categories: (1) one visual valency verbs with a human role, concerning movement or partial movement of the human role, (2) two visual valency verbs (at least one human role), (3) visual valency ≥ 3 (at least one human role), (4) verbs without distinct visualisation when out of context such as trying and helping verbs, (5) high level behaviours or routine events, most of which are political and social activities/events consisting of a sequence of basic actions.

The class of one visual valency verbs (2.2.1.1) is further categorised into *Biped kinematics* (2.2.1.1.1) and *facial expressions and lip movement* (2.2.1.1.2) according to somatotopic effectors. The animation of class 2.2.1.1.1 usually involves body postures or movements, e.g. “walk”, “jump”, “swim”, and the class 2.2.1.1.2 subsumes communication verbs and emotion verbs, and often involves multimodal presentation. These verbs require both visual presentation such as lip movement (e.g. “speak”, “sing”), facial expressions (e.g. “laugh”, “weep”) and audio presentation such as speech or other communicable sounds.

There are two subcategories under the two visual valency verbs (2.2.1.2) based on which type of roles they require. Class 2.2.1.2.1 requires one human role and one object role. Most transitive verbs (e.g. “throw”, “eat”) and intransitive verbs with an implicit instrument or locational adjunct (e.g. “sit” on a chair, “trolley”) belong to this class. Verbs in class 2.2.1.2.2, such as “fight” and “chase”, have two human roles.

Class 2.2.1.3 includes verbs with three or more visual roles, at least one of which is a human role. The subclass 2.2.1.3.1 has two human roles and one or more object roles. It subsumes ditransitive verbs like “give” and transitive verbs with an implicit instrument/goal/theme (e.g. “kill”, “bat”). The subclass 2.2.1.3.2 has one human role and two or more object roles. It usually includes transitive verbs with an inanimate object and an implicit instrument/goal/theme, e.g. “cut”, “write”, “butter”, “pocket”. The visual valency of verbs conflating with the instrument/goal/theme of the actions, such as “cut”, “write”, “butter”, “pocket”, “dig”, and “trolley”, have one more valency than their syntactic valency. For instance, the transitive verb “write” (in “writing a letter”) is a two syntactic valency verb, but its visualisation involves three roles, “writer”, “letter”, and an implicit instrument “pen”, therefore it is a three visual valency verb.

There is a correlation between the visual criteria and lexical semantics of verbs. For instance, consider the intransitive verb “bounce” in the following sentences. It is a one visual valency verb in both 6 and 7 since the PPs following it are optional. The visual role in 6 is an

object, whereas in 7 it is a *human* role. This difference coincides with the two word sense differences in WordNet. The “bounce” in 6 means “spring back; spring away from an impact”, and in 7 it means “move up and down repeatedly”.

- 6) The ball *bounced* over the fence.

WordNet sense #: 01837803. Spring back; spring away from an impact.

Hypernyms: jump, leap, bound, spring

Verb class 1.1 in Figure 5.3

- 7) The child *bounced* into the room.

WordNet sense #: 01838289. Move up and down repeatedly.

Hypernyms: travel, go, move

Verb class 2.2.1.1.1 in Figure 5.3

Verbs applied to atomic entities can also apply to non-atomic objects, for instance, “disappear”/“vanish” can apply to both atomic and non-atomic objects (the Cheshire cat in the following example 8) or component(s) of non-atomic objects like “turn” applying to the human head in example 9.

- 8) ‘All right,’ said the Cheshire Cat. And this time it *vanished* quite slowly, beginning with the end of its tail and ending with its grin.
- 9) He *turned* his head and looked back.

Non-action verbs (class 2.2.2 in Figure 5.3, e.g. “feel”) and verbs with metaphor meanings (e.g. “We *breezed* through the test.”) are not easily observable. They are difficult to describe by physical changes but are nevertheless common in language use. In performance art, they are often expressed by face expressions, body poses, or more straightforward, by speech modality. Static language visualisation is used to express mental activity by thinking bubbles. In a multimodal presentation system, these types of verbs could be presented via speech modality if not obvious in visual modality.

5.3 Verb ontology and audio semantics

In human commonsense knowledge, there is a natural mapping between audio and objects, events, status, and emotions. We discuss relations between lexical semantics and the audio modality in this section. They will aid multimedia allocation by providing an audio semantic based verb ontology where a media allocator can assign verbs in certain categories to specific auditory display. Various English verb classifications have been analyzed in terms of their syntactic and semantic properties, and conceptual components, such as syntactic valency, lexical semantics, semantic/syntactic correlations, and visual semantics that we discussed in section 5.2. Here the audio semantics of verbs, particularly their sound sources, is investigated.

The verb ontology shown in Figure 5.4 represents a classification of sound emission verbs based on audio semantics. First, we divide sound emission verbs into three classes: 1)

sounds made by one object, 2) audio expressions of human, and 3) verbs of impact by contact, i.e. sounds made by two objects, based on sound source. In the first class, we classify the verbs to those emitting typical sounds of a particular object (class 1.1), sounds made by animals (class 1.2), those emitting break/split sounds (class 1.3), and weather verbs which emit environmental sounds (class 1.4). Class 2 includes sounds made by human, either speech (class 2.1) or nonspeech expressions (class 2.2). Nonspeech expressions are composed of nonverbal expressions such as “laugh”, “sign” (class 2.2.1), musical expressions such as “hum”, “sing” (class 2.2.2), auditory gestures such as “clap”, “snap” (class 2.2.3), and hiccup/breathe verbs such as “fart”, “sneeze” (class 2.2.4). Class 3, the verbs of impact, includes *nonagentive verbs of impact* (class 3.1), e.g. “My car *bumped* into the tree”, *contact of an instrument and an object* (class 3.2), and *contact of body part and an object* (class 3.3).

The importance of the audio modality varies from class to class. For instance, audio is indispensable for the class 2.1 (speaking or manner of speaking) and class 2.2.2 (musical expressions), whereas it is merely an addition to the visual modality for the class 3 (verbs of impact by contact). This information can be used in media allocation and animation generation, for example, verbs of speaking or manner of speaking (class 2.1) cause the part enclosed in quotation marks in a sentence to be assigned to speech modality with simultaneous lip movements of the speaker in generated 3D animation.

- | |
|---|
| <ul style="list-style-type: none"> 1. Sounds made by one object <ul style="list-style-type: none"> 1.1 Typical sounds of a particular (object) source, e.g. “toll”, (gun) “fire”, (clock) “tick”, “trumpet” 1.2 Sounds made by animals, e.g. “baa”, “bark”, “quack”, “tweet” 1.3 Break/split verbs, e.g. “break”, “crack”, “snap”, “tear” 1.4 Weather verbs, e.g. “storm”, “thunder” 2. Audio expressions of human <ul style="list-style-type: none"> 2.1 Verbs of speaking or manner of speaking, e.g. “say”, “order”, “jabber”, “shout” 2.2 Nonspeech expressions <ul style="list-style-type: none"> 2.2.1 Nonverbal expressions, e.g. “laugh”, “giggle”, “moan”, “sign” 2.2.2 Musical expressions, e.g. “hum”, “sing”, “play” (musical instruments) 2.2.3 Gestures, e.g. “clap”, “snap” 2.2.4 Hiccup/breathe verbs, e.g. “fart”, “hiccup”, “sneeze”, “cough” 3. Verbs of impact by contact <ul style="list-style-type: none"> 3.1 Nonagentive verbs of impact (by contact of two objects), e.g. “bump”, “crash”, “slam”, “thud” 3.2 Contact of one instrument and one object, e.g. “strike” (with a stick) 3.3 Contact of body part and one object, e.g. “kick”, “knock”, “scratch”, “tap” |
|---|

Figure 5.4: Verb ontology for audio semantics

5.4 Word Sense Disambiguation

In human natural languages, many words have multiple senses. Word sense disambiguation (WSD) is the process of determining the correct sense of a word in context. WSD is a fundamental problem in NLP, and important for most NLP applications such as machine

translation, information retrieval/extraction, and language animation. There are four knowledge-based WSD techniques we use:

1. Grammatical relations coded as subcategorisation frames.
2. Semantic relations coded as the presence of arguments in LCS database.
3. Lexical relations coded in WordNet, including hyponymy/hyperonymy, antonymy, and meronymy.
4. Statistical information

The theta roles information of verbs from the LCS database (see Chapter 3, section 3.4.3) is the key to disambiguate word senses. For example, the verb “leave” has the following two senses:

1. Verbs of Inherently Directed Motion / AWAY_FROM-AT

Example: They ~ed the scene

Theta roles: Theme V [Source]

LCS: (go loc (* thing 2) (away_from loc (thing 2) (at loc (thing 2) (* thing 4))) (abdicate+ingly 26))

2. Verbs of Future Having

(a) Example: He ~ed money (to John)

Theta roles: Agent V Theme [(to)_Goal]

LCS: (cause (* thing 1) (go poss (* thing 2) ((* to 5) poss (thing 2) (at poss (thing 2) (thing 6)))) (advance+ingly 26))

(b) Example: He ~ed John money

Theta roles: Agent V Goal Theme

LCS: (cause (* thing 1) (go poss (* thing 2) ((to 5) poss (thing 2) (at poss (thing 2) (* thing 6)))) (advance+ingly 26))

The first line in each sense is the Levin verb class that this word sense belongs to. Then an example usage of each word sense is given, and the “~ed” indicates the past tense of the verb. In “Theta roles”, the arguments bracketed in [] are optional, and particles in brackets are the preposition of the role. Finally, the LCS specification of the word sense is given. The numbers in the specification denote some logical arguments or modifiers, e.g. 2 in the LCS specification of sense 1 means the logical argument THEME. Appendix C gives a complete description of the LCS notation.

The number and type of arguments in a verb sense entry can be used for disambiguation. For example, in the sentence “John left the house”, the sense 2(b) can be ruled out because the number of theta roles does not match. To further disambiguate between sense 1 and sense 2(a), frequency information is considered. There are two readings of this sentence. One interprets “leave” as a verb of motion, the other interprets it as a verb of change of possession, like “John left the house (to Mary)”, regarding “the house” as an inheritance, for instance. The frequency information in WordNet shows that the first reading is more probable than the other. We use frequency information to reorder entries for the same verb in the LCS database, i.e. to move the most frequently used word sense first and the least frequent sense last, and always select the first matched word sense in the transformation to semantic representation. In the above example, the word sense 1 of “leave”, which is the correct meaning, is chosen.

The statistical information needed is the *number of senses of lemma* in WordNet, which are ranked according to their frequency of occurrence in semantic concordance texts. We use this information to reorder the entries in the LCS database. It is a useful means for word sense disambiguation because the order in which senses are presented is important for a semantic analyser. Verb entries in the LCS database are ordered according to Levin's verb classes. Using its index to WordNet sense number, we reorder the database based on sense frequency in WordNet. For example, the verb "leave" has 9 entries in the LCS database, referring to Levin's 7 verb classes and 14 senses in WordNet 2.0. Here is the original order of "leave" in the LCS database:

```
:CLASS "13.3"
:NAME "Verbs of Future Having"
:THETA_ROLES ((1 "_ag_th,goal(to)"))
:SENTENCE "He !!+ed money to John"

:CLASS "13.3"
:NAME "Verbs of Future Having"
:THETA_ROLES ((2 "_ag_goal_th"))
:SENTENCE "He !!+ed John money"

:CLASS "13.4.1.a"
:NAME "Verbs of Fulfilling - Possessional / -to"
:THETA_ROLES ((1 "_ag_th,goal(to)"))
:SENTENCE "He !!+ed the money to John"

:CLASS "13.4.1.b"
:NAME "Verbs of Fulfilling - Change of State / -with"
:THETA_ROLES ((1 "_ag_th,mod-poss(with)"))
:SENTENCE "He !!+ed John with the money"

:CLASS "13.5.1.a"
:NAME "Get - No Exchange"
:THETA_ROLES ((1 "_ag_th,src()),ben(for)"))
:SENTENCE "He !!+ed a flower (from/off the wall) (for Mary)"

:CLASS "13.5.1.a"
:NAME "Get - No Exchange"
:THETA_ROLES ((2 "_ag_ben_th,src()"))
:SENTENCE "He !!+ed Mary a flower (from/off the wall)"

:CLASS "15.2.a"
:NAME "Keep Verbs"
:THETA_ROLES ((1 "_ag_th,loc()"))
:SENTENCE "She !!+ed the book in the shelf"

:CLASS "51.1.a"
:NAME "Verbs Inherently Directed Motion / -from/to"
:THETA_ROLES ((1 "_th,src(from),goal(to)"))
:SENTENCE "The dog !!+ed (from the house) (to the street)"

:CLASS "51.1.d"
:NAME "Verbs of Inherently Directed Motion / AWAY_FROM-AT"
:THETA_ROLES ((1 "_th,src"))
:SENTENCE "They !!+ed the scene"
```

After we rearrange these entries based on WordNet sense frequency, the order becomes: Levin's class 51.1.a, Verbs Inherently Directed Motion /from/to

Levin's class 51.1.d, Verbs of Inherently Directed Motion /AWAY_FROM-AT

Levin's class 15.2.a, Keep Verbs

Levin's class 13.3, Verbs of Future Having

Levin's classes 13.4.1.a, b, 13.5.1.a, Verbs of Fulfilling, Get-No exchange

To find out the advantages of the rearrangement, let's consider the sentence "John left the gym". The verb "leave" has two arguments: "John" the agent and "gym" the theme. Consulting the theta roles information of the above 9 entries, a semantic analyser can rule out 3 of the entries (see Table 5.1), which leaves 6 entries. The statistical order of the revised database helps the semantic analyser choose the most frequent sense from the 6 possible entries (13.3 theta roles 1, 13.4.1.a, b, 13.5.1.a theta roles 1, 15.2.a, and 51.1.d), i.e. the class 51.1.d — *Verbs of inherently directed motion /away_from-at*, which is the correct sense in this case.

<i>Theta roles of the entries</i>	<i>Reasons to rule out the entry</i>
13.3, THETA_ROLES ((2 "_ag_goal_th"))	Requires three arguments at least
13.5.1.a, THETA_ROLES ((2 "_ag_ben_th,src()"))	Requires three arguments at least
51.1.a, THETA_ROLES ((1 "_th,src(from),goal(to)"))	At least one role. Other role(s) must follow the specified prepositions

Table 5.1: Word senses of "leave" ruled out by semantic analyser

We consider not only thematic roles of verbs, but also the conceptual hierarchy of a word obtained through the WordNet semantic network — as a means for generalization. It is possible to disambiguate a word using the logical arguments and modifiers of the LCS field (see Chapter 3, section 3.4.3 and Appendix C) in the LCS database and WordNet semantic network. For instance, in "he left the country", using the conceptual hierarchy of WordNet and LCS entries of "leave", we are able to successfully disambiguate the verb sense. This is done via the generalization learned from the semantic network of "country":

```
country, state, land
=> administrative district, administrative division, territorial division
=> district, territory, territorial dominion, dominion
=> region
=> location
```

This hypernym tree allows us to infer a more general relation "leave a location". The entry of "leave" with the word sense "inherently directed motion / away_from-at" (Levin's class 51.1.d) specifies

```
:LCS (go loc (* thing 2)
      (away_from loc (thing 2) (at loc (thing 2) (* thing 4)))
      (leave+ingly 26))
```

The second argument, (* thing 4), indicates a source of "Logical argument Paths FROM and Path AWAY_FROM", i.e. where the THEME, (* thing 2), started its motion (in LOC), or what its original state (IDENT) was¹ (e.g. John left *the house*). It is a location indicating

¹ This definition is taken from the LCS database documentation (see Appendix C for LCS notation).

source of the movement, which fits the inference of “country”. Hence we can disambiguate this word sense from 13.4.1 and 13.5.1 which involve change of possession.

5.5 Commonsense reasoning using WordNet

For animation generation it is necessary to explicitly generate implied actions, instruments (means), goals, or even themes which are underspecified in language input. The lexicon is the focal point where these problems are resolved so that a continuous, correct animation can be achieved. A lexical entry can contain, for example, information about the orthography, syntax and semantics of a word. We have discussed various lexicons, the lexical information they store, and how they organise it in Chapter 3, section 3.4. Here, we examine how to utilize existing lexicons to fill underspecified thematic roles such as default arguments of verbs.

One limitation of WordNet is that it has neither predicate argument structure, nor explicit constitutive and functionality information which are important for commonsense reasoning in language visualisation. This is why the LCS database is used to enhance semantic analysis. However, relations in WordNet do present additional semantic information implicitly. For example: the hypernym tree of “lancet” in Figure 5.5 contains (1) domain, (2) constitutive, (3) purpose, and (4) agentive information. This feature makes lexical inference with WordNet possible, i.e. default arguments may be extracted by inference programs. In the “lancet” example, the purpose information can link the noun to the verb “cut” as a possible instrument.

```
lancet
=> surgical knife (1)
  => knife
    => edge tool (2)
      => cutter, cutlery, cutting tool
        => cutting implement (3)
          => tool
            => implement
              => instrumentality, instrumentation
                => artifact, artifact (4)
                  => object, physical object
                    => entity
```

Figure 5.5: The hypernym tree of “lancet” in WordNet

Language visualisation requires lexical/common sense knowledge such as default instruments (or themes) of action verbs, functional information and usage of nouns. In Table 5.2, the default instruments (or themes) are the highest nodes of the hypernymy (is_a) tree in WordNet, all of whose children are possible instruments. We start from an acceptable candidate (an instrument/theme in this case), then propagate upward, and check if all the hyponyms of this lexical item are acceptable. If this is the case, we continue the propagation until we reach a level at which at least one of whose hyponyms is not acceptable.

For example, “knife” is a possible instrument for the verb “cut”. We propagate its hypernym tree in WordNet (Figure 5.6), and find that all hyponyms of “edge tool” are acceptable instruments of cutting, same for “cutter, cutlery, cutting tool”, and “cutting

implement”. But when we propagate one more level, we find that some hypernyms of “tool”, e.g. “drill”, “comb”, cannot be used for cutting. Therefore the “cutting implement” is the highest node of possible instruments for the verb “cut” and should be stored as a default argument in its lexical entry.

<i>Verb</i>	<i>Default instrument/theme</i> (highest node of possible candidates in WordNet)	<i>Example instrument/theme</i>
cut	cutting implement	knife, scissors, lancet
bake	oven	oven
fry	pan	frying pan
boil	pot	kettle, caldron
drive	self-propelled vehicle	car, tractor, van
write	writing implement	pen, chalk
adorn	decoration, ornament, ornamentation	flower, jewel

Table 5.2: Default instruments of verbs

This approach provides a flexible specification of lexical knowledge, avoiding over-specific specifications. Consider the following examples 10-12, if we store “knife” as the default instrument in *cut*’s entry, it might not be appropriate for (12), whereas “cutting implement” suits all the cases.

- 10) John cut the bread. (bread knife)
- 11) The doctor cut the cancer from some healthy tissue around it. (lancet)
- 12) John cut the crop. (scythe)

```

knife
=> edge tool
=> cutter, cutlery, cutting tool
=> cutting implement
=> tool
=> implement
=> instrumentality, instrumentation
=> artifact, artefact
=> ...

```

Figure 5.6: Hypernym tree of “knife” in WordNet

This selection algorithm could be automated based on corpus data and linguistic ontologies. A generative lexicon with this knowledge provides the capability of visualising various activities without hardcoding them as part of an animation library.

However, there is a possibility that some acceptable candidates of default instrument/theme might not have the highest nodes of possible instruments/themes in their hypernym trees. The verb “adorn” in Table 5.2, for instance, has “flower” as one of its possible instruments/themes. However, we cannot find the highest node of possible candidates “decoration, ornament, ornamentation” in the “flower” hypernym tree, whereas it can be found in the hypernym tree of “jewel” or “flower arrangement” (Figure 5.7). The need to start searching from an appropriate candidate increases the complexity of searching for default arguments in WordNet.

In this section we have argued that language visualisation relies on lexical knowledge, such as the default arguments of verbs, which may not be included in existing computational lexicons. A selection algorithm based on WordNet is used for finding the highest hypernym of default instruments/themes of verbs.

<pre>flower => angiosperm, flowering plant => spermatophyte, phanerogam, seed plant => vascular plant, tracheophyte => plant, flora, plant life => organism, being => living thing, animate thing => object, physical object => entity</pre>
<pre>jewel => jewelry, jewellery => adornment => decoration, ornament, ornamentation => artifact, artefact => object, physical object => entity</pre>
<pre>flower arrangement => decoration, ornament, ornamentation => artifact, artefact => object, physical object => entity</pre>

Figure 5.7: Hypernym trees of “flower”, “jewel”, and “flower arrangement”

5.6 Negation

There are many theories about negation in language. Expressions with a negative element usually convey negative meaning. However, they do not always lead to negative interpretation. Phrases such as “cannot help”, “cannot resist” (means “like”) give a positive interpretation. A hardcoded set of such phrases can be used to distinguish them from other real negative expressions.

We identify three types of negation in regard to events: propositional negation, counterfactive negation, and double negatives. Propositional negation is introduced by negative particles (e.g. “not”, “nor”, “neither”) or adverbs (e.g. “never”, “barely”, “hardly”, “rarely”, “seldom”, “no longer”). For example, the negation in “Mary didn’t marry John” is expressed by the negative particle “not”, and the negation in “Mary can hardly hear the noise” is introduced by the negative adverb “hardly”. Counterfactive negation is introduced by negative lexical meanings or morphological means (i.e. negative affix such as “un-”). These events introduce a presupposition about the non-veracity of its argument. Typical counterfactive verbs include “forget to”, “unable to” (in past tense), “prevent”, “cancel”, “avoid”, “decline”, “fail”, which indicate the events following them did not happen. For instance, “John forgot to buy birthday gifts for her”, “Mary was unable to pass the exam”, and “John prevented the divorce” are counterfactive. Double negatives express events which actually happened/occurred, for example, “John didn’t forget to buy some gifts” means that John bought some gifts.

To identify negations, a language knowledge base can have a list of negative particles (i.e. the n-words), negative adverbs (e.g. “never”) and counterfactive verbs (e.g. “forget”). For the first two types of negation in regard to events, narration (i.e. a narrator’s voiceover) can be used to present the negation because the events mentioned in the sentences did not actually happen; whereas double negative expressions are transformed into positive expressions before they are represented in LVSR and sent for animation generation.

Negative meaning in quantifiers is introduced by the group of n-words such as “none”, “nobody”, “nothing”, and “no one”. Negative propositions concern attributes which can be rewritten using contrary formation, e.g. “Jane is not happy.”-> “Jane is depressed/sad/angry.”; “The king is not bald.”-> “The king has hair.” Visual presentation of these negative statements depends on that of their antonyms.

5.7 Summary

In this chapter, the problems of NLP which are vital to language visualisation were introduced, and based on visual and auditory semantics, a verb ontology was proposed. We introduced the notion of *visual valency* and use it as a primary criterion to re-categorise event verbs for language visualisation. This chapter also discussed various lexicon-based approaches used for WSD. More precisely, the context and the senses of ambiguous verbs are analysed using hypernymy relations and word frequency information in WordNet and thematic roles in LCS database. We proposed a methodology to extract common sense knowledge on default arguments of action verbs from WordNet to solve the underspecification problem and meet the needs of explicit information required in language visualisation. Identification and presentation of various negative expressions in language were also discussed. The next chapter discusses automatic generation of animated 3D worlds.

Chapter 6

3D Animation Generation

Automatic generation of 3D animation incorporates design expertise and automated selection, creation and combination of graphical elements, and most crucial, control of 3D character animation. Most movies and games involving 3D characters use commercial software to produce character animations. Keyframe is the most popular animation technique, though some current software offers other facilities such as Inverse Kinematics (IK) and dynamic simulation. Our work on animation generation involves how to use precreated keyframes to produce animation based on LVSR. This chapter presents a framework for authoring 3D virtual environments for language visualisation. Various issues of 3D animation are discussed. First, the structure of an animation producer is delineated. We then discuss virtual human and 3D object modelling, followed by collision detection for simulating realistic and natural object behaviour. Finally, automatic camera placement and control in virtual storytelling is explored.

6.1 The structure of animation generation

An animation generation module can accept an LVSR representation and produce 3D animation with camera behaviour that conforms to a set of cinematic principles. The animation producer concerns two functional models: the world model and the body model, according to the elements of theatre art, i.e. performers, sets, costumes, lights, makeup, sound, audience, what is performed, and environment. As shown in Figure 6.1, an animation producer can consist of *actor manager*, *world builder* and *graphic library*. The world builder simulates the world model, i.e. it sets up the stage: background, environmental objects (including lights and sound) and props. A set is a tiled background layer which can be grassland, water, or gravel ground. The actor manager simulates the body model, i.e. it creates virtual actors. It also manages speech and motion of virtual characters. The graphic library contains reusable graphic components of the knowledge base (e.g. actors, props, tiles, and animations). It is important to reuse sets, props and actors in other applications since they are built on the reusable components in the graphic library. The virtual world can be structured this way with the aim of increasing reusability. Since the actor's speech and motion may have implications on what is happening on the stages and props, the *world builder* exchanges information like spatial relationships with the *actor manager*.

6.2 Virtual human animation

Virtual human animations are separated into facial animation and body animation. Facial animation results primarily from deformations of the face. Body animation generally involves animating a set of joints which construct the virtual human's skeleton. Here, we mainly focus on body animation.

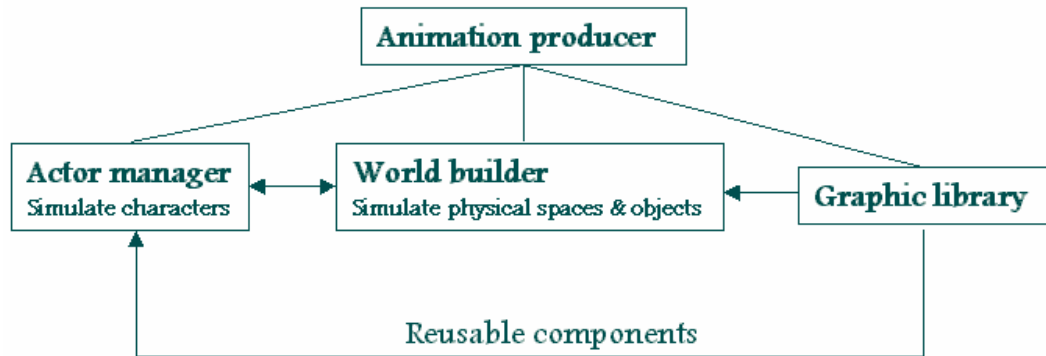


Figure 6.1: Structure Diagram of an *animation producer*

6.2.1 Animated narrator

An animated narrator is an intelligent interface agent who tells stories beside the virtual environment. The requirements of a narrator are less than characters in multimodal storytelling, and therefore the number of body movement types that a narrator is required to carry out is much less than that of characters. In Larsen and Petersen's (1999) interactive stories reviewed in Chapter 2, section 2.3.1, the narrator is only required to react to physical events (e.g. users queries or commands), but not to virtual events (e.g. reaction of his sense of sight in the virtual world). This is due to the fact that the narrator's role is not as an actor in the world, but a side-speaker outside the world. Even for characters in a virtual world, reaction to the virtual world events is not required because virtual events and their corresponding reaction can be found either in the input story or requests for user input. For instance, in Larsen and Petersen's stories when a character (not the user controlled avatar) "sees" (perceives) another character or a prop, he might have some reaction such as surprising expression or a greeting through sound modality, which is decided by his behaviour model. This mechanism enables a story to be more dynamic and characters can stand on their own. In language-to-animation conversion, the events of encountering (a character's perception) and his corresponding response are provided by the input text.

In addition, the narrator's voiceover has another significant role besides introducing the storyline. It may cover the information that cannot be presented successfully by other modalities. For example, to present the event "marry" in the cliché story ending, "They married and lived merrily thereafter" conventional authoring media like movies or cartoons may use a shot of wearing a wedding ring on the bride's ring finger or the couple going to church in their wedding dress, but for intelligent storytelling systems the event "marry" is difficult to present.

Even if the procedures of weddings are modelled in the knowledge base, for instance, in the script form as discussed in Chapter 3, section 3.1.2, there are still problems with respect to graphics, such as simulating clothes which involves changeable dressing of virtual humans and cloth deformation and self-collision detection for dynamics. The verb “marry”, like “interview”, “eat out” (go to restaurant), “call” (make a telephone call), and “go shopping”, is a routine event in the class 2.2.1.5 in the verb ontology shown in Figure 5.3. Verbs in this class require either an algorithm to choose available contents (i.e. 3D models and animations) to convey certain concepts, or a specified script defining the routine in a knowledge base. When a verb of class 2.2.1.5, which is difficult to be visualised via straightforward physical movements of characters, is detected, the narrator’s voiceover will do the job by speaking this sentence with the accompanying animation showing the couple is together.

6.2.2 Agents and avatars – how much autonomy?

Rather than regard actors in stories as autonomous agents as in Loyall (1997), a 3D human in virtual storytelling is just a character controlled by the *actor manager* (originally by input text) while the narrator who tells the story is regarded as an interface agent.

Cavazza et al. (1998) proposed four classes of virtual actors: pure avatars/clones, guided actors, autonomous actors, and interactive-perceptive actors. We classify virtual humans into two types based on their autonomy: autonomous agents and avatars. Autonomous agents have higher requirements for behaviour control, sensing, memory, reasoning, and planning, whereas avatars require fewer autonomous actions since their behaviour is “player-controlled”. Therefore, autonomous agents’ behaviours are usually built on a sense-control-action structure that permits reactive behaviours to be locally adaptive to the environment context. They need to constantly re-sense their environments and re-evaluate their course of action. With these agents virtual humans are beginning to exhibit the early stages of autonomy and intelligence as they make decisions in changing environments and react to other virtual humans and real people rather than carrying on fixed movements regardless of environment context. However, there are serious limitations on the degree to which this is feasible since rendering can consume arbitrary amounts of CPU time.

The high cost of sensory processing can be saved by symbolic AI systems where all information is available in a centralised database. Consider the example of “chasing”¹, for autonomous agents in computer games, the system usually re-senses the properties of all the objects/characters in the scene, reruns all of its inference rules, and feeds the output of the inference rules back to the motor systems of the follower to retarget the goal and avoid obstacles on each cycle of its control loop, so the follower can continually adjust his course as the goal

¹ Chase verbs, Levin’s (1993) class 51.6.a, includes words “chase”, “follow”, “pursue”, “shadow”, “tail”, “track”, “trail”.

changes his course. But virtual characters in storytelling systems need not do so because the moving target, whether a virtual human or an animal, is controlled by the system rather than a player. The course of the target's movement is known by the system and stored in the centralised database. Therefore, virtual characters in a non-interactive storytelling system are somewhere in between an autonomous agent and an avatar, as shown in Figure 6.2.

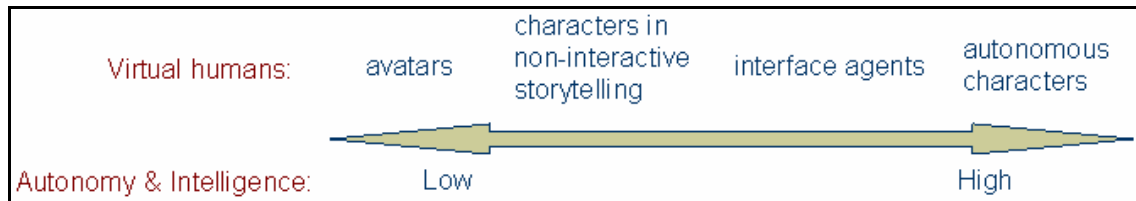


Figure 6.2: Autonomy of virtual humans

Because most of a character's behaviours and responses to the changing environment are described in story input and controlled by a storytelling system, it does not require sophisticated reasoning and behaviour control. The character neither receives world information from his own sensors nor calculates and saves perceptual status in his memory, since the world information of the virtual story world is already available in the system database. However, simple behaviour planning on path plans, e.g. to avoid obstacles, is still required, whereas it is not needed in avatars. This planning is performed by a controlling model rather than by the character itself. Therefore, the autonomy requirement of characters in a virtual story world is greater than that for avatars and less than that for agents.

Since our task is to produce neither autonomous agents nor avatars, but virtual actors whose behaviours are tightly coupled with verbal behaviours in story narratives, it requires more reasoning and inferences in natural language understanding than agent autonomy. For example, to infer that the preconditions of the action "giving" in the sentence "A gives x to B" are: (1) "A has x" (possession) and (2) "A is at B" (location).

6.2.3 Animating human motions

Simulating human motion and behaviour by computer involves not only animation techniques but also many subtle details in terms of believability, aesthetics, and physical fidelity such as deformable body modelling and the kinematics and dynamics of the figure. Here, we focus on humanoid animation techniques. Existing virtual human animations are either controlled by pre-created animations (e.g. hand-animated using authoring tools like 3D Studio Max (3ds Max 2005), Maya (Maya 2005), and Poser (Poser 2005), or motion captured data), or dynamically generated by animation techniques such as inverse kinematics (IK).

The most common kinematics animation technique is keyframing. In keyframing animation, virtual humans are moved by animators, who specify explicit definition of the key values of the virtual human's joints and body positions at specific time instants, namely "keyframes". Then the key values are interpolated by the computer so that in-between frames

are generated and the animation is rendered frame by frame. Key frames are usually hand-created using authoring tools such as Character Studio (a plug-in of 3D Studio Max) or converted from motion-captured data. Motion capture consists of measurement and recording of direct actions of a real person or animal for analysis and playback. The technique involves mapping of measurements onto the motion of the virtual human. Motion capture methods offer natural motions. Consider the case of walking. A walking motion is recorded and applied to a virtual character. It provides a good motion, because it comes directly from reality. However, for any new motion, it is necessary to record the reality again. Hence, we use hand-created keyframe animation for virtual human motions in language visualisation.

Kinematics animation techniques include forward kinematics and inverse kinematics. When a bone is moving as a parent or a child in a hierarchical skeleton tree, it passes the movement to its attached parents or children using forward kinematics or inverse kinematics. Forward kinematics is a system in which the transforms of the parent in a hierarchical tree structure are passed on to the children, and animation is performed by transforming the parent. IK is a system in which the movement of the children is passed back up the chain to the parent in the hierarchical skeleton tree. Given a desired position and orientation for a final link in a hierarchy chain, IK establishes the transformations required for the rest of the chain. Animation is performed by affecting the ends of the chain, e.g. in biped walking animation, by moving the foot and the shin, knees and thighs rotate in response. IK models the flexibility and possible rotations of joints and limbs in 3D creatures. IK adds flexibility which avoids canned motion sequences seen in keyframe animations, and hence enables having an infinitely expandable variety of possible animations available to a virtual character. The character control file is also reduced to a physical description of the character and a set of behavioural modifiers that are used to alter the flavour of the animations (Badler et al. 1993).

There are some basic requirements that are needed for virtual actors which are overlooked by conventional storytelling arts like drama and film. This is because these requirements are taken for granted in those arts. A movie scripter, for example, does not have to specify that his characters should be able to speak while walking, or to make a detour to avoiding collision with obstacles or other characters because all human actors who perform the script can talk at the same time they walk and walk on a proper path. When trying to synthesize animated actors, this is a luxury that we do not have. If we want these actors to have the basic facilities that every actor or living creature has, then we must build those properties into them. To make the movement of the actors appear believable, e.g. circumventing objects and avoiding collision with other actors, *path planning* involves obstacle avoidance and collision detection.

Behaviour models are a feasible way to create an autonomous actor in an interactive story generating process. The more complex behaviours given to the actors, requires less scripting in story input. Behaviour models are suitable for users' real-time control in interactive stories. For instance, in usual graphic games when an avatar enters the virtual world the first

time, the player is required to set his/her internal values of personality like emotion, intelligence, strength, endurance. Later the avatar's behaviour in the world is decided partially by his personality, status, and capabilities. For virtual humans in language visualisation, characters' behaviours rely on story input rather than behaviour models.

6.2.4 Virtual character animation with the H-Anim standard

Currently there are two standards for virtual human animation modelling: H-Anim and MPEG 4 SNHC as discussed in Chapter 2, section 2.2.1. Based on the VRML97 specification, H-Anim represents humanoids and allows humanoids created using authoring tools from one vendor to be animated using tools from another. MPEG-4 SNHC concerns efficient compression and transmission of human geometry and animation parameters on the Web.

Since our virtual human animation focuses on off-line generation which does not concern real-time interaction and communication via the Internet, we adopt the H-Anim standard to model the virtual human characters and to define their skeleton hierarchy and articulated features. The H-Anim standard not only provides realistic geometry modelling of human size, movement capabilities, and joint limits, but more importantly it enables decoupling of humanoid geometry and the change of geometry and allows for the manipulation of either without affecting the other, therefore we can apply the pre-defined animations in the library to any humanoid models with H-Anim, i.e. any humanoid that is built to the same joint hierarchy and dimensions is able to share animations.

Most types of humanoid animation are independent of the body's actual dimensions. For example, walking, tilting the head to a specific angle or waving a hand have the same effect on any humanoid that has a skullbase *Joint*. However, some animations may be dependent on the lengths of individual segments or on the ratios of the segment lengths. For example, scratching one's head will require knowledge of the arm's dimensions. Humanoids that are sized differently, but which use the same ratios of segment lengths, may also be able to share certain animations provided that the application adjusts the animation values accordingly.

LOD is a useful concept for managing graphic complexity across many scales. In many Virtual Reality (VR) systems, a virtual object often has multi-resolution representations of polygonal mesh, either refining or simplifying according to certain criteria such as the distance of the object to the camera, so that when it is close to the camera higher LOD mesh is presented, and when it is far away from the camera a low LOD model is used. This has been proved an effective optimization strategy in computer graphics. LOD could be supported by having multiple articulations, a.k.a. Levels Of Articulation (LOA), when apply to jointed systems like virtual humans. For instance, a low LOA virtual human may be based on a 6-joint skeleton, and a higher LOA virtual human can have 18 or even 71 joints (e.g. H-Anim LOA1 and LOA2).

H-Anim provides four LOAs for applications which require different levels of detail. Some applications such as medical simulation and design evaluation require high fidelity to

anthropogeometry and human capabilities, whereas games, training and visualised living communities are more concerned with real-time performance. Language visualisation is not usually concerned with accurate simulation of humans. We use the Level 2 of Articulation (LOA2) of H-Anim in human modelling. This level ensures enough joints for human movements in language visualisation, e.g. it includes enough hand joints for grasp postures. Figure 6.3 illustrates the joints of LOA2. The dots denote joints and lines segments.

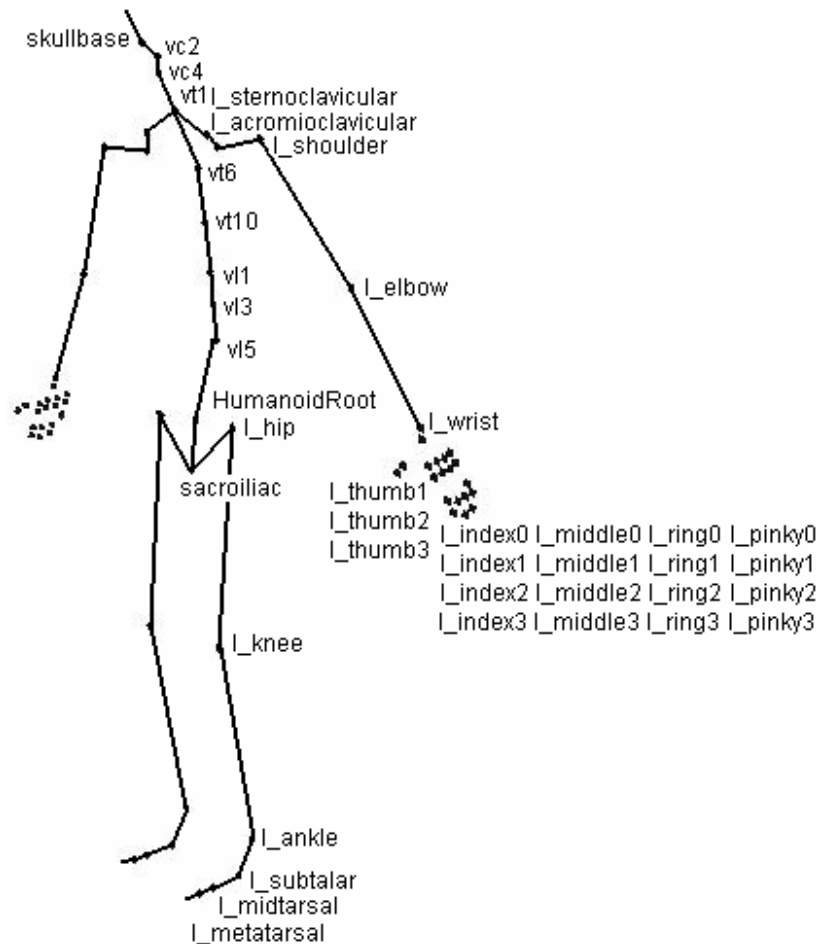


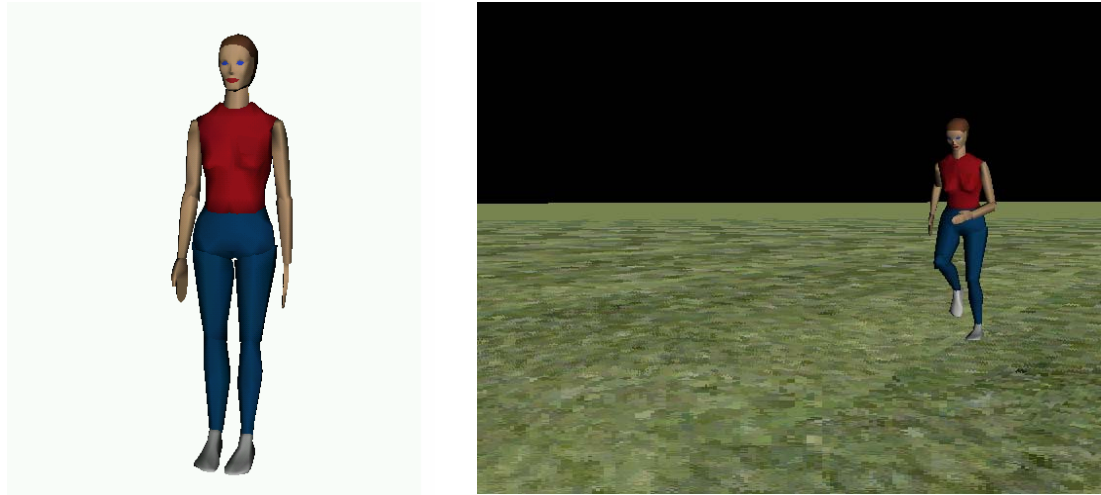
Figure 6.3: Joints and segments of LOA2

Figure 6.4 shows visualisation of the sentence “Nancy ran across the field”. Figure 6.4A is the 3D model of Nancy (Ballreich 1997) in *geometry & joint hierarchy files*, and Figure 6.4B is the output after instantiating Nancy to the `run` animation and placing this in a `field` environment. The action `run` is a basic human action predefined in an animation library.

6.2.5 Simultaneous animations and multiple animation channels

There is a lack of consideration for presenting temporal relations between multiple animation sequences and integrating different human animation sequences to present simultaneous motions. Chapter 2, section 2.2.6 has discussed an animation blending approach of building mutually exclusive animation groups. In this section, we propose an approach to presenting various temporal relations of virtual human actions, especially overlapped interval relations, using multiple animation channels, and show how this approach is employed in animation

production. The approach achieves more flexibility and control of virtual characters' animation. The temporal relations of simultaneous animations playing on multiple channels of a virtual human include the overlapped temporal relations, i.e. the relations 5-13 of Table 3.3 in Chapter 3.



A. 3D model of Nancy

B. Visualisation of “Nancy ran across the field.”

Figure 6.4: “Nancy ran across the field.”

Performing simultaneous animations is not a problem for the third level human animation modeling languages, i.e. VHML and STEP in Chapter 2, section 2.2.1, since they provide a facility to specify both sequential and parallel temporal relations. Figure 2.8 shows how VHML and STEP represent the parallel temporal relation. However, simultaneous animations cause the Dining Philosophers problem (Dijkstra 1971) for higher level animation using pre-defined animation data, i.e. multiple animations may request to access the same body parts at the same time. In order to solve this problem, we introduce the approach of multiple animation channels to control simultaneous animations.

A character that plays only one animation at a time has only a single channel, while a character with upper and lower body channels will have two animations playing at the same time. Multiple animation channels allow characters to run multiple animations at the same time, such as walking with the lower body while waving with the upper body. Multiple animation channels often need to disable one channel when a specific animation is playing on another channel to avoid conflicts with another animation.

We use an animation registration table, part of which is shown in Table 6.1, to facilitate multiple animation channels. Every pre-defined animation must register in the animation table and specify which joints are used for the animation. In Table 6.1, each row represents one animation, and each column represents one joint. 0 indicates that the joint is not used for the animation; 1 indicates that it is used and can be disabled when playing simultaneous animations; and 2 means that the joint is used and cannot be disabled. When simultaneous animations are requested, an animation engine checks the animation table and finds if the involved joints of

these animations conflict, i.e. if there is any joint whose values for both animations are 2, these animations conflict and they cannot be played at the same time. If two animations do not conflict (for example, “run” and “throw”), the animation engine merges their keyframes information, i.e. interpolaters, and creates a new animation file which will be applied to the virtual character.

<i>Involved joints /Animations</i>	<i>sacroiliac</i>	<i>l_hip</i>	<i>r_hip</i>	<i>...</i>	<i>r_shoulder</i>
walk	2	2	2	...	1
jump	2	2	2	...	1
wave	0	0	0	...	2
run	2	2	2	...	1
scratch head	0	0	0	...	2
sit	2	2	2	...	1
...

Table 6.1: The animation registration table

Figure 6.5 shows an example of integrating the two animations “walk” and “wave”. Figure 6.5A is a snapshot of walking animation, B is waving animation, and C is integrated animation of walking and waving, using the multiple animation channels approach. The motion of waving only uses three rotation interpolaters: *r_shoulder*, *r_elbow*, *r_wrist*. The animation engine looks up the animation table and finds that the walking animation also uses these three joints and their values are all 1, which means the right arm movements of walking can be disabled and overwritten by the movements of waving. The animation engine then replaces the keyframes of these three joints in the walking animation file with those in the waving file and generates an integrated motion.

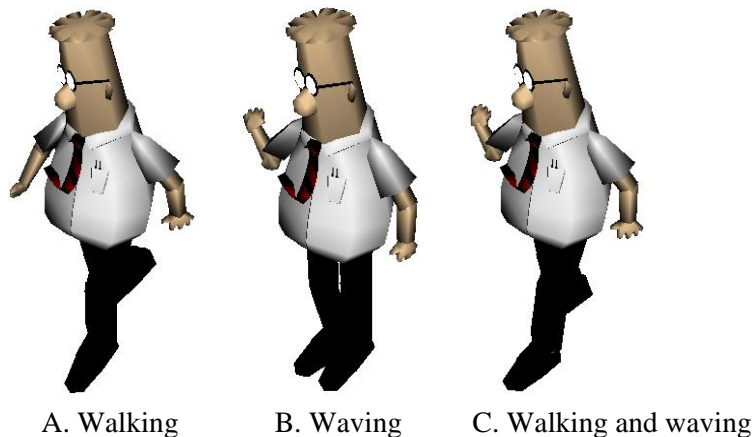


Figure 6.5: An example of motion integration

This approach combines pre-created and dynamically generated (procedural) animation facilities into a unified mechanism, and employs multiple animation channels to integrate non-conflict simultaneous animations. It allows language visualisation to take advantage of procedural animation effects in the same manner as regular animations. Compared with Improv’s grouping method, this approach provides a finer integration of simultaneous animations, and hence achieves more flexibility and control on virtual character animation.

Animation loops are used to present action repetition which was introduced in Chapter 4, section 4.5.2. The semantic representation indicates whether an animation should loop. If not specified, the animation will play once and stop, i.e. looping is disabled as default, which is controlled by a time sensor in a VRML file.

6.2.6 Facial expression and lip synchronisation

Lip movement concerns another modality (speech) by creating the illusion of corresponding speech. Traditional animators use a system called track reading in which the animation is carefully analysed for mouth positions laid out against a time sheet. The animator's true skill is knowing how to condense speech into as few positions of lips as needed to create the speaking illusion. Using lip movement to support speech output helps to alleviate communication problems by redundant coding. Previous user interface agents focus on the visualisation of a fully animated talking head (CSLU 2002, Alexa et al. 2000). Facial expression can be implemented by using `CoordinateInterpolator` and `NormalInterpolator` in VRML to animate morphing and shading on a character's face.

We distinguish three visemes (Table 6.2) and adopt the six expressions of MPEG4 definition (Chapter 2, Table 2.2) using parameters of eyes, eyelids, brows, lips, and jaw. We ignore all consonant visemes because they are not distinct enough for a rough simulation and have high computational costs. The five vowel visemes defined in MPEG4 (Chapter 2, Table 2.1, 10-14) are merged to three visemes according to the two articulation features which can be shown via jaw and lip movement: (1) high/low tongue position (jaw close/open), and (2) lip roundness. The three visemes are shown as the three bounded areas in the cardinal vowel quadrilateral in Figure 6.6. Viseme *a* is an open-jaw articulation; viseme *i* is a close-jaw, extended-lip articulation; and viseme *o* is a rounded-lip articulation. Figure 6.7 shows lip synchronisation of these three visemes.

<i>Simplified visemes</i>	<i>MPEG4 visemes</i>	<i>Examples</i>
Viseme a	viseme_a viseme_e	car bed
Viseme i	viseme_i	Tip, tea
Viseme o	viseme_q viseme_u	top book

Table 6.2: Three simplified visemes

It is computationally economical to introduce Level-Of-Detail (LOD) for facial expression and lip synchronisation to accelerate the generation and rendering of virtual humans. Simplifications of the virtual human, i.e. omitting facial animation, are produced and contain fewer details. These simplifications can then be used when the virtual human is further away and the facial details are not noticed anyway.

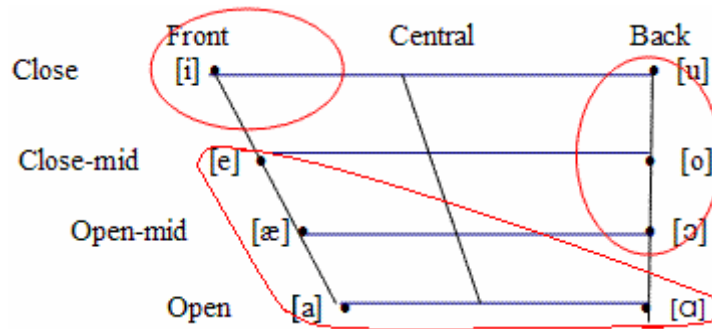


Figure 6.6: Cardinal vowels quadrilateral

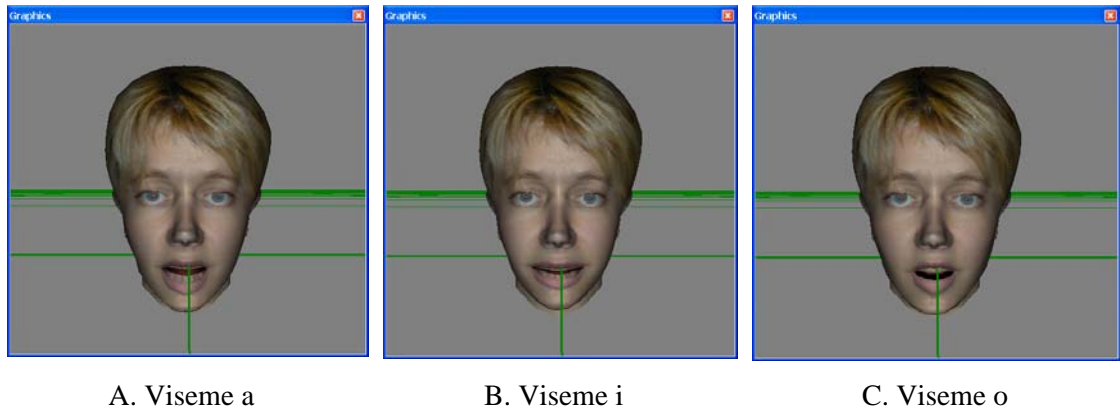


Figure 6.7: Lip synchronisation of viseme a, i, o

6.2.7 Space sites of virtual humans

In the geometric VRML files of 3D objects and H-Anim files of virtual humans, there are a list of grasp sites and their purposes, and intrinsic directions such as top and front, defined with respect to an object, and a list of sites for manipulating and placing/attaching objects defined with respect to a virtual human. We classify three types of objects:

1. *Small props* which are usually manipulated by hands or feet, e.g. cup, box, hat, ball.
2. *Big props* which are usually sources or targets (goals) of actions, e.g. table, chair, tree.
3. *Stage props* which have internal structure, e.g. house, restaurant, chapel.

To figure out where to place these three types of props around virtual human bodies, we create corresponding site tags for virtual humans using H-Anim Site nodes.

(1) Manipulating small props:

For manipulation of small props, a virtual human has six sites on the hands (three sites for each hand, `l_metacarpal_pha2`, `l_metacarpal_pha5`, `l_index_distal_tip`, `r_metacarpal_pha2`, `r_metacarpal_pha5`, `r_index_distal_tip`), one site on the head (`hanim_skull_tip`), and one site for each foot tip (`l_forefoot_tip`, `r_forefoot_tip`). The sites `metacarpal_pha2` are used for grip and pincer grip; `metacarpal_pha5` are for pushing; and `index_distal_tip` are for pointing. The sites `forefoot_tip` are for kicking. Figure 6.8 shows the position of these sites.

(2) Placing big props:

For big props placement, we use five sites indicating five directions around the human body: `x_front`, `x_back`, `x_left`, `x_right`, `x_bottom`. We leave out `x_top` because there is already a site node, `hanim_skull_tip`, defined on the head of every virtual human for attaching headdress. Big props like a table or chairs are usually placed at these positions.

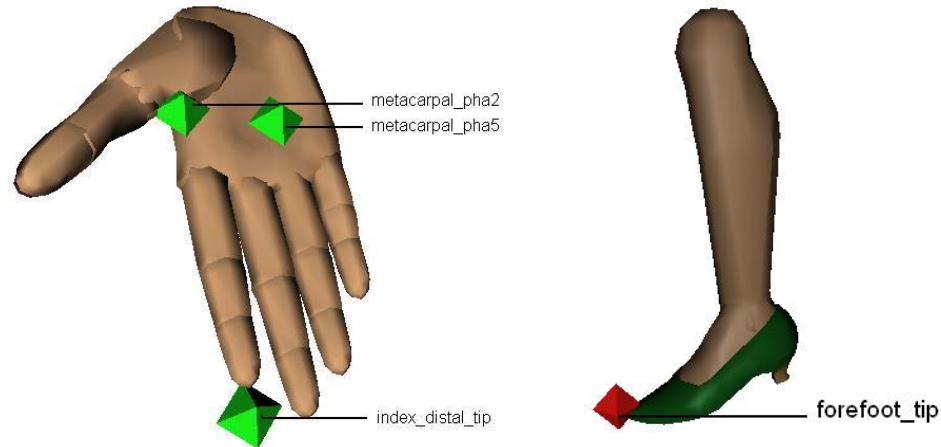


Figure 6.8: Site nodes on the hands and feet of a virtual human

(3) Setting stage props:

For stage props setting, we have five more space tags besides those in (2) around a virtual human to indicate further places: `x_far_front`, `x_far_back`, `x_far_left`, `x_far_right`, `x_far_top`. Figure 6.9 shows the positions of these sites. Stage props such as a house often locate at these far sites of virtual humans.

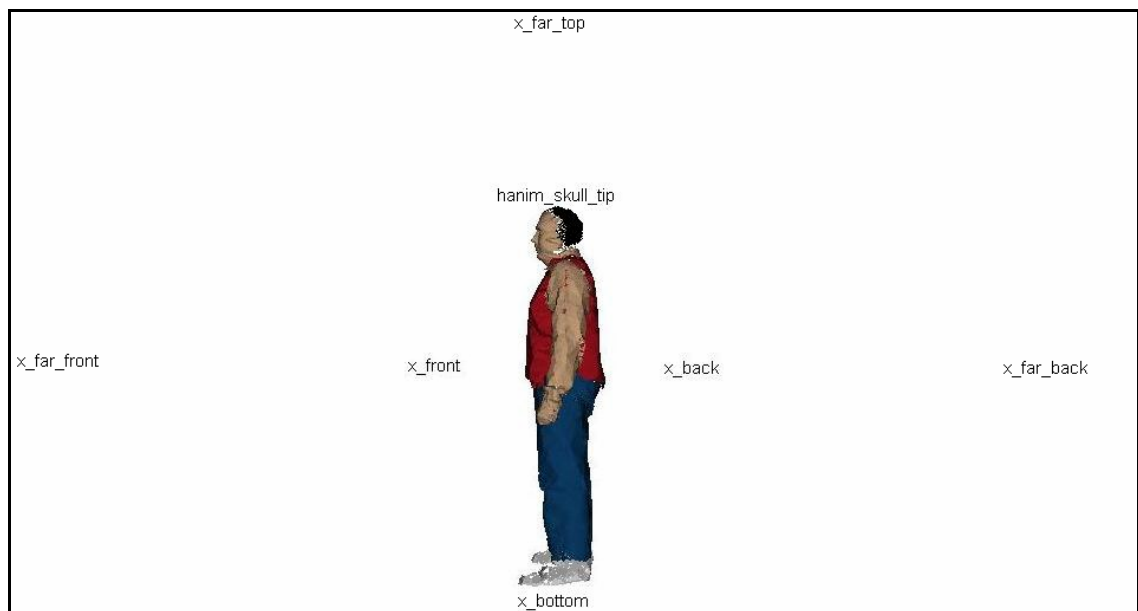


Figure 6.9: Site nodes around a virtual human's body

6.2.8 Multiple character synchronisation and coordination

It is very likely that virtual scenes concern not just one major character (the protagonist) but other minor characters who may communicate, react, or coordinate with the protagonist. Modelling multiple characters raises the requirements of synchronisation and coordination,

especially in a multimodal situation. A character can start a task when another signals that the situation (pre-conditions) is ready, characters can communicate with one another, or two or more characters can cooperate in a shared task. These multiple characters' activities concern many issues such as action synchronisation and collision detection. An event-driven timing mechanism is used in our language-to-animation conversion for action synchronisation because VRML provides a utility for event routing (ROUTE node). For instance, in the modelling of the following event between two actors Nancy and John:

Nancy was walking along the street. John called her. Nancy stopped and saw John. John walked towards her. They exchanged greetings.

the end of the animation `john_speech` (calling Nancy) triggers three events: (1) to stop the animation of `nancy_walk`; (2) to start the animation of `nancy_gazeWander` (searching for who's calling) and (3) to start the animation of `john_walk` (walking towards Nancy). Moreover, in a dialogue between two characters the facial expressions or gestures of one character may trigger the other's response such as the addresser's raising eyebrow triggers the addressee's nodding. It is obvious that event-driven synchronisation is more natural and also convenient than an exact time-driven mechanism.

6.3 Object modelling

Object-oriented 3D models suitable for animating human-object interaction in simulated virtual environments are proposed in this section. The models encapsulate not only objects' geometry but also their behaviour including auditory icons and human-object interaction such as grasping site and hand posture. This approach decentralizes the animation control since object interaction information is stored in the objects, and hence most object-specific computation is released from the main animation control.

6.3.1 Default attributes in object visualisation

Graphic representation of natural language gives rise to the problem of the gaps between meanings represented by images and those by natural language, as well as problems of ambiguity and underspecification of natural language as discussed in Chapter 5. Default attributes which are in humans' common sense are used to bridge the gap between image and language meanings. Requiring the computer to construct and display a scene corresponding to its interpretation of an input text forces us to be explicit about much of the common sense that pertains to an object, such as size, orientation, location and colour. The choice of defaults is a useful method to help solve this problem and hence enables animation to approach reality. Unless indicated particularly in the story (c.f. the example in Figure 6.10), the attributes of an object are decided by default values.

Without the particular measurement in the story, a door about 7 feet high and a 3 feet tall little girl will be drawn. So after interpretation the first paragraph in the example, a 3 feet tall (default value of a child) girl and a 15-inch-high (value indicated in the story) door are

generated. The next paragraph specifies Alice’s height as ten inches. The system then shows her height relative to the door, i.e. two thirds of the door height.

```

On the second time round, she came upon a low curtain she had not
noticed before, and behind it was a little door about fifteen
inches high.
...
And so it was indeed: she was now only ten inches high, and her
face brightened up at the thought that she was now the right size
for going through the little door into that lovely garden.

```

Figure 6.10: Alice in Wonderland: Down the Rabbit-hole

Default values of an object’s attributes are indispensable not only in object visualisation but in setting values to modifiers that modify the object as discussed in Chapter 4, section 4.7. Moreover, default values may also concern events such as speed and mode of movement. When translating “a running van” the van moves at a usual speed, its default speed, and when translating “a fast running van” it moves at a higher speed than the default one.

6.3.2 Space sites of 3D objects and grasping hand postures

The geometric file of a 3D object defines its shape, default size, functions, as well as any constraints that might be associated with the manipulation of that particular object, such as allowable actions that can be performed on it, and what the expected outcome of the actions will be, e.g. the outcome state of a lamp when it is switched on/off.

Similar to virtual humans, objects in our graphic library usually have six space sites indicating six directions around the object if applicable: `x_front`, `x_back`, `x_left`, `x_right`, `x_on`, `x_under`, one functional space site `x_in`, and several grasp site-purpose pairs. Stage props, such as “house”, “restaurant”, “chapel”, normally don’t have grasp site-purpose pairs. Figure 6.11 illustrates sites of a desk which has two grasp site-purpose pairs, one on the drawer knob for opening, and the other at the side of the desk for pushing. Space sites often relate to the object’s function, for instance, the front and back sites of a desk or a chair depend on their functionalities. Objects’ space sites are not only useful for objects/virtual human positioning, but also for expected virtual human behaviours in order to accomplish the interaction with them. For example, before opening a drawer of the desk in Figure 6.11, the actor is expected to be in a suitable position (i.e. `x_back`) so that the drawer will be in the reach and not collide with the virtual human when opening.

Figure 6.12 gives a list of verbs describing hand movements. Some of them (e.g. verbs of empty-handed gestures and haptic exploration) can be defined in the animation library, while others cannot be defined solely on the verbs themselves because their hand shapes are closely associated with the shape, size, or functionality of the object they manipulate. Cadoz (1994) defined the latter group as *ergotic* hand movements, which are associated with the notion of

work and the capacity of humans to manipulate the physical world and create artefacts. For example, the hand shape and movement of “wave” is defined in an animation key frame file wave.wrl, but the hand shape of “pick up” is uncertain if we don’t know what the virtual human picks up because the hand shapes and grasp points of picking up a cup and a bottle are quite different.

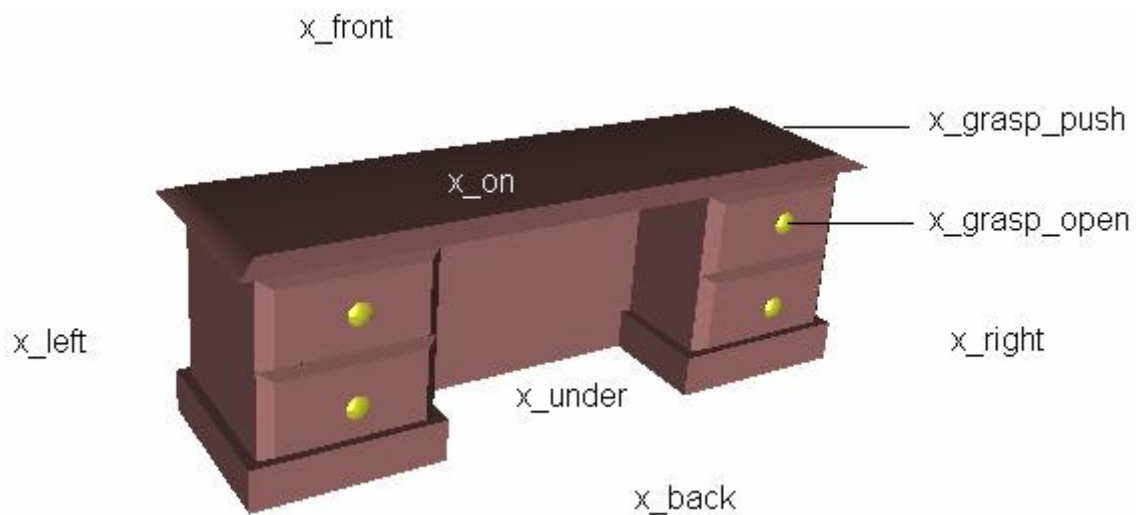


Figure 6.11: Space sites of a 3D desk

Ergotic hand movements

- Contact with the object: grasp, seize, grab, catch, embrace, grip, lay hold of, hold, snatch, clutch, take, hug, cuddle, cling, support, uphold
- Contact and changing position: lift, move, heave, raise, translate, push, pull, draw, tug, haul, jerk, toss, throw, cast, fling, hurl, pitch, depress, jam, thrust, shake, shove, shift, shuffle, jumble, crank, drag, drop, pick up, slip, hand over, give
- Contact and changing orientation: turn, spin, rotate, revolve, twist
- Contact and changing shape: mold, squeeze, pinch, wrench, wring, stretch, extend, twitch, smash, thrash, break, crack, bend, bow, curve, deflect, tweak, spread, stab, crumble, rumple, crumple up, smooth, fold, wrinkle, wave, fracture, rupture
- Joining objects: tie, pinion, nail, sew, button up, shackle, buckle, hook, rivet, fasten, chain up, bind, attach, stick, fit, tighten, pin, wrap, envelop, swathe
- Indirect manipulation (via other objects): cut, whet, set, strop, whip

Empty-handed gestures

- wave, snap, point, urge, show, size, count

Haptic exploration

- beat, bump, brush, caress, clink, drub, flick, fondle, hit, jog, kick, knock, nudge, pluck, prick, poke, pat, rap, rub, slam, slap, strike, stroke, struck, strum, tap, touch, thrum, twang, twiddle, throb, thwack, tickle, wallop, whop

Figure 6.12: Verbs of hand movements

Ergotic hand movements can be classified according to the physical characteristics or their function (Table 6.3). Ergotic verbs in Figure 6.12 are grouped by physical characteristic, i.e. change effectuated and indirection level. It is more common to classify ergotic hand movements according to their function, either prehensile or non-prehensile. Non-prehensile movements include pushing, lifting, tapping and punching.

<i>Classification standards</i>	<i>Physical characteristics</i>	<i>Functions</i>
Classes	<ul style="list-style-type: none"> • Change effectuated: position, orientation, shape • How many hands are involved: one or two • Indirection level: direct manipulation or through another object or tool 	<ul style="list-style-type: none"> • Prehensile • Non-prehensile

Table 6.3: Taxonomy of ergotic hand movements

To illustrate how complex it can be to perform a simple task of ergotic hand movement, let's consider the example of picking up a mug: walking to approach the mug, deciding which hand to use, searching for the graspable site (i.e. the handle), moving body limbs to reach the handle, deciding which hand posture to use, adjusting hand orientation and the approaching aperture, grasping, close the grip, and finally lifting the mug.

There are two approaches to organizing the knowledge required in the above task to achieve "intelligence" for successful grasping. One is to store applicable objects in the animation file of an action and using lexical knowledge of nouns to infer hypernymy relations between objects. For instance, one animation file of "pick-up" specifies the applicable objects are cups. The hand posture and movement of picking up a cup are stored in the animation file. From the lexical knowledge of the noun "mug" the system knows that a "mug" is a kind of "cup" and its meronymy² relations, and the system then accesses the mug's geometric file to find its grasp site, i.e. the location of the handle. The system then combines the "pick up" animation for a cup object with the virtual human and uses it on the mug.

The other approach includes the manipulation hand postures and movements within the object description, besides its intrinsic object properties. Kallmann and Thalmann (2002) call these objects "smart objects" because they have the ability to describe in detail their functionality and their possible interactions with virtual humans, and are able to give all the expected low-level manipulation actions. This approach decentralises the animation control since object interaction information is stored in the objects, and hence most object-specific computation is released from the main animation control. The idea comes from the object-

² The "parts of" relationship. The meronyms of "mug", for example, are handle, stem, brim, and base.

oriented programming paradigm, in the sense that each object encapsulates data and provides methods for data access.

Robotics techniques can be employed for virtual hand simulation of ergotic hand movements, as for automatic grasping of geometrical primitives. They suggest three parameters to describe hand movements for grasping: hand position, orientation and grip aperture. Su (1994) suggests touching, pointing and gripping as a minimal set of gestures that need to be distinguished. Sign language synthesis using avatars who can translate spoken languages or text to sign languages for deaf people (Elliott et al. 2000, Fabian and Francik 2001) also contributes to animation of hand movements.

We use four stored hand postures and movement (Figure 6.13) for moving, touching and interacting with 3D objects: index pointing (Figure 6.13A, e.g. press a button), grip (Figure 6.13B, e.g. hold cup handle, knob, or a cylinder type object), pincer grip (Figure 6.13C, i.e. use thumb and index finger to pick up small objects), and palm push (Figure 6.13D, e.g. push big things like a piece of furniture). They use different hand sites to attach objects (see section 5.2.7 for these sites). Hand postures and movements are defined as the motions of fingers and hands in virtual humans' VRML files. Different kinematic properties, such as movement velocity and grip aperture are fixed since further precision might involve significant costs in terms of processing time and system complexity but the result is only a little more realistic.

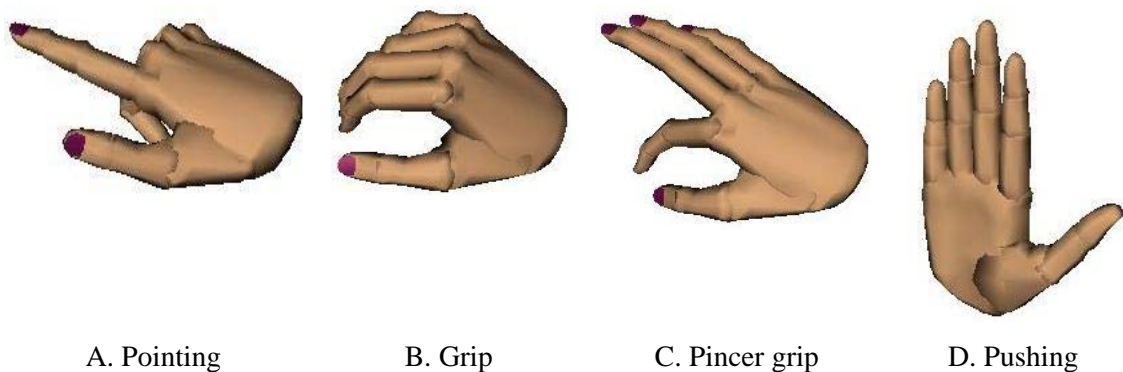


Figure 6.13: Four hand postures for physical manipulation of objects

6.4 Collision detection

When we design the layout of the virtual story world, i.e. the placement of props, and the movement of actors, besides considering their size and position one inevitable problem we may encounter is *collision detection* — a ‘naïve’ physical problem about how to examine collision/intersection of objects and actors. Although VRML provides a built-in collision detection mechanism for the avatar, the mechanism does not apply to other objects. However, a large proportion of collisions occur among objects and actors’ motions, especially where actors are not the first person (non avatar).

To reduce computation time collision avoidance algorithms often use coarse approximations, such as bounding volumes, for complex geometries like virtual humans. Two

objects are detected as colliding when their bounding volumes intersect. Typical types of bounding volumes are bounding spheres and bounding boxes (axis-aligned or oriented) as shown in Figure 6.14. There are two possible implementations in VRML. The first is to write up scripts detecting intersection between the bounding boxes/spheres of objects/characters. This requires 3D translation calculation especially when the objects detected are moving (rotating). ParallelGraphics' Cortona VRML client provides a VRML extension for object-to-object collision detection interface (Parallelgraphics 2003). The interface is built around two native ECMAScript objects, *Collidee* and *Collision*. The former acts as a proxy for the shape that is transformed, bearing the parameters of the transformation matrix and other relevant data, and the latter describes the point of the shape in question that came into contact with another shape, in the case of a collision.

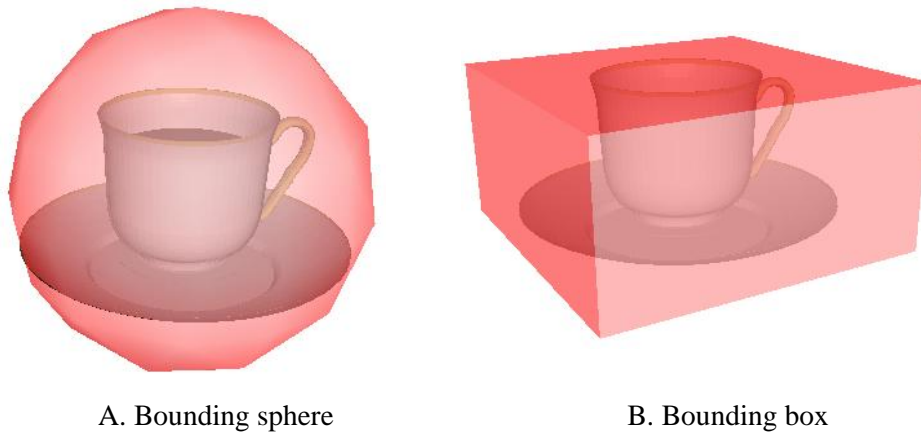


Figure 6.14: Bounding volumes

Another approach is to bind a viewpoint with object geometry temporarily in order to detect collisions, therefore we may utilize automatic collision detection for the avatar (the active viewpoint) by using *Collision* node and setting *avatarSize* in *NavigationInfo* Node meanwhile. But this approach restricts the observation at the collision time, i.e. the user has to observe the event from the view of one *collidee*.

When detecting collisions concerns the protagonists, those characters which are observed more closely and get more attention, more accurate collision handling is needed. Real people may bump up against each other in an enclosed area, hold hands, or grasp objects. At the extreme, there are computationally expensive approaches for highly detailed collision detection with virtual humans, performing polygon level checks between humans and objects in the scene, e.g. for cloth simulation and contact modelling. In order to adapt to different requirements in both path planning and contact modelling, and give a computationally economical solution, we use bounding cylinders around the human body segments for protagonists and a bounding cylinder around the whole human body for minor characters and characters beyond the scope of attention, and perform an approximate collision detection with them.

Collision detection is also a crucial issue in characters' manoeuvring of objects and multiple characters' activities. Grasping an object is an ordinary movement of a virtual human.

The character should be able to reach for the object, check the object-specific information about its graspable site, position his/her hand on the proper approach direction, detect a collision/connection of his hand and the object, close his hand, and finally attach the object to his hand's *Site*³. Collision detection is a critical point for visualising contact verbs, such as “hit”, “collide”, “rub”, “stroke”, and “scratch”. In the visualisation of “stroke the dog”, the animation engine needs to know when the hand contacts the dog and then changes the direction of the hand's movement and makes sure it contacts the dog all the time in the stroke.

6.5 Automatic camera placement

Since there is no available director or editor in real-time, automated camera placement and control are important in virtual environments. A *camera controller* applies cinematic principles to place camera and control the camera behaviour. Some of the cinematic principles, which describe the relation between the action and the camera behaviour, are action-dependent. For instance, the over-the-shoulder shot shown in Figure 6.15 emerges from conversation rules, presenting verbs of communication. Other principles are action-independent, i.e., cinematic constraints on combinations of shots.



Figure 6.15: Over-the-shoulder shot for presenting verbs of communication

In automatically generated virtual worlds, the camera must provide a clear view of what the character is doing and the related scene context. Since each 3D object and virtual human in the graphic library has several predefined viewpoints, when combined together in a 3D scene, there are many candidates available for a default camera position. The camera controller needs to select the best perspective from them to view as much animation as possible according to the cinematic principles. Figure 6.16 shows several viewpoints of the animation “Bob left the gym”.

³ A *Site* node in H-Anim defines an attachment point for accessories such as jewelry, clothing, and other objects.

A and B are viewpoints defined in the house’s geometric file, and C-F are some of the viewpoints defined in the virtual human Bob’s H-Anim file. Figure 6.16F (Bob far inclined view) is chosen by the camera controller as the default view of the animation because it is one of the best viewpoints to observe the 3D scene.

Similar to space tags, object-related viewpoints are predefined in the object’s VRML file. Table 6.4 lists viewpoints defined for small props, big props, stage props as defined in section 6.2.7, and virtual humans. Functional information of props is also considered, e.g. the bottom view of lights and ceiling fans is defined in their geometric files as well.

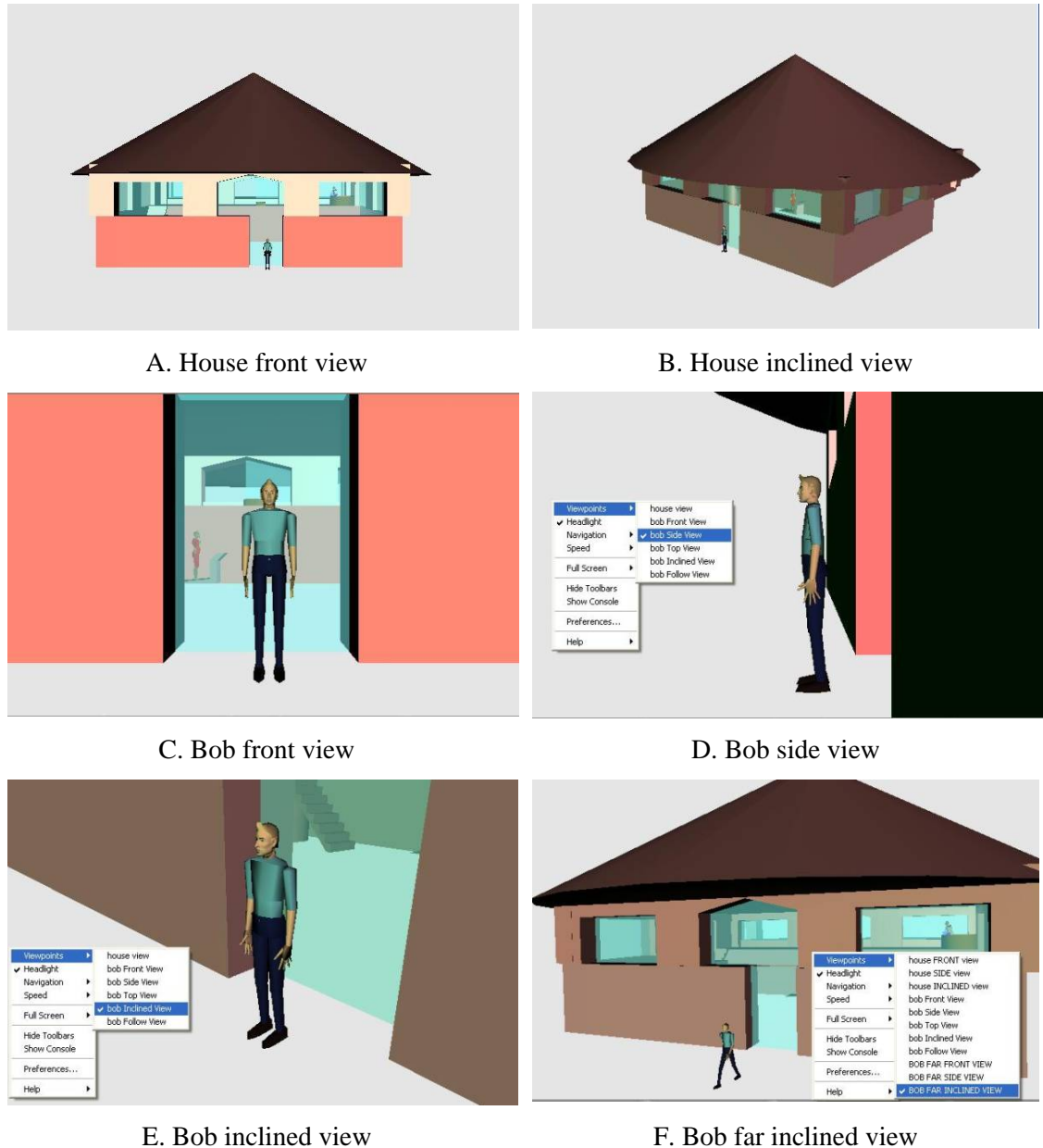


Figure 6.16: Viewpoints of the animation “Bob left the gym”

Since our output virtual world allows users to move (or plan, pan, turn, roll) the viewpoint with the mouse, an animated camera is not frequently used except in the situation of in-vehicle view where the viewpoint is attached to a moving vehicle or character. The cinematic

principles tell the animation engine where to place the default camera to make sure anything (anybody) important is in shot. Examples of the cinematic principles are listed in the following:

- For a static target, use inclined view to allow users to observe most faces of the target.
- For an animated target, use far inclined view to ensure most of the movement is in shot.
- For vehicles, use inside view to allow the user to follow the vehicle's movement.
- For presenting verbs of speaking, use front view if only one character is involved; and use the listener's follow view (over-the-shoulder shot) if two characters are involved.
- For character animations involving two characters, use inclined view of the agent (if the input is not passive) if no spatial movement involved; and use far inclined view if spatial movement is involved.

<i>VRML files</i>	<i>Predefined viewpoints</i>	<i>Examples</i>
small props	front view, side view, top view, inclined view	cup, box, hat, pen
big props	front view, far front view, side view, far side view, top view, far top view, inclined view, far inclined view	table, chair
stage props	front view, far front view, side view, far side view, inclined view, far inclined view, inside view	house, restaurant, beach
Virtual humans	front view, far front view, side view, far side view, top view, far top view, inclined view, far inclined view, follow view, far follow view	Bob, Nancy

Table 6.4: Predefined viewpoints of 3D props and characters

6.6 Summary

This chapter discussed various issues of computer animation such as virtual human and 3D object animation, animation of facial expressions and lip synchronisation, autonomy, collision detection, and automatic camera placement. An approach of multiple animation channels to blend non-exclusive animations and present overlapping interval relations was proposed. In 3D object modelling, we use object-oriented models to encapsulate object-related information, such as geometry, behaviour, sound effects, and human-object interaction, to decentralise the control of the animation engine. In virtual human animation, we used general-purpose virtual human characters and their behaviours which follow the industry standard (H-Anim) and balance between computational efficiency and accuracy to produce believable human motions.

Chapter 7

CONFUCIUS:

An Intelligent MultiMedia Storytelling System

We have developed an intelligent multimedia storytelling system, CONFUCIUS, to test the theories discussed in previous chapters. This chapter describes the implementation details of CONFUCIUS, which converts natural language to 3D animation and audios, with regard to natural language understanding, 3D animation generation, virtual humans and Text-To-Speech (TTS) synthesis. CONFUCIUS is implemented using VRML, Java and Javascript, and it integrates existing tools for language parsing and text-to-speech synthesis.

7.1 Architecture of CONFUCIUS

The architecture of CONFUCIUS is given in Figure 7.1. The dashed part in the figure is the knowledge base, including language knowledge (lexicons and a syntax parser) which is used in NLP, and visual knowledge such as 3D models of characters, props, and animations of actions, which is used for generating 3D animations. The *surface transformer* is a pre-processing module which takes natural language sentences as input and manipulates surface text, e.g. transforming indirect quotations to direct quotations. The *NLP* module uses language knowledge to parse sentences, to analyse their semantics, and outputs Lexical Visual Semantic Representation (LVSr). The *media allocator* then assigns content to three modules: animation engine, TTS engine, and narration with *Merlin the narrator*. For example, it assigns the parts bracketed in quotation marks near a communication verb to the speech modality. The *animation engine* takes semantic representation in the LVSr format discussed in Chapter 4 and uses visual knowledge to generate 3D animations including sound effects. The TTS engine synthesises the characters' speech and generates wav files. The outputs of the animation engine and TTS engine are synchronised in a VRML file which presents a 3D virtual world including animation and speech. The *narration* module generates Javascript embedded in HTML files which controls the presentation agent, Merlin the Narrator, to complete a multimodal story presentation. Finally, the *narration integration* module integrates the VRML file with the HTML file into a two-frame HTML file in which one frame contains the virtual world and the other frame controls Merlin's presentation behaviours.

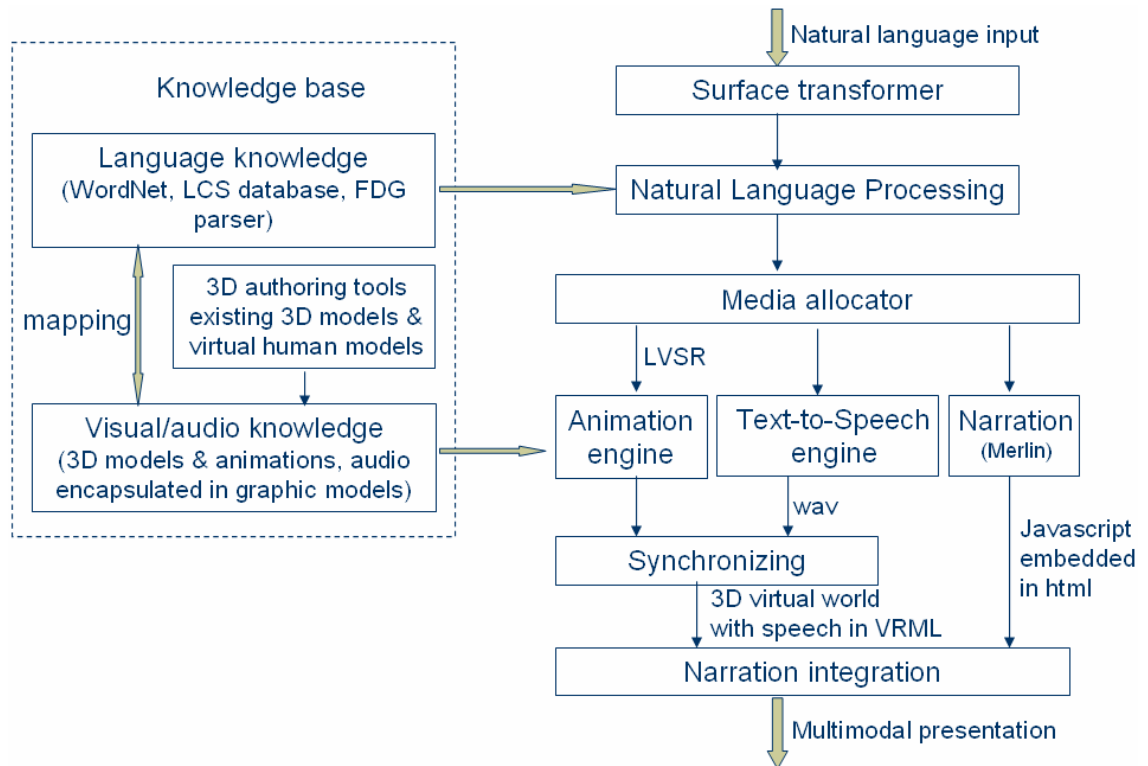


Figure 7.1: Architecture of CONFUCIUS

7.2 Input and output

As illustrated in Figure 7.2, the input of CONFUCIUS is natural language text. At present, CONFUCIUS only deals with single sentences with non-metaphor word senses and output animations of rigid¹ objects and human bodies. Natural language expresses concepts at different levels of abstraction. Currently, CONFUCIUS handles concepts, actions and states at low abstraction levels. Most of the sentences used in implementation and evaluation are chosen from children’s stories like “Alice in Wonderland”, because entities in these stories are usually not abstract but tangible and amenable to visualisation. Hence presentation difficulties caused by abstract expressions may be circumvented, e.g. it is possible to represent the meaning of “slow” but difficult to present abstract adjectives like “eccentric”. On a longer-time scale, it is planned to apply CONFUCIUS to other story domains of increasing complexity and abstraction that require metaphor understanding.

CONFUCIUS’ multimodal output includes 3D animation with speech and nonspeech auditory icons, and a presentation agent, Merlin the narrator. Our work on synthesising multimodal output focuses on generating virtual character animation and speech with particular emphasis on how to generate virtual humans’ movements for verb classes discussed in Chapter 5, section 5.2.4.

¹ We do not simulate deformable objects and bodies.

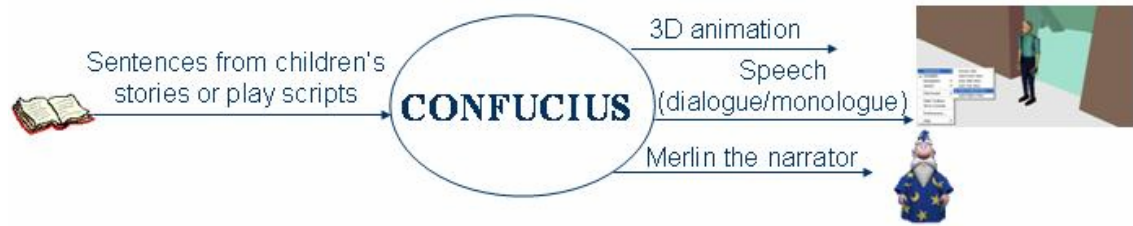


Figure 7.2: Input/output of CONFUCIUS

CONFUCIUS' audio presentation includes auditory icons, which are nonspeech sound effects such as real world sound accompanying animated events in the scene, and text-to-speech. Auditory icons are encapsulated in the 3D models of objects and virtual humans, e.g. the firing sound of a gun is encapsulated in the gun's geometric file, and the "hiccup" and "yawn" sounds of a virtual character are encapsulated in his/her VRML file. Since auditory information can be redundant with visual and language modalities, determining whether to eliminate the visual (or speech) information or make the audio information redundant is a task of the media allocation module.

CONFUCIUS is developed using existing software tools such as the Connexor Machine (2003) and WordNet 2.0 (Fellbaum 1998) for natural language processing, 3D Studio Max 5 (3ds Max 2002) for object modelling, Microsoft Agent (Microsoft Agent 2002) for authoring the animation of the presentation agent Merlin, and FreeTTS (FreeTTS 2004) for speech synthesis. Virtual Reality Modelling Language (VRML 2002) is used to model 3D graphics and animations, and VRMLPad 2.0 (Parallelgraphics 2003) is used for editing VRML files. Java programs generate/assemble VRML code and integrate all these components into CONFUCIUS. CONFUCIUS has approximately 2600 lines of source code. It runs on a Mobile Intel(R) Celeron 2.20 GHz CPU with 224 MB of RAM and NVIDIA GeForce 6600 GT graphics card, under Microsoft Windows XP.

Currently, CONFUCIUS is able to visualise single sentences which contain action verbs with visual valency of up to three, e.g. "John left the gym", "Nancy gave John a loaf of bread". Verbs that CONFUCIUS cannot handle include verbs involving 3D morphing² (e.g. "melt" in class 2.1.1, Chapter 5, Figure 5.3), deformable and breakable objects (e.g. "bend" in class 1.2, "break" in class 2.1.2), verbs without distinct visualisation when out of context (e.g. "play" in class 2.2.1.4) and high level behaviours or routine events (e.g. "interview" in class 2.2.1.5).

7.3 Knowledge base

To generate understandable story animation, human common sense including social conventions, other aspects of the culture and world in which the story occurs, and default attributes of objects must be incorporated into a knowledge base. Without proper knowledge of

² The construction of a sequence depicting a gradual transition between two 3D models.

a specific domain that a story pertains to, a system could not tell the story intelligently. Schank's SAM (c.f. Chapter 3, section 3.1.2) suffered from this problem. It had difficulties to infer reasonably from the story and answer questions about it when the relevant *script* does not exist in SAM's knowledge base.

The knowledge representation required for CONFUCIUS must provide the following capabilities: (i) model both declarative and procedural knowledge, (ii) inference mechanisms such as classification based inference. Figure 7.3 illustrates the design of the knowledge base of CONFUCIUS, which is also a general design of knowledge base for intelligent multimedia applications which integrate natural language and visual processing. It consists of language knowledge, visual knowledge, and cinematic principles.

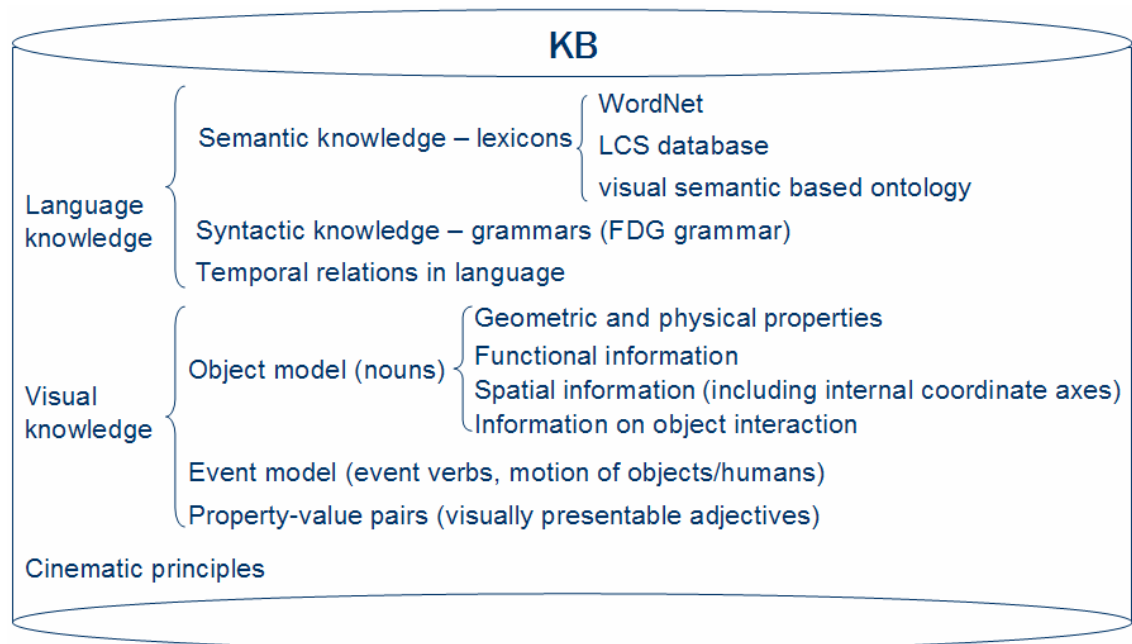


Figure 7.3: Knowledge base of CONFUCIUS

The language knowledge is used in CONFUCIUS' NLP component to extract concept semantics from text. It includes semantic knowledge (i.e. lexicons), syntactic knowledge (i.e. Functional Dependency Grammar used in the Connexor parser), and interval based temporal relations discussed in Chapter 4. We use off-the-shelf lexicons WordNet, which has 155,327 words all together (117,097 nouns, 11,488 verbs, 22,141 adjectives, and 4,601 adverbs), LCS database, which has 11,000 verb entries totally (4,432 verbs in 492 classes), and the visual semantics based language ontology we proposed in Chapter 5. The lexicons also include syntactic knowledge such as subcategorisation, and statistical information such as WordNet's word frequency, both of which are important for Word Sense Disambiguation (WSD).

The visual knowledge consists of the information required to generate animations. It consists of *object model*, *event model* and *property-value pairs*. The *object model* comprises visual representation of the ontological category (or conceptual "parts of speech") — things (nouns), which consists of simple geometry files for props and places, and H-Anim files for

human character models, which are defined in geometry & joint hierarchy files following the H-Anim specification (H-Anim 2001). Object model includes *geometric/physical properties* of an object (e.g. the object's position, shape, size, and colour), *functional information* (i.e. the object's function), *spatial information* and *information on object interaction*. Spatial information is for spatial reasoning and refers to proximity and gaze directions of virtual humans. It includes *internal coordinate axes*, which are indispensable in some primitive actions of event models, such as rotating operations, requiring spatial reasoning based on the object's internal axes. Information on object interaction is for human-object interaction. The current version of CONFUCIUS has 40 object models that are able to visualise not only the particular nouns they present but also their synonyms, hyponyms and hypernyms, for instance, a house model can present “house”, “bungalow”, and “cottage”.

The *event model* comprises visual representation of events (verbs) that contain explicit knowledge about the decomposition of high level acts into basic motions, and defines a set of basic animations of objects or virtual humans such as “walk”, “jump”, “give”, “push”. The current version of CONFUCIUS has 25 basic event models (verbs) which are able to visualise not only these basic actions but also their synonyms, hypernyms, troponyms, coordinate terms, and a group of verbs in corresponding verb classes (Levin 1993). Additionally, the visual knowledge is capable to be expanded by appending more event models and object models. The *event model* requires access to other parts of visual knowledge. For instance, in the event “he cut the cake”, the verb “cut” concerns kinematical knowledge of the subject — human being, i.e. the movement of his hand, wrist, and forearm. Hence it needs access to the *object model* of the man who performs the action “cut”. It also requires the lexical knowledge (e.g. instrument) of “cut” and *function information* of the instrument (e.g. “knife”), the *internal coordinate axes* information of “knife” and “cake” to decide the direction of the movement. To interpret the verb $wear(x, y)$ ³, the *event model* needs access to the *object model* of y, which might be a hat, a ring, a pair of glasses, or shoes, and y's *function information* that concerns its typical location (e.g. hat on the head, ring on a finger). Property-value pairs are used to define visually presentable adjectives, e.g. size-1.5 for the word “big”, and colour-(1 0 0) for “red”. Presently, CONFUCIUS has 20 property-value pairs defined.

Figure 7.4 illustrates the composition of the 3D graphic library (i.e. the visual knowledge in Figure 7.3) of CONFUCIUS. It consists of 3D object geometry files, virtual human H-Anim files, property-value pairs, and the animation library. The *animation library* defines a set of basic human animations such as “walk”, “run”, “jump”, “crouch” and “give”, by determining corresponding orientation and position interpolators for the rotations and movements of joints and body parts involved. Animation can receive parameters to adjust the movement or set the goal. For instance, the most frequently used action is “reach”. It animates a

³ Means x wears y. x is a person or personated character, y is an object.

virtual human to walk to a given location, a translation parameter, and extend his arm to put his hand in a given position, another translation parameter. After the hand has reached the defined goal, the virtual human can then carry out an action which interacts with an object within his easy reach. Object/prop models can be found in *simple geometry files*, and character models are defined in *geometry & joint hierarchy files* following the H-Anim specification.

CONFUCIUS' knowledge base is usable for many story domains, provided the corresponding visual knowledge is added. Therefore, to present a new story, it would be necessary to expand the graphics library by adding the new object/human's object model and event model. This is a substantial task, but so is the effort required to create conventionally authored storytelling presentations (e.g. films and cartoons) for a new story. Also, note that the visual knowledge should be available in VRML format⁴ to ensure that CONFUCIUS' animation engine works.

Finally, the cinematic principles that we have discussed in Chapter 6, section 6.5 are used to control camera behaviour such as automatic camera placement.

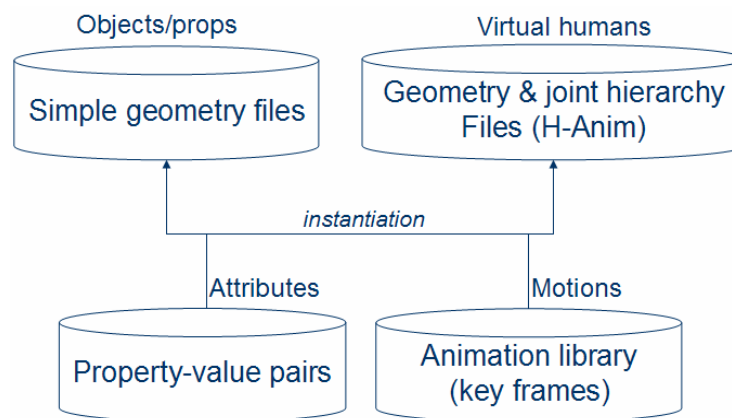


Figure 7.4: Composition of the graphic library

7.4 NLP in CONFUCIUS

The main task of the NLP module is to derive visual semantics from the natural language text. Visual semantics refers to information present in textual input which is useful in generating a visual analogue. Deriving visual semantics involves lexical, syntactic and semantic processing of text. One problem that text-to-animation applications face is the vagueness of natural language. Visual modalities always require more specific information than language modalities. Most text-to-graphics systems solve vagueness in natural language by substituting an object type with a more specific object of the type. For example, to visualise the phrase “give her a

⁴ Other common formats such as 3ds could be converted to VRML through most 3D graphic authoring and conversion tools.

toy” by substituting “a toy” with a specific one such as “a teddy bear”. In CONFUCIUS, the specific-general substitution is conducted by using lexical semantic networks in WordNet.

CONFUCIUS’ NLP mainly consists of two parts: syntactic parser and semantic analyser. The Connexor (2003) Machine Functional Dependency Grammar (FDG) parser is used for syntactic analysis, WordNet (Fellbaum 1998) and the LCS database are used as lexicons in the semantic analyser. In this section, we discuss the application of these tools in the NLP module of CONFUCIUS.

The composition of CONFUCIUS’ NLP module is shown in Figure 7.5. In the pre-processing module, surface transformation, multi-word phrases (idioms), negation patterns, and name recognition are performed. The Connexor parser then tags parts of speech, analyses morphological form, syntactic structure, and dependency relations. After syntactic parsing, the semantic analyser performs semantic inference, word sense disambiguation, anaphora resolution and temporal reasoning. The output of the NLP module is represented in LVSR, which lists each action mentioned in the input sentence, the agent that performed the action, the theme of the action, and other information such as the time or location of the action. Anaphora resolution is performed by the existing JavaRAP tool (Qiu et al. 2004), which identifies intersentential and intrasentential antecedents of third person pronouns and lexical anaphor. The temporal reasoning module analyses tenses of the input sentence and transduces events expressed in different tenses to sequential/parallel order. For example, the sentence “Having had a rich dinner, John walked along the river.” is translated into two sequential events: (1) eat (John, dinner) and (2) walk (John, along(river)).

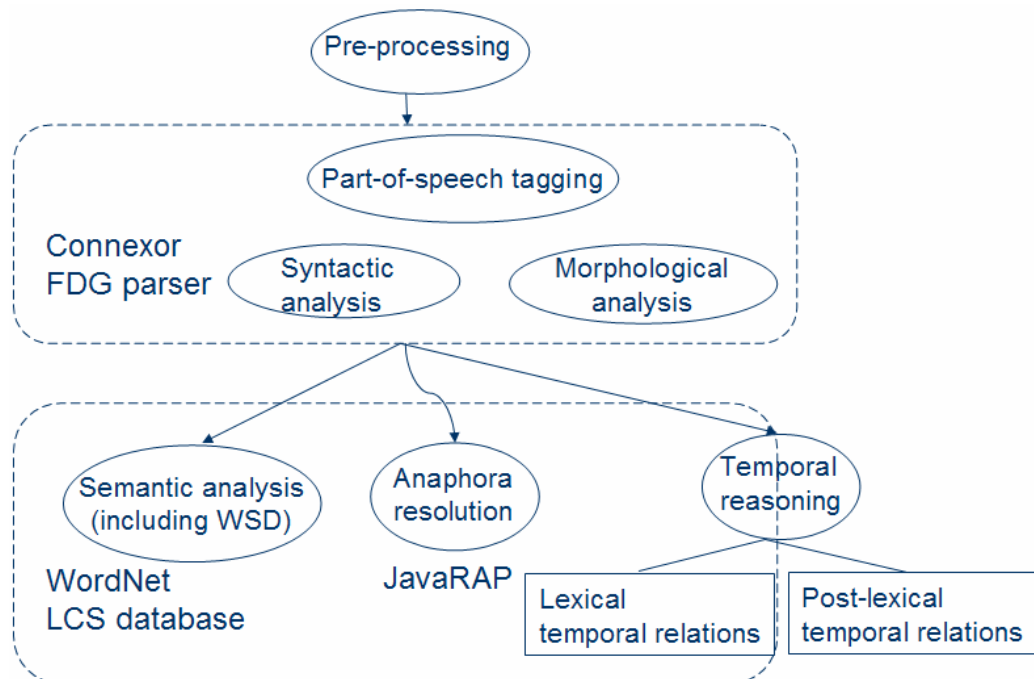


Figure 7.5: Composition of CONFUCIUS’ NLP module

7.4.1 Syntactic parsing

There is much work on parsing theory and the design and implementation of parsers. We have a variety of grammar formalisms and syntactic parsers to choose from, which are generally descended from Transformational Grammar: Government-Binding Theory (GB), Generalized Phrase Structure Grammar (GPSG), Head-Driven Phrase Structure Grammar (HPSG), Lexical Functional Grammar (LFG), and dependency grammars, e.g. Functional Dependency Grammar (FDG). FDG parsers have expressive power which exceeds Context-Free Grammars (CFG), since CFG must have a single rule for every possible word order, and hence these rules typically either over-generate or under-generate for freer word order.

The Connexor Machine syntax parser (Connexor 2003, Järvinen et al. 2004) is used in CONFUCIUS' NLP component. The parser provides morphological, syntactic and semantic information in various levels. Connexor's lexicons and grammars are based on linguistic generalisations and rules. It uses FDG based on Tesnière's Dependency Theory (1959). Texts of hundreds of millions of words have been used in testing and improving the performance of these analysers. The approach is robust; the parsers are capable of producing analysis of any input, whether well-formed sentence, sentence fragment or just a single word-token. The reliability of the analysis improves considerably when some context is provided. It analyses sentences and constructs dependency trees, where every word is a modifier of exactly one other word (called its head or modifiee), unless the word is the head of the sentence or a fragment of the sentence in case the parser failed to find a complete parse of the sentence. A dependency tree is made up of a set of dependency relationships. A dependency relationship consists of a modifier, a modifiee and a label that specifies the type of the dependency relationship.

Output of the Connexor syntactic parser for the sentence “Jack put the jug in his pocket” is presented in Figure 7.6A. Each row is a string of labels representing token, the base form of the words, word class, surface syntactic tag, dependency relation and function, and some morphological tags of a word. For example, on the 6th row, “his” is a token, “he” is its base form, “attr:>7” is its dependency relation and function, indicating that it is an attributive nominal, modifying the word “pocket” at position 7. “@A> %>N PRON PERS GEN SG3” is syntactic and morphological tags, where “@A>” denotes pre-modifier of a nominal, “%>N” denotes determiner or pre-modifier of a nominal, “PRON” denotes pronoun, “PERS” denotes personal (pronoun), “GEN” denotes genitive, and “SG3” denotes singular third person. For a description of the tags see Appendix D. The dependency relationships between nuclei in Figure 7.6A form the dependency tree depicted in Figure 7.6B where the head element of the sentence is the root of the tree.

Shortcomings of Connexor Machine Syntax Parser

Here we will analyse some cases of incorrect parsing of Connexor Machine and discuss their solutions in CONFUCIUS.

(1) Confusion of prepositional phrase attachment and post-modifier of object

The following examples show that the Connexor parser often marks prepositional phrase attachment (Example 1B) as the object's pos-modifier (Example 1A). The phrase “for bass” should be dependent on the main verb indicating purpose, while in Connexor Parser's output it is a post-modifier of “the lake”.

Example 1: Lewis trolled the lake for bass.

A. The output of Connexor Parser

```

1   Lewis lewis subj:>2      @SUBJ %NH N NOM SG
2   trolled troll main:>0    @+FMAINV %VA V PAST
3   the  the  det:>4         @DN> %>N DET
4   lake lake  obj:>2        @OBJ %NH N NOM SG
5   for  for   mod:>4        @<NOM %N< PREP
6   bass bass  pcomp:>5      @<P %NH N NOM
7   .      .
8   <s>   <s>

```

B. The expected output of syntax parser

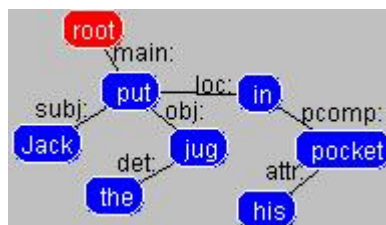
```

1   Lewis lewis subj:>2      @SUBJ %NH N NOM SG
2   trolled troll main:>0    @+FMAINV %VA V PAST
3   the  the  det:>4         @DN> %>N DET
4   lake lake  obj:>2        @OBJ %NH N NOM SG
5   for  for   ha:>2         @<NOM %N< PREP
6   bass bass  pcomp:>5      @<P %NH N NOM
7   .      .
8   <s>   <s>

```

#	Text	Baseform	Syntactic relation	Syntax and morphology
1	Jack	jack	subj:>2	@SUBJ %NH N NOM SG
2	put	put	main:>0	@+FMAINV %VA V PAST
3	the	the	det:>4	@DN> %>N DET
4	jug	jug	obj:>2	@OBJ %NH N NOM SG
5	in	in	loc:>2	@ADVL %EH PREP
6	his	he	attr:>7	@A> %>N PRON PERS GEN SG3
7	pocket	pocket	pcomp:>5	@<P %NH N NOM SG
8	<s>	<s>		

A. Text output



B. Dependency tree

Figure 7.6: Connexor output for “Jack put the jug in his pocket.”

Example 2 shows that the Connexor parser can only recognize the closer chunk as the main verb's dependent. In the sentence “Lewis tunnelled for socks in the drawer”, the phrase “in the drawer” is identified as a modifier of “socks”, whereas in “Lewis tunnelled in the drawer for socks”, the phrase “for socks” is identified as a modifier of “drawer”. Therefore, the order of

chunks in input sentences affects the parser’s output. This causes problems for parsing sentences with multiple-arguments (roles) verbs such as “troll” and “tunnel” in the examples and conflicts with the essential idea of Dependency Theory (Tesnière 1959) that the variation in word order in the sentence does not affect the structural analysis when the syntactic function of the words remain the same.

Example 2:

A. Lewis tunnelled for socks in the drawer.

```

1   Lewis lewis subj:>2      @SUBJ %NH N NOM SG
2   tunnelled tunnel main:>0 @+FMMAINV %VA V PAST
3   for   for   ha:>2        @ADVL %EH PREP
4   socks sock  pcomp:>3     @<P %NH N NOM PL
5   in    in    mod:>4       @<NOM %N< PREP
6   the   the   det:>7       @DN> %>N DET
7   drawer drawer pcomp:>5   @<P %NH N NOM SG
8   .     .
9   <s>  <s>

```

B. Lewis tunnelled in the drawer for socks.

```

1   Lewis lewis subj:>2      @SUBJ %NH N NOM SG
2   tunnelled tunnel main:>0 @+FMMAINV %VA V PAST
3   in    in    loc:>2       @ADVL %EH PREP
4   the   the   det:>5       @DN> %>N DET
5   drawer drawer pcomp:>3   @<P %NH N NOM SG
6   for   for   mod:>5       @<NOM %N< PREP
7   socks sock  pcomp:>6     @<P %NH N NOM PL
8   .     .
9   <s>  <s>

```

These errors of syntactic analysis are corrected in semantic analysis by consulting semantic components of the main verb in the LCS database (see Figure 7.7). For instance, the THETA_ROLES of “troll” specifies three roles: *agent*, *purpose* which follows the preposition “for”, and *location*, and a purpose is obligatory. When the semantic analyser processes the dependency tree like Example 2B, it will correct the dependency relationship of the purpose phrase “for socks”.

(2) Dealing with phrasal adverbs and idioms

Multiword expressions are a bottleneck problem in NLP. Machine Syntax cannot deal with multiword expressions such as phrasal adverbs and idioms properly. Instead of being recognized as multiword units, phrases such as “once upon a time” are treated as separate units in tokenisation and lexical analysis in Machine Syntax (Figure 7.8). In CONFUCIUS, this problem is solved by a pre-processing routine recognising these phrases and tagging them as one multiword token. The routine scans input sentences and checks with a phrasal adverbs and idioms collection file to detect these multiword expressions. Alternatively, it can be solved by adding entries to Machine Syntax’s custom lexicon.

7.4.2 Semantic analysis

To represent semantic structure in LVSR, we need some semantic features which the Machine Syntax’s dependency structure does not provide. In this section, we will discuss the

semantic features we added for nouns, verbs, and adjectives in the semantic analyser of NLP module.

```
(
:DEF_WORD "troll"
:CLASS "35.2.a"
:WN_SENSE (("1.5" 00648291) ("1.6" 00775788))
:PROPBANK ("arg0 arg3(for) argm-LOC(in/on - up.)")
:THETA_ROLES ((1 "_ag_purp(for),loc()"))
:LCS (act loc (* thing 1) ((* [in] 10) loc (*head*) (thing 11))
      ((* for 21) intent (*head*) (thing 22)) (troll+ingly 26))
:VAR_SPEC ((10 :optional) (21 :obligatory))
)
```

A. Word entry of “troll”

```
(
:DEF_WORD "tunnel"
:CLASS "35.5"
:WN_SENSE (("1.5" 01167301) ("1.6" 01395316))
:PROPBANK ("arg0 arg3(for) argm-LOC(in/on - up.)")
:THETA_ROLES ((1 "_ag_purp(for),loc()"))
:LCS (act loc (* thing 1) ((* [in] 10) loc (*head*) (thing 11))
      ((* for 21) intent (*head*) (thing 22)) (tunnel+ingly 26))
:VAR_SPEC ((10 :optional) (21 :obligatory) (1 (animate +)))
)
```

B. Word entry of “tunnel”

Figure 7.7: Example verb entries in the LCS database



Figure 7.8: Dependency tree of “Once upon a time there was a poor widow.”

For nouns, we use WordNet, a proper noun list, and a popular given name and surname list to mark proper nouns, persons and their gender. The proper noun is recognized by searching the proper noun list which includes places, months, days of the week, holidays, countries and languages. The features of person and gender are identified either by searching the popular name list (e.g. “Jane” is a female person; “Andrew” is a male person) for proper nouns, or by looking up WordNet’s hypernym trees (e.g. a “chef” is a male person, by default; and a “witch” is a female person.) for common nouns. These semantic features of nouns are requisite for visualisation procedures in CONFUCIUS’ animation generation module. The semantic analyser also checks the graphic library to find out if a 3D model of the noun in the original text input is available. If its 3D model is not available, the program will search its hypernyms in WordNet

until it finds an available 3D model. For example, we have an input sentence “John left the cottage”. There is no 3D model of cottage in the graphic library. The semantic analyser searches WordNet for its hypernyms and finds “house”. Because a house model is available in the graphic library, the “cottage” is substituted by a “house”, and the final animation presents “John left a house”. This approach ensures full usage of the limited graphic resources and requires minimal user interaction.

The semantic features of verbs indicate the verb categories shown in Figures 5.3 and 5.4. This information is used in media allocation and animation generation in later processing. For instance, *verbs of speaking or manner of speaking* causes the part enclosed in quotation marks in the sentence to be transferred to the TTS engine and the simultaneous lip movement of the speaker in generated 3D animation. The hypernym substitution approach is also applied to verbs (e.g. using “run” to replace “trot” if the animation for the latter one is not available) to ensure the maximal usage of the animation library and minimal user interaction.

Adjectives’ semantic features indicate visually and audio presentable properties that were listed in Chapter 4, section 4.7.3, such as size, length, width, thickness, height, depth, shape, speed, colour, numerousness, gender and age. The semantic analyser tags these properties of ascriptive adjectives during NLP. Appendix B shows some working files of semantic analysis for an example sentence. Appendix B.1 is the output of the syntactic parser, and Appendix B.2 shows an example output file after addition of semantic features.

7.4.3 Using WordNet for semantic inference and WSD

To fulfil the task of semantic inference and Word Sense Disambiguation (WSD), we use WordNet 2.0 as the lexicon. The conceptual relations distinguished between WordNet’s synsets are useful for semantic inference in language visualisation. `Hypernym` and `hyponym` relations are frequently used for semantic inference when the language input is too general, for WSD as was discussed in Chapter 5, section 5.4, and for coreference resolution. For example, `{toy}` `HYPONYM` `{teddy bear}` may bridge the gap in visualising the phrase “give her a toy” by substituting “a toy” with a specific toy “a teddy bear”; while the relation `{teddy bear}` `HYPERNYM` `{toy}` may resolve the reference in the context “John gave her *a teddy bear*. She was happy to get *the toy*.” and hence reason that the toy is referring to the teddy bear. In addition, hypernyms and hyponyms can help solve the different granularity between language models and graphic models. In CONFUCIUS, language models have finer granularity than the corresponding graphic models. For example, “cottage” and “mansion” have to share the same graphic model of “house”. The hypernym relationship provides a facility for this type of inference. The `value of` relation in WordNet is used in the semantic analyser for identifying adjectives’ properties. Figure 7.9 lists the beginning of the `adjProperty()` method that identifies the visually and auditory presentable properties: size, length, width, thickness, height, depth, shape, speed, colour, quantity, age, and volume.

Furthermore, WordNet has 35 simple skeletal argument-frames for verbs as listed in Figure 7.10. The distinction between human (sb) and non-human (sth) fillers of the frame-slots represents a shallow type of selection restriction. These frames provide the constituent structure of the complementation of a verb, where --s/--ing represents the verb and the left and right strings the complementation patterns.

```

/* return the adj's property: SIZE, LENGTH, WIDTH, THICK, HEIGHT,
DEPTH, SHAPE, SPEED, COLOUR, QUANT, AGE, VOLUME, or an empty string
for none of these */
public static String adjProperty(String wnOutfile)
{
    String line = null;
    String cat = "";
    String[] CATS = {"SIZE", "LENGT", "WIDTH", "THICK",
"HEIGHT", "DEPTH", "SHAPE", "SPEED", "COLOUR", "QUANT", "AGE",
"VOLUME"};
    String[] catsFeature = {
        "size",
        "length",
        "width",
        "thickness",
        "height",
        "depth",
        "shape, form",
        "speed, swiftness, fastness",
        "color, colour, coloring, colouring",
        "numerousness, numerosity, multiplicity",
        "age",
        "volume, loudness, intensity"
    };
    .....
    return cat;
}

```

Figure 7.9: Using “value of” relation to look for adjectives’ properties

- | | |
|-------------------------------|-------------------------------|
| 1. Sth --s | 19. Sb --s sth on sb |
| 2. Sb --s | 20. Sb --s sb PP |
| 3. It is --ing | 21. Sb --s sth PP |
| 4. Sth is --ing PP | 22. Sb --s PP |
| 5. Sth --s sth Adjective/Noun | 23. Sb's (body part) --s |
| 6. Sth --s Adjective/Noun | 24. Sb --s sb to INFINITIVE |
| 7. Sb --s Adjective | 25. Sb --s sb INFINITIVE |
| 8. Sb --s sth | 26. Sb --s that CLAUSE |
| 9. Sb --s sb | 27. Sb --s to sb |
| 10. Sth --s sb | 28. Sb --s to INFINITIVE |
| 11. Sth --s sth | 29. Sb --s whether INFINITIVE |
| 12. Sth --s to sb | 30. Sb --s sb into V-ing sth |
| 13. Sb --s on sth | 31. Sb --s sth with sth |
| 14. Sb --s sb sth | 32. Sb --s INFINITIVE |
| 15. Sb --s sth to sb | 33. Sb --s VERB-ing |
| 16. Sb --s sth from sb | 34. It --s that CLAUSE |
| 17. Sb --s sb with sth | 35. Sth --s INFINITIVE |
| 18. Sb --s sb of sth | |

Figure 7.10: Verb-frames in WordNet

There is a mapping between the two lexical resources of theta roles of the LCS database and WordNet verb frames. Both reflect how many and what kinds of arguments a verb may

take. However, they take rather different approaches in conveying this information. The LCS database makes use of theta roles to list arguments and their types in an integrated unit. An example is the theta roles `_ag_purp(for), loc()` (i.e. agent, purpose, location) of “troll” in Figure 7.7. WordNet lists all the frames a verb sense may be found in. The LCS database distinguishes 67 individual theta roles (e.g. `ag, th, instr`), while, by way of contrast, WordNet's smallest syntactic unit is the frame, of which 35 are used. This suggests that the integration of argument components into theta roles is more systematic and more informative than the verb frames in WordNet.

Although we found that theta roles in the LCS database are more informative than verb frames, the latter provides a finer specification in human roles. Consider the class of “hit verbs”, the verb frames in Figure 7.11B specify whether an argument is a person or a thing, whereas theta roles in Figure 7.11A only indicate their semantic roles and specify the agent is an animate thing in `VAR_SPEC`. Therefore we use the information from WordNet verb frames when theta roles aren't enough for WSD.

```
(
  :NAME "Hit Verbs - Change of State / -with"
  :WORDS (bang bash batter beat bump butt dash drum hammer hit kick
knock ...)
  :THETA_ROLES ((1 "_ag_th,mod-loc(),instr(with)"))
  :SENTENCES "She !!+ed him (on the arm) (with a stick)"
  :LCS (act_on loc (* thing 1) (* thing 2)
        ((* [on] 23) loc (*head*) (thing 24))
        ((* with 19) instr (*head*) (thing 20))
        (!!+ingly 26))
  :VAR_SPEC ((1 (animate +)) (26 :conflated))
)
```

A. Theta roles of hit verbs

```
Sb ----s sth
Sb ----s sb
Sth ----s sb
Sth ----s sth
Sb --s sth with sth
Sb --s sb with sth
.....
```

B. Verb frames of hit verbs

Figure 7.11: Theta roles and verb frames of hit verbs

Finally, semantic relations in WordNet can be augmented with specific features to differentiate the precise semantic implication expressed. Figure 7.12 shows some relations with such augmented features. `HAS_MERO_PART` is the relation between an object and its parts, while `HAS_HOLO_PART` is the relation between one part and the object of the whole. The knowledge in these relations is an advantage for semantic inference in the disambiguating process for visualisation, for example, when a door is mentioned in language input, the animation generator needs to know which door, a door of a car, a door of a room, or a door of an airplane, should be

created in the story world. Figure 7.12D shows the negation of implications expressed by relations, which provides a potential to automatically extend the visual library. In the example of Figure 7.12D, if the relation `monkey NEAR_SYNONYM ape` is known with the negation of implications expressed by the relations in the figure, it is possible to extend the visual knowledge of *ape* in the case where the original visual library has only *monkey* but no *ape*. This learning process is similar to a child's when he is told that "ape is like monkey but it has no tail".

7.4.4 Action representation

To summarise Theta roles, Jackendoff's LCS EVENT parameters, and Badler et al. (1997)'s Parameterized Action Representations (PARs), an action may be specified by the following parameters: *agent/experiencer*, *objects*, *precondition*, *result state*, *spatiotemporal*, *manner*, *instruments*, and *sub-actions*. Precondition and result state are conditions that must exist before or after the action can be performed, e.g. reachability precondition for verbs of putting, and the *theme at the goal* as result. Spatiotemporal information may use LVSR's PATH/PLACE predicates. Subactions can be specified using the VHML. To fully use the theta roles information in the LCS database for disambiguation, we design the following event structure to integrate theta roles and LVSR in our implementation:

```
[ EVENT
  agent:
  theme:
  space/time:
  manner:
  instrument:
]
```

<p>A. HAS_MERO_PART</p> <p style="padding-left: 40px;"><i>airplane</i> HAS_MERO_PART <i>door</i></p> <p style="padding-left: 40px;"><i>airplane</i> HAS_MERO_PART <i>engine</i></p> <p>B. HAS_HOLO_PART</p> <p style="padding-left: 40px;"><i>door</i> HAS_HOLO_PART <i>car</i></p> <p style="padding-left: 40px;"><i>door</i> HAS_HOLO_PART <i>room</i></p> <p style="padding-left: 40px;"><i>door</i> HAS_HOLO_PART <i>airplane</i></p> <p>C. Causal relations</p> <p style="padding-left: 40px;"><i>kill</i> CAUSES <i>die</i> factive</p> <p style="padding-left: 40px;"><i>search</i> CAUSES <i>find</i> non-factive</p> <p>D. Negation of implications expressed by relations</p> <p style="padding-left: 40px;"><i>monkey</i> HAS_MERO_PART <i>tail</i></p> <p style="padding-left: 40px;"><i>ape</i> HAS_MERO_PART <i>tail</i> not</p>
--

Figure 7.12: Relations with features to differentiate semantic implication

Some working examples of this structure are shown in Figure 7.13. The structure also provides a way to specify properties of objects and humans. The content bracketed in {} gives

attributes of the object or human it follows, e.g. `john{male}`, `chair{high}`, `box{black}`. The common attributes include, for example, colour, size, gender, and age. The information is used in the animation generation module.

7.4.5 Representing active and passive voice

The differences between active and passive voice are not only syntactic but also semantic: the subject of an active sentence is often the semantic agent of the event described by the verb (e.g. “*He* received the letter”) while the subject of the passive is often the undergoer or patient of the event (e.g. “*The letter* was received”), i.e. the *topic* of active voice is the performer but the topic of passive voice is the undergoer. In CONFUCIUS’ visualisation, the semantic difference of voice is represented by *point of view*, the perspective of the viewer in the virtual world. Since the virtual world in CONFUCIUS is modelled in VRML, *Viewpoint node* is used to represent voices. With the Viewpoint node, one can define a specific viewing location for a scene like a camera. In the previous example, although the two sentences describe the same event, *receiving the letter*, in active voice the focus is the person who received it while in passive voice it is the letter. Therefore the modelling of the event and concerned object/character for the two sentences are identical, with the only difference being the parameters (orientation and position) of Viewpoint node to represent the topic in each voice.

[EVENT give agent:(nancy{female}) theme:(bread) spatial/time:(to(jim{male}))]	[EVENT walk theme:(he{male}) spatial/time:(from(house))]
[EVENT push agent:(john{male}) theme:(door)]	[EVENT sit agent:(john{male}) spatial/time:(to(on(chair{high})))]
[EVENT carve agent:(emma{female}) theme:(turkey) instrument:(with(knife))]	[EVENT pick agent:(jack{male}) theme:(box{black})]

Figure 7.13: Event structures used in CONFUCIUS

Besides voice, Viewpoint node may also present *converse* verb pairs, e.g. “give”/“take”, “buy”/“sell”, “lend”/“borrow”, “teach”/“learn”. They refer to the same activity but from the viewpoint of different participants.

7.5 Media allocation

Multimedia integration requires the selection and coordination of multiple media and modalities. The selection rules are generalised to take into account the system’s communicative goal, features characterising the information to be displayed and features characterising the

media available to the system. To tell a story by complementary multimodalities available to CONFUCIUS the system divides information and assigns primitives to different modalities according to their features and cognitive economy. Since each medium can perform various communicative functions, designing a multimedia presentation requires determination of what information is conveyed by which medium at first, i.e. allocating contents to media according to *media preferences*. For example, presenting spatial information like position, orientation, composition and physical attributes like size, shape, colour by visual modality; presenting events and actions by animation; presenting dialogue/monologue and temporal information like “ten years later” by speech; presenting dog bark by both audio and visual modalities (or by audio icon solely). We formulate the principles for media allocation within CONFUCIUS as the following:

1. Realise spatial information, physical attributes, physical actions and events in 3D animation.
2. Realise dialogues, monologues, and abstract concepts including abstract actions and abstract relations in speech, i.e. voiceover narrative. For example, if the media allocator detects an unobservable adjective that can not be presented visually in the narration part of a story, the sentence is sent to the presentation agent and is output in Merlin the narrator’s speech and gestures, while the other visually presentable parts of the sentence are still allocated to the animation engine and TTS to create animations which will be played when Merlin is talking.
3. Realise (or augment other modalities) sound emission verbs in Figure 4.4 and audio representable adjectives in Figure 3.22 in audio modality.
4. Realise failed attempts (e.g. animation files not available) and successful attempts with low confidence in principle 1 in other modalities according to the feedback from the animation engine.

Figure 7.14 shows CONFUCIUS’ multimedia presentation planning. Media allocation receives feedback from media realisation, such as the animation engine, to influence the selection of media for a specific content. Thus failed realisation at a later stage of processing can propagate back to undo an earlier decision. For example, realisation in the animation engine may fail because of visualisation difficulties, and this message should be fed back to the *media allocator*, where the content could be re-allocated to other media. Currently, this feedback and re-allocation facility has not yet been implemented in CONFUCIUS.

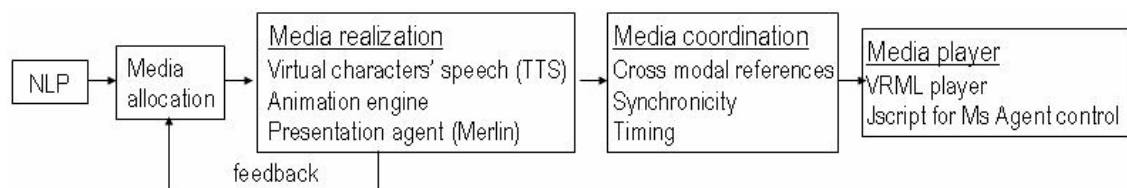


Figure 7.14: CONFUCIUS’ multimedia presentation planning

Having solved the problem of media selection (How to present information?), we should deal with the integration and coordination problem, i.e. how should the presentation be arranged, *in space* and *in time*? Media coordination in CONFUCIUS concerns three problems: (1) temporal coordination between animation and speech (e.g. lip-speech synchronising), (2) cross-modal reference, and (3) duration constraints for different media. Similar to static multimedia presentation, the cross-modal reference in temporal media representation resolves identification of referents. For instance, Merlin could refer to several virtual humans that appear in the animation by mentioning their names and action/characteristic. A coherent presentation should enable users to identify each one easily. Duration constraints require that the duration of actions which occur in different temporal media be coordinated, for example, the duration constraints of animation and its auditory icon, or scheduling Merlin and virtual humans' speech/movements. Finally, the media player consists of VRML player and Javascript for Microsoft Agent control. The former is for playing virtual characters' animation and speech, and the latter is for playing Merlin's speech and behaviours.

7.6 Animation engine

Larsen and Petersen (1999) analysed three possible ways to implement 3D animations. First is the classical approach where the graphical engine translates, rotates and scales each individual part of the object. Creating animations in this way is similar to what Disney animators do. A file containing the animation must be parsed to the animation part of the graphical modality in this method. Second is inverse kinematics (IK) which was introduced in chapter 5. The third is to import pre-created animations made in a 3D-Studio like Character Studio or from motion capture data. Hence one can create the animations in a tool which has been designed to produce animation of 3D objects. In CONFUCIUS, human animations are pre-created manually or by Character Studio and exported to VRML 97 format (VRML 2002). 3D models available on the Internet are used, particularly H-Anim models from Babski (2000), to save substantial effort on graphic design. Microsoft Agent (2002) Merlin is used for the narrator in CONFUCIUS' storytelling since it provides adequate movements for presentation agents.

Figure 7.15 illustrates the flowchart of CONFUCIUS' animation engine algorithm. *User interaction, human-object interaction, environment placement, and camera control*, shown in rounded rectangles in the figure, are optional processes. The animation engine receives LVSR representations and checks whether they contain simultaneous motions. If they do, the engine then checks the animation registration table to decide if the simultaneous motions are mutually exclusive. If they are not mutually exclusive or there is only one event specified in the semantic representation, the animation engine tries to match the event predicate(s) to available actions in the *animation library*; and if the motions are mutually exclusive or the predicate(s) are not available, the user is prompted for interaction. The user can either load a keyframe animation or provide a VHML specification for animation generation. The *animation controller* then

instantiates keyframing information in the animation library to the motion bearer (an OBJ) or agent (a HUMAN) and schedules the execution of the sequence of basic actions (i.e. timing). It also deals with applying PATH to the keyframing information of motions. For instance, besides joints' rotation the motion *climb* in “climb a tree” is a vertical upward movement whereas in “climb the mountain” or “climb through the tube” (*climb* + PP) is a slope upward movement and the slant depends on the surface feature of the object. In the animation library, only the joint rotations of *climb* are defined. The animation controller needs to add an appropriate *upward positionInterpolator* and rotate the whole body of the character to suit the slope if necessary. *Human-object interaction* uses object knowledge to adjust human motions for 3D object manipulation, such as grasping planning, i.e. applying different hand postures for grasping according to the functional information and grasping site of the object. *Environment placement* applies spatial information and places objects and virtual characters into a specified environment. Finally, *camera control* handles automatic camera placement and applies cinematic rules to guide the viewpoint.

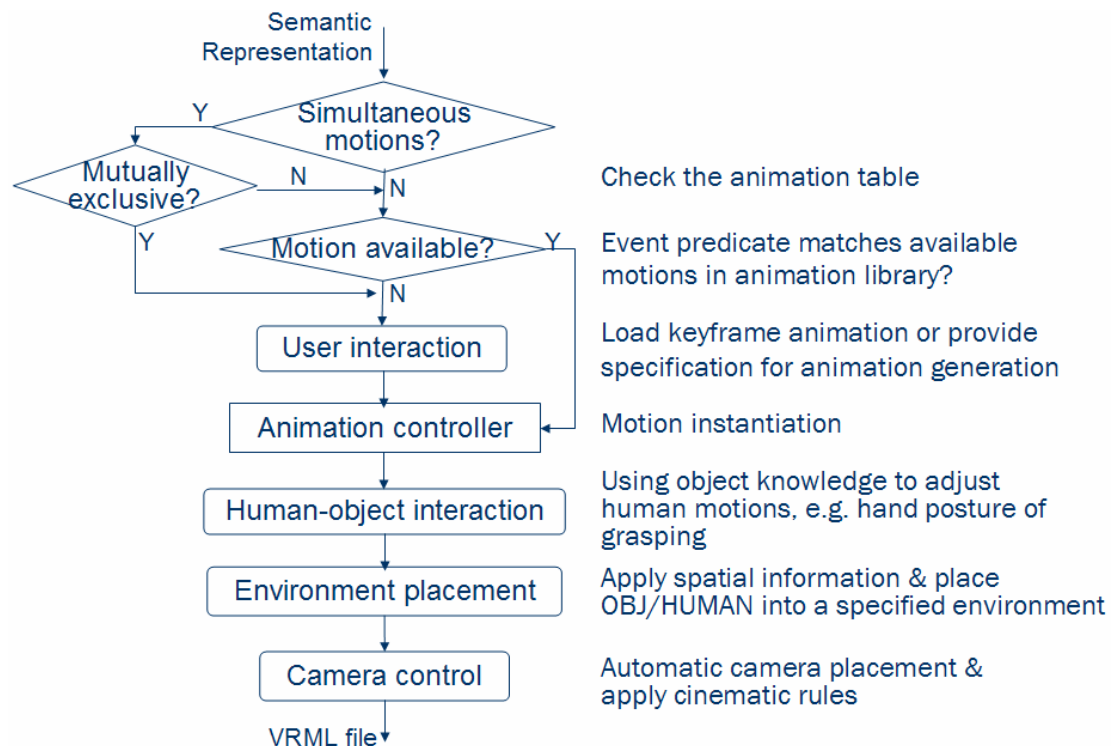


Figure 7.15: Flowchart of animation engine algorithm

According to our verb taxonomy shown in Figure 5.3, VRML's linear interpolators (*ColorInterpolator*, *PositionInterpolator*, *OrientationInterpolator*, and *ScalarInterpolator*) for keyframed animation is adequate for *atomic events*. They can be used to change the values of the attributes of objects like colour, position, and size. Precreated animations or blended animations based on precreated ones are used to present human action verbs. One difficult task is to visualise an event whose goal (target) is changing. For instance, the goal *house* in “John walked towards the house” is still whereas the goal *thief* in “John chased the thief” is moving. It is requisite for VRML to know the target value of each interpolator in order to calculate

intermediate values. For the situations of *chasing verbs* with moving goals and same walking speed of the agent and the theme, e.g. the verb “follow”, we use one `PositionInterpolator` for both the target and the follower, so they can keep the same pace and a suitable distance between them (one metre in the example code). Figure 7.16 shows an example VRML code of *following* verbs and Figure 7.17 a snapshot of its animation. The italics in Figure 7.16 show that the `PositionInterpolator` is shared by both the target and the follower.

Figure 7.18 shows an example VRML code for *chasing* verbs, e.g. “catch up”. The follower and the target have a different pace, each character has a `PositionInterpolator`, and the follower’s destination `keyValue` is dynamically changed by a script. It is calculated by subtracting the follower’s original position from the target’s destination position and leaving a one-metre distance at the meeting point.

```

DEF target Transform {
  children Inline {url "h-anim\nanaWalk.wrl"}
}
Transform {
  translation 0 0 -1 # one metre distance between target &
  follower
  children
  DEF follower Transform {
    children Inline {url "h-anim\bobWalk.wrl"}
  }
}
DEF path PositionInterpolator {
  key [0,1]
  keyValue [0 0 0, 0 0 5 ]
}

DEF clock TimeSensor {
  cycleInterval 4.0
  loop FALSE
}
ROUTE clock.fraction_changed TO path.set_fraction
ROUTE path.value_changed TO target.translation
ROUTE path.value_changed TO follower.translation

```

Figure 7.16: Example VRML code for *following* verbs

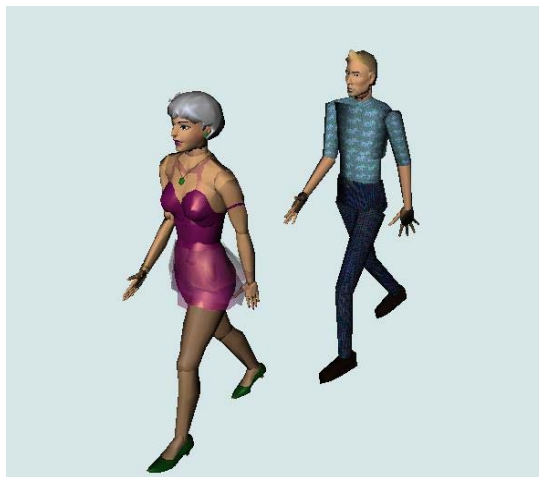


Figure 7.17: Snapshot of animation for *following* verbs

```

DEF target Transform {
    children Inline {url "h-anim\bobWalk.wrl"}
}

DEF followerP Transform {
    translation 0 0 -6 # follower's original position
    children
        DEF follower Transform {
            children Inline {url "h-anim\nanaWalk.wrl"}
        }
}

DEF rPath PositionInterpolator {
    key [0,1]
    keyValue [0 0 0, 0 0 4 ]
    # 0 0 4 is the destination of the target, i.e. the meeting point
    # where the follower catches up with the target
}

DEF fPath PositionInterpolator {
    key [0,1]
    keyValue [0 0 0, 0 0 0]
    # the destination keyValue is changed in the script. It is rPath's
    # destination keyValue - follower's original position - 1
}

DEF clock TimeSensor {
    cycleInterval 4.0
    loop FALSE
}

DEF v1 Viewpoint {
    position 5.9 1 -.4
    orientation 0 1 0 1.57
    description "narrator's view"
}

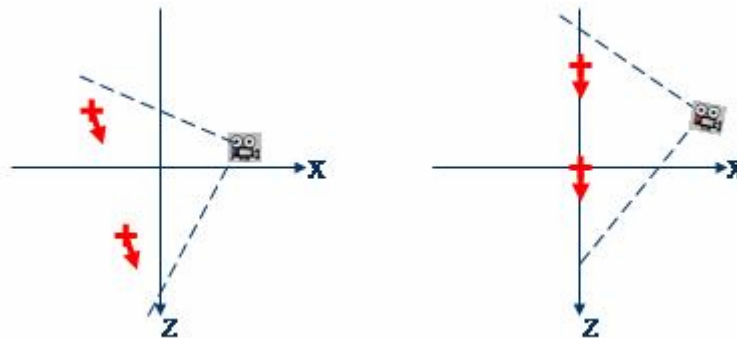
DEF s Script {
    field SFNode node USE followerP
    field SFNode path1 USE rPath
    field SFNode path2 USE fPath
    eventIn SFTime startRun
    directOutput TRUE
    url "vrmlscript:
    function startRun() {
        inc = new SFVec3f(0, 0, 1);
        tmp = path1.keyValue[1].subtract(node.translation);
        path2.keyValue[1] = tmp.subtract(inc);
    } "
}

ROUTE start.touchTime TO clock.startTime
ROUTE start.touchTime TO s.startRun
ROUTE clock.fraction_changed TO rPath.set_fraction
ROUTE clock.fraction_changed TO fPath.set_fraction
ROUTE rPath.value_changed TO target.translation
ROUTE fPath.value_changed TO follower.translation

```

Figure 7.18: Example VRML code for *chasing* verbs

In this example the virtual characters are designed to move only along the z axis, though they are free to move on the x-z plane as long as there is no obstacle on their course. It is possible to achieve the same effect by setting an appropriate viewpoint. Figure 7.19 illustrates two situations with different viewpoints and virtual human movements. The cameras indicate the position and the field-of-view of viewpoints, the “+” sign indicates virtual human position, and arrows indicate the direction of their movement. The viewpoint in Figure 7.19B sees the same scene as the one in Figure 7.19A, whereas Figure 7.19B saves computation of the orientation of the follower since virtual humans should face the direction in which they go. Apart from the moving goal, the example of presenting “chase” verbs reveals an underlying problem of interaction and coordination of multiple virtual humans, involving two valency verbs (class 2.2.1.2.2. two humans, e.g. “fight”, “chase”, “guide”) or three valency verbs (class 2.2.1.3.1. two humans and one object, including ditransitive verbs, e.g. “give”, “buy”, “sell”, “show”, and transitive verbs with an instrument, e.g. “beat (sb with sth)”) in Figure 5.3.



A. Free chasing movement

B. Chasing movement along the z axis

Figure 7.19: Two situations with same effect by placing appropriate viewpoints

7.6.1 3D object modelling

A graphic modelling language is required to represent visual data symbolically and combine the prefabricated visual primitives created by authoring tools to create a virtual world. The representation of visual information of a 3D object usually consists of its colour, texture, geometric shape, size, position, orientation and its sub-components. The syntax of VRML and its hierarchical structure suit for the representation of the visual information aforementioned. In this section, we discuss how we use VRML to model the story worlds in CONFUCIUS.

Using Background node to build stage setting

VRML provides stage-like facilities that suit story presentation. Background binding facilitates similar operations to scene-changing in plays. When a browser initially reads in the VRML file, it binds the first Background node it finds. It pushes that node onto the top of the Background stack, and the node issues an `isBound` outgoing event with value `TRUE`. The browser doesn't automatically bind any Background nodes other than the first one; thus, the background displayed when the user first arrives in the world is the first background listed in the file, which

acts like the background of the first act in a play. When a particular Background node that isn't already at the top of the stack receives a `set_bind` event with value `TRUE`, the browser places that node on top of the Background stack and makes it the current Background. The previously displayed background is replaced with the one described in the newly bound node. The Background node previously at the top of the stack (now in the second position on the stack) sends an `isBound` event with value `FALSE`; and the new current Background node sends an `isBound` event with value `TRUE`. If the newly bound node was already somewhere on the Background stack, it is moved from wherever it was in the stack to the top. When a Background node anywhere in the stack receives a `set_bind` event with value `FALSE`, it is removed from the stack, whether or not it's at the top of the stack. If there is only one background in the world, it never needs to be explicitly bound or unbound because the first Background node in the file is automatically bound.

The Background node is more powerful than conventional theatre design since it provides background for all six sides of the stage, that is, besides setting the back background (the usual one) it also allows setting of front, left, right background, top background (for sky), and bottom background (for the ground). To present a multi-act play or a story with different backgrounds in each part, we may list all Background nodes in the beginning of the VRML file and bind one after each act in order. A more advanced feature of the Background node than that in traditional drama is the background could be changed within an act using the ordinary event-and-route method. This feature enlarges the usage of background and enables us to put some properties in the background and hence reduce the 3D rendering of unimportant objects to 2D imaging.

Using interpolators and ROUTE to produce animation and build trigger system

VRML defines piecewise linear interpolators (e.g. `ColorInterpolator`, `PositionInterpolator`, `OrientationInterpolator`, `ScalarInterpolator`) for keyframe animation. They can be used to change values of objects' physical properties like position, orientation, colour and size. The ROUTE statement in VRML is a construct for establishing event paths between nodes, i.e., an object (node) may generate events (`eventOut`) and send it to any other objects (`eventIn`) connected via a ROUTE statement. This message-passing facility enables communication between characters and objects. Figure 7.20 shows two examples of ROUTE statement: `TIMER1` controls a box's movement and `TIMER2` starts a character's action. We will investigate the details of interchangeable human animation in the next section.

```
ROUTE TIMER1.fraction_changed TO boxPositionIntp.set_fraction
ROUTE TIMER2.isActive TO nancyAction.set_animationStarted
```

Figure 7.20: Examples of ROUTE statement

Using Viewpoint node to guide users' observation

Viewpoint node is another useful facility provided by VRML. First, it may be used to visually represent semantic difference between active and passive voice in input natural language (section 6.4.5). Secondly, animating the position and orientation of a viewpoint achieves special movie-editing effects like *cut* or to guide the viewer to explore around in the world. Figure 7.21 shows two examples of animated viewpoint. The first example sets the `jump` field of a Viewpoint to `TRUE`, and hence cuts the scene to the next location (bus stop 1). Another method of guiding viewpoints is binding a viewpoint to a user who is in a moving vehicle or conveyance, a train, for instance, or an elevator to guide tours and hence presents a story in the first person *I*. The viewer, also the narrator in this case, passes through any of the space between the former location and the new location, and arrives there gradually. The second example in Figure 7.21 guides the viewer's sight gradually as if (s)he is in an elevator. Most stories are told in the first person (i.e. a character's point-of-view) and the third person (i.e. a neutral non-character point-of-view). Setting a viewpoint properly can fulfil the narrative on any desired perspective in storytelling.

```

DEF CUT_TO_STOP1 Viewpoint {
    position 10 5 30
    fieldOfView 1.8
    description "Initial viewpoint, cut to bus stop 1"
    jump TRUE
}

DEF IN_ELEVATOR Viewpoint {
    position 38 17 -40
    orientation 0 1 0.05 2.36
    fieldOfView 0.8
    description "In elevator view"
    jump FALSE
}

```

Figure 7.21: Examples of Viewpoint node

VRML also saves efforts on media coordination since its *Sound node* is responsible for describing how sound is positioned and spatially presented within a scene. It can also describe a sound that will fade away at a specified distance from the Sound node by `ProximitySensor`. This facility is useful in presenting non-speech sound effects in storytelling. It enables us to encapsulate sound effects within object models, e.g. to encapsulate the engine hum within a car model and hence locate the sound at a certain point where the car is. The sound node enables a scene to be imbued with ambient background noise or music.

Additionally, using VRML can relieve computation in pathfinding. Pathfinding is probably the most popular AI problem in 3D games. Generally, there are two levels of pathfinding: one for high level paths concerning using strategic AI to find the best (cheapest) path to the goal (they are usually long paths, involving terrain analysis), and one for lower level obstacle avoidance issues. Since only the low level pathfinding capability is required for

CONFUCIUS' virtual characters, we rely on VRML's collision detection mechanism for avatar-object collision and ParallelGraphics' VRML extension for object-to-object collision (see section 6.4).

As a graphic modelling language VRML enables an animation system to meet the following requirements: (1) be able to create objects, both their geometric shape and their autonomous motion behaviour, and to pass messages to objects to alter their properties and behaviour; objects also have to be able to pass messages to each other; (2) be able to facilitate programming complex behaviour, e.g., the motions of biped kinematics, and a library of versatile motion methods are available.

7.6.2 Virtual human animation

The virtual human animation (H-Anim) modelling language that we use for CONFUCIUS' virtual characters is a subset of VRML. It is specified by the Web3D H-Anim (H-Anim 2001) working group so developers could agree on a standard naming convention for human body parts and joints. Appendix E gives a list of existing H-Anim models available on the web, from which suitable ones are selected for storytelling according to their Level-Of-Detail (LOD), gender, and age. Based on Babski's (2000) animation prototype, an animation engine is designed, which is capable of applying human animations to various virtual character models listed in Appendix E with different LODs. Needed ROUTEs are generated dynamically based on the joint list of the H-Anim body and the joint list of the animation. Figure 7.22 shows an example external prototype inserted at the end of Nana's (see Appendix E) H-Anim file by the animation engine. It uses keyframe information in the external VRML file `..\animation\walk.wrl` to make the virtual character walk.

```
# ----- inserted by CONFUCIUS animation engine -----
EXTERNPROTO Behaviour [
eventIn SFTime LaunchAnim
exposedField SFTime set_startTime
exposedField SFTime set_stopTime
field MFNode HumansList
]"..\animation\walk.wrl"
DEF behv Behaviour {
HumansList [
USE nana
]
}
ROUTE hanim_BodyTouch.touchTime TO behv.LaunchAnim
```

Figure 7.22: External prototype of H-Anim animation

The animation file defines keyframes of all `OrientationInterpolator` and `Position-Interpolator` involved in the movement, as shown in the code in Appendix F.1. The Script node dynamically adds ROUTEs according to the list specified in `InvolvedJointNameList` and `InvolvedJointPtrList`. The matching between the animation and the body is performed by using the joints list in the Humanoid prototype.

Therefore, `InvolvedJointNameList` must have a one-to-one matching to the humanoid joints list defined in the virtual character's geometry file (see Appendix F.2 for Nana's joints list, and compare it with Appendix F.1). If the animation is applied to a different LOA character, e.g. Nancy (LOA1), and a joint is not implemented, the corresponding field should be a dummy Transform/Joint node. Nancy's joints list is given in Figure 7.23.

```

    joints [
    USE hanim_HumanoidRoot,
    USE hanim_sacroiliac,
    USE hanim_l_hip,
    USE hanim_l_knee,
    USE hanim_l_ankle,
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
    USE hanim_r_hip,
    USE hanim_r_knee,
    USE hanim_r_ankle,
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
    USE hanim_v11,
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
    .....
    USE hanim_vc4,
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
        Transform {},          # <-- Dummy Node
    USE hanim_skullbase
    .....
    ]

```

Figure 7.23: Nancy's joints list

Figure 7.24 shows that one animation applies to two virtual humans, Nancy and Baxter, with different LOAs. There is a difference in the angle of their upper bodies because Nancy uses a dummy node for the joint `hanim_v15`. The problem of angle difference may occur in virtual humans with the same LOA. Some animations can not be applied from one H-Anim body to another without making any corrections on angle values. This is mainly due to the difference between segment length and it is difficult to detect such a problem, as long as it has nothing to do with angle limitation value. It is close to a self-collision detection problem.



Figure 7.24: Applying one animation on two H-Anim bodies

7.6.3 Java in VRML Script node

A programming language (or script language) is needed in coordination with the graphic modelling language to define animation, to link events occurring on different objects/characters, and to implement advanced animated effects. Scripts embedded in VRML of CONFUCIUS are written in Java because this is one of the most commonly-supported programming languages and hence enables CONFUCIUS to have maximum portability across VRML browsers.

There are two specified methods to use Java with VRML. One method is to use Script nodes. There is a normative Java Script⁵ node implementation annex to VRML specification. It defines required implementation for Java functionality from Script Nodes. The External Authoring Interface (EAI) is the other way to use Java with VRML. It is not required but several browsers have implemented it. Both the internal Java Script Node and the External Authoring Interface allow programmers to control the nodes in the scene graph from within Java. The choice between them is largely down to the taste of the programmer, using the script node for behaviours purely within the world and the external interface for behaviours linking outside. Within a WWW browser the EAI provides simple access from a Java applet on the same page as the VRML browser using Live Connect, i.e. the EAI allows the user to control the contents of a VRML browser window embedded in a web page from a Java applet on the same page. It does this with a browser plugin interface that allows embedded objects on web pages to communicate with each other. We use internal Script nodes in CONFUCIUS because it does not focus on user interaction when it tells a story and Java Script nodes are enough for creating and modifying the story world dynamically.

For the VRML browser, we have compared and analysed the most common browsers, Blaxxun Contact (Blaxxun Contact 2001), Cosmos player (Cosmos player 2001) and

⁵ Here Java Script is not Javascript, but script in the Script node of VRML written in the Java™ language. Javascript (vrmlscript—a subset of Javascript) and Java are the most popular languages used in the VRML Script node.

Parallelgraphics' Cortona (Parallelgraphics 2001). Blaxxun Contact does not have support for Java in the Script node; and Cosmo complained about missing classes when we compiled our testing Java code; only Cortona works fine for our testing code. Hence, Parallelgraphics' Cortona 4.0 is used as our VRML browser and its VRML packages of Java to compile our Java classes in CONFUCIUS.

When we proceed to design sophisticated heuristics for character actions, changing facial expressions, say, or simulate lip movements when they are speaking, we are limited only by the processing speed of Java scripts. Advanced artificial intelligence algorithms might be too much for a browser to handle while it's trying to keep up a minimum frame rate.

7.6.4 Applying narrative montage to virtual environments

We have investigated the narrative montage in Chapter 2, section 2.3.6, here we sketch possibilities for their application in VRML and discuss how to apply these montage techniques to intelligent storytelling.

1. *Cut* is easily implemented in VRML, by just using guided Viewpoint with the field `jump` set to `TRUE` (see Figure 7.21) or the `replaceWorld (MFNode nodes)` function in the VRML browser Application Programming Interface (API).
2. *Lap dissolve*. Unlike with 2D media, it is not feasible to implement *lap dissolve* in a 3D VRML world. We have to use *cut* to substitute for *lap dissolve* in CONFUCIUS.
3. *Pan shots* can be achieved in VRML by guided Viewpoint with the field `jump` set to `FALSE`.
4. *Strange camera angles* can be defined by setting the fields `position` and `orientation` in Viewpoint node of VRML.
5. *Cross-cutting (parallel editing)* may be achieved by some VRML browser API functions such as `loadURL(MFString url, MFString parameter)` and `replaceWorld (MFNode nodes)` to switch between two VRML files.
6. *Flashback* can be implemented by `loadURL(MFString url, MFString parameter)` and using `Script` with `TimeSensor` to control returning to the present.
7. *Subliminal shots*. The implementation of subliminal shot is the same as flashback except the value of `cycleTime` of `TimeSensor` is shorter than that of flashback.
8. *Visual rhythm and distortion of natural rhythms* can be controlled through proper setting of `TimeSensors`.
9. *Zoom-freeze* may be carried out by guided Viewpoint binding. Because a VRML world always expects user interaction, if the viewer didn't have any action, the scene just freezes there. We may guide a `viewpoint` to a closer position and proper orientation to the object that we want to zoom in.

10. *Iris* can be accomplished by simply adding a physical pipe-liked object near the active `viewpoint` position, pointing it to the emphasized object, to allow the viewer to see through.
11. *Imagery* can be done by creating a second window where the allusion is presented. This new channel is impossible for conventional film-editing and allows direct communication of the character's thoughts.
12. *Voiceover*. CONFUCIUS' default setting is no narrator, and it supports *omniscient external narrator* through a speak-aside presentation agent Merlin and *character as narrator* through synthesising speech for cognition verbs without lip synchronisation.

As enumerated above, most of the montage techniques can be formalized and simulated by software. Since the current version of CONFUCIUS only deals with single sentences, we have implemented *cut* and *voiceover* in its storytelling.

7.7 Text-to-speech

For Text-To-Speech (TTS) software, choices can be made from current industry standard speech middleware, such as SAPI (Speech Applications Programmers Interface) from Microsoft (SAPI 2002), JSAPI (Java Speech API) from Sun (JSAPI 2002), and Festival (Taylor et al. 1998). The selection of a TTS engine should take operating system platforms into account since some of them only work on specific platforms.

There are two ways to synchronise a character's lip animation with his speech through a TTS engine: time-driven and event driven. The time-driven method is to obtain estimates of word and phoneme timings and construct an animation schedule prior to execution. The event-driven method is to assume the availability of real-time events from the TTS engine-generated while the TTS is producing audio, and compile a series of event-triggered rules to govern the generation of the animation. The first approach allows us to choose a TTS engine more freely, whereas the second must be used with TTS engines supporting event-driven timing, such as Microsoft Whistler (Huang et al. 1996).

FreeTTS is used for speech synthesis within CONFUCIUS since it is written entirely in the Java programming language, supports Java Speech API, and fits well to our development environment. FreeTTS is derived from the Festival Speech Synthesis System (Taylor et al. 1998) and the FestVox project (FestVox 2003). The algorithm of CONFUCIUS' TTS module interfacing with FreeTTS is described as follows:

1. Find a pair of quotation marks.
2. In the context, looking for a *verb of speaking or manner of speaking* in the class 2.1 in Figure 5.4 or a *cognition verb* (e.g. "decide", "doubt", "think", "want") in the class 2.2.2.4 in Figure 5.3.
3. Find the speaker, gender, age, give it an ID (name) for a specific voice, and annotate the text input to speech synthesisers using Java Speech API Markup Language (JSML).

4. Output synthesised speech to a .wav file, which is picked up by the animation engine and used in an inline object of the generated VRML file.

Here are three examples taken from *Alice in Wonderland*:

- (1) “You ought to have finished,” *said* the King. “When did you begin?”
- (2) “I beg pardon, your Majesty,” he *began*, “for bringing these in. But I hadn’t quite finished my tea when I was sent for.”
- (3) “And that’s the jury-box,” *thought* Alice, “and those twelve creatures.”

The first example has a typical verb of speaking “say” indicating an utterance. In Example 2 the verb “begin” has a word sense “begin to speak or say”, which belongs to the class 2.1 *verb of speaking* in Figure 5.4. The verb “think” in Example 3 is a *cognition verb* in the class 2.2.2.4 in Figure 5.3. Covering cognition verbs ensures that the speech modality takes charge of what a character is thinking. It is common in temporal visual arts like movies or cartoons that a character actually speaks out what (s)he is thinking. Visual modality is used to differentiate between verbs of speaking and verbs of thinking (i.e. cognition verbs). Though both contents are expressed by speech, cognition verbs are not accompanied by lip movements.

7.8 Using presentation agents to model the narrator

The animation of interface agents includes their body poses, gestures, facial expression, and lip synchronisation. We chose Microsoft Agent (Microsoft Agent 2002) Merlin as our story narrator from existing 3D interface agents such as Baldi in the CSLU toolkit (CSLU 2001) and Cassell et al.’s (2000, 2001) BEAT because Baldi is a talking head without any body movements, and BEAT can not be easily interfaced to other systems. Microsoft Agent provides a set of programmable software services that supports the presentation of interactive animated characters in Windows. It enables developers to incorporate conversational interfaces, which leverage natural aspects of human social communication. In addition to mouse and keyboard input, Microsoft Agent includes support for speech recognition so applications can respond to voice commands. Characters can respond using synthesised speech, recorded audio, and/or text in a cartoon word balloon. One advantage of agent characters designed by Microsoft Agent is they provide high-level of a character’s movements that often found in the performance arts, like blink, look up, look down, and walk.

Table 7.1 shows a sample character — Merlin’s details. There are not only fundamental locomotions but also emotions such as *confused*, *pleased*, and *sad* in the 59 defined animations of Merlin. This leads to another important requirement for our characters: emotion. How to draw on emotions from the story input? Fortunately, most children’s stories explicitly describe characters’ emotion. As shown in the story *The Tortoise and the Hare* in Figure 7.25, words in italics express the emotions of the tortoise and the hare. What the natural language processing module needs to do is to classify these words into fixed categories (e.g. happy, sad, angry) which can be performed by animated agents. In performance art a character’s emotion is

revealed only by what he says and does (body poses and facial expressions). What he says is determined by the script, and what he does, besides those actions specified in the script, requires the actor to extract information from the dialogue and the situation, and present it to the audience. The current version of CONFUCIUS does not have the function of emotion extraction. This can be one of the directions for future work.


	Name:	Merlin
	File Name:	Merlin.acs
	File Size:	455635
	Original Width:	128
	Original Height:	128
	Speed:	130
	59 Animations:	Acknowledge Alert Announce Blink Confused Congratulate DontRecognize Explain GestureDown (Left/Right/Up) GetAttention Greet Hearing Hide Idle Listening LookDown (Left/Right/Up) MoveDown (Left/Right/Up) Pleased Read RestPose Sad Search Searching Surprised Think Wave Write ...
	0 Looping Animations:	

Table 7.1: Character Merlin specification

All presentation agents including Microsoft Agent have limits in allowing users to specify a large variety of actions for action-rich stories/scripts that are much more popular than dialogue-rich ones, especially for films. Typically, the actions that most presentation agents can perform comprise the following types:

- High-level presentation acts. This group of actions includes pointing gestures, speaking and the expression of emotions, e.g. `Merlin.Explain`, `Merlin.GestureLeft`, `Merlin.Pleased`.
- Idle-time acts. To achieve a lifelike and natural behaviour of the agent, it even *stays alive* in an idle phase. Typical acts to span pauses are blinking, breathing, and thumb twiddling, e.g.

periodic blinking, mouth and eye movements, and eye tracking when the agent is not speaking.

- Reactive behaviours. In any interactive system, the agent should be able to react to some user interactions immediately. For example, if the user moves the window to which the agent is currently pointing, the consistency of the pointing gesture has to be restored as soon as possible, by a prolongation of the pointing stick or by moving the agent to a new position.
- Basic postures/acts. An action is basic if it cannot be decomposed into less complex sub-actions. Technically speaking, a basic posture corresponds either to a single frame of the agent or to an uninterrupted sequence of several frames, e.g. step or turn.

Then one day, the *irate* tortoise answered back: "Who do you think you are? There's no denying you're swift, but even you can be beaten!" The hare squealed with *laughter*.

.....

Annoyed by such bragging, the tortoise accepted the challenge.

.....

Smiling at the thought of the look on the tortoise's face when it saw the hare speed by, he fell fast asleep and was soon snoring *happily*.

.....

Tired and in disgrace, he slumped down beside the tortoise who was silently *smiling* at him.

Figure 7.25: *The Tortoise and the Hare* from Aesop's Fables

These actions are more than enough for the narrator and minor characters of CONFUCIUS. Figure 7.26 shows a snapshot of Merlin telling a story which is realised by a two-frame HTML file shown in Figure 7.27. In Figure 7.27B, the agent frame controls Merlin's behaviours and speech, and the `vrml` frame shows the virtual world presenting input sentences. Figure 7.27A gives the Javascript in the Merlin.html file.

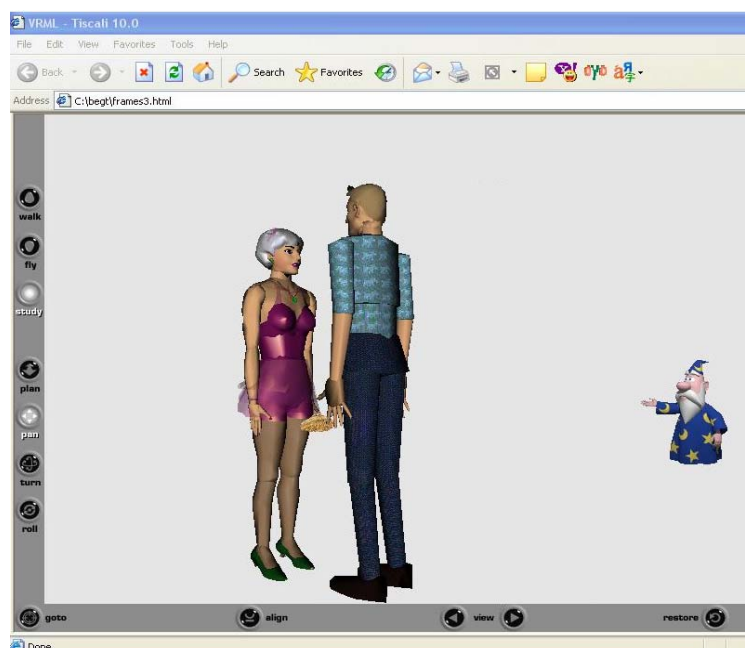


Figure 7.26: Narrator Merlin speaks alongside a story

```

<html>
<head><title>MS agent: Merlin</title>
</head>

<body language=Javascript onload=OnLoad(>
  <OBJECT classid=CLSID:D45FD31B-5C6E-11D1-9EC1-00C04FD7081F
codeBase="#VERSION=2,0,0,0" height=0 id=AgentControl width=0>
  </OBJECT>
  <OBJECT classid=CLSID:B8F2846E-CE36-11D0-AC83-00C04FD97575
codeBase=#VERSION=6,0,0,0 height=0 id=TruVoice width=0>
  </OBJECT>

  <SCRIPT language=Javascript>
var Merlin;          // a global variable to hold the character object
function OnLoad() {
  AgentControl.Characters.Load("Merlin", "merlin.acs");
  Merlin = AgentControl.Characters.Character("Merlin");
  //Merlin.LanguageID = 0x0409; // needed under come conditions
  Merlin.Show();
  Merlin.Play("Greet");
  Merlin.Play("GestureRight");
  Merlin.Speak("Bob gave Jane a loaf of bread.");
}
  </SCRIPT>
</body>
</html>

```

A. Merlin.html

```

<HTML>
  <HEAD>
    <TITLE>VRML</TITLE>
  </HEAD>

  <FRAMESET COLS="1,*">
    <FRAME NAME="agent" SRC="Merlin.html">
    <FRAME NAME="vrm" SRC="vrm\h-anim\bob_give.wrl">
  </FRAMESET>

</HTML>

```

B. A two-frame HTML file

Figure 7.27: The HTML code of Figure 7.26

7.9 CONFUCIUS worked examples

Below are two examples showing how CONFUCIUS runs from single sentence input to 3D animation and listing outputs of each step in NLP and animation generation.

Example 1:

Input: John put a cup on the table.

The input is given in a text file in a designated folder. A Java program `FDG_Parse.java` opens an HTTP connection, sends the content of the input text file to the URL of the Connexor parser's CGI-Bin script, receives the data sent back from the URL and writes it to another file in the same folder. Figure 7.28 is the output of the Connexor Machinese

syntax parser for the input. The data received from Connexor is in HTML format, and hence the output of `FDG_Parse.java` is an HTML file.

Analysis of Machine Syntax for English:

#	Text	Baseform	Syntactic relation	Syntax and morphology
1	John	john	subj:>2	@SUBJ %NH N NOM SG
2	put	put	main:>0	@+FMAINV %VA V PAST
3	a	a	det:>4	@DN> %>N DET SG
4	cup	cup	obj:>2	@OBJ %NH N NOM SG
5	on	on	loc:>2	@ADVL %EH PREP
6	the	the	det:>7	@DN> %>N DET
7	table	table	pcomp:>5	@<P %NH N NOM SG
8	.	.		
9	<s>	<s>		

Figure 7.28: Connexor parser output of “John put a cup on the table.”

Another Java program `SemanticMarking.java` then removes all the HTML tags and unnecessary content such as the title and table headings, and adds semantic marks for each word (i.e. line) based on the proper noun list, popular name list, and WordNet, as described in section 7.4.2. Figure 7.29 shows the output of `SemanticMarking.java`. There is only one semantic feature being added for this example, which is the italics text in the first line — `PERS_M`, indicating “John” is a male person.

```

1 John john subj:>2 @SUBJ %NH N PERS_M NOM SG
2 put put main:>0 @+FMAINV %VA V PAST
3 a a det:>4 @DN> %>N DET SG
4 cup cup obj:>2 @OBJ %NH N NOM SG
5 on on loc:>2 @ADVL %EH PREP
6 the the det:>7 @DN> %>N DET
7 table table pcomp:>5 @<P %NH N NOM SG
8 . .
9 <s> <s>

```

Figure 7.29: After removing HTML format and adding semantic marking

Next, the `SemanticAnalysis.java` program processes the text file shown in Figure 7.29 using the WSD algorithm described in section 7.4.3 and outputs the LVSR shown in Figure 7.30. It identifies each semantic role of the verb and decides the verb class in the visual semantic based verb ontology discussed in Chapter 5, Figure 5.3. The verb “put” belongs to the class 2.2.1.3.2 with one human role and two object roles, one of which is the theme “cup” and the other is a goal “table”.

Following WSD, another program `VerbInterpret.java` replaces the main verb in the LVSR in Figure 7.30 to a base-level verb that was discussed in Chapter 5, section 5.2.3, based on WordNet. Figure 7.31 shows the result: the verb “put” is replaced by “place”.

```
[EVENT put, class 2.2.1.3.2
agent:(john{male})
theme:(cup)
spatial/time:(on(table))
]
```

Figure 7.30: After semantic analysis and WSD

```
[EVENT place, class 2.2.1.3.2
agent:(john{male})
theme:(cup)
spatial/time:(on(table))
]
```

Figure 7.31: After verb replacement

Since there are neither verbs of speaking nor verbs of cognition in the sentence, and the verb belongs to a visually presentable verb class, the animation engine finally takes over and process the LVSR in Figure 7.31 as described in section 7.6. It finds that the event “place” is available in the animation library, uses the male H-Anim model for John, and adds an external prototype to the H-Anim file, as shown in Figure 7.22, to link the virtual human and the animation. It then finds the models of a cup and a table, the spatial site of “on” in the table’s VRML file and the spatial site of `l_metacarpal_pha2` on the virtual human’s hand (Chapter 6, section 6.2.7) for placing the cup and the `x_front` site of the human for placing the table. Finally, a viewpoint (camera) is chosen as the default viewpoint by the camera control module. Figure 7.32 shows the VRML file generated by the animation engine, displaying in an Internet Explorer browser with the Cortona VRML plugin.

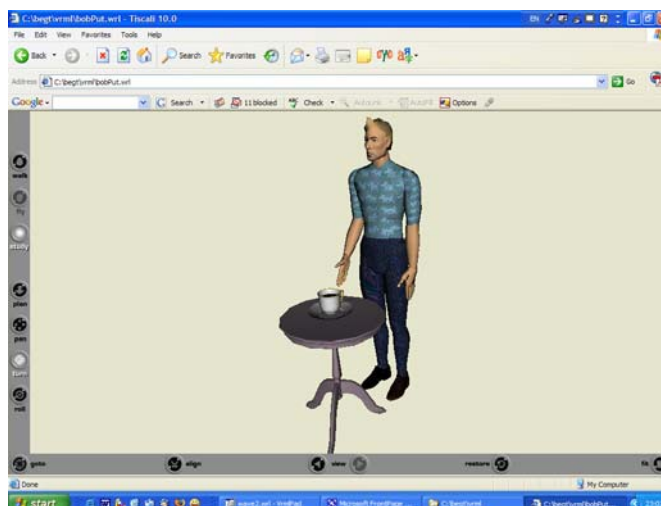


Figure 7.32: The output animation of “John put a cup on the table.”

Example 2:

Input: John left the gym.

Figures 7.33-7.37 show the output of each step for the sentence “John left the gym”. Note in Figure 7.35, after disambiguation from various word senses of “leave” as discussed in

Chapter 5, section 5.4, the semantic analyser decides that the verb “leave” belongs to the class 2.2.1.2.1.4 with one human role and one object role (i.e. the goal) and that it is a combined action involving both upper and lower limb movement. Then the `VerbInterpret.java` program replaces “leave” with “walk” since it is a human moving action.

Analysis of Machine Syntax for English:

#	Text	Baseform	Syntactic relation	Syntax and morphology
1	John	john	subj:>2	@SUBJ %NH N NOM SG
2	left	leave	main:>0	@+FMAINV %VA V PAST
3	the	the	det:>4	@DN> %>N DET
4	gym	gym	obj:>2	@OBJ %NH N NOM SG
5	.	.		
6	<s>	<s>		

Figure 7.33: Connexor parser output of “John left the gym.”

```

1 John john subj:>2 @SUBJ %NH N PERS_M NOM SG
2 left leave main:>0 @+FMAINV %VA V PAST
3 the the det:>4 @DN> %>N DET
4 gym gym obj:>2 @OBJ %NH N NOM SG
5 . .
6 <s> <s>

```

Figure 7.34: After removing HTML format and adding semantic marking

```

[EVENT leave, class 2.2.1.2.1.4
 agent:(john{male})
 spatial/time:(from(gym))
]

```

Figure 7.35: After semantic analysis and WSD

```

[EVENT walk, class 2.2.1.2.1.4
 agent:(john{male})
 spatial/time:(from(gym))
]

```

Figure 7.36: After verb replacement

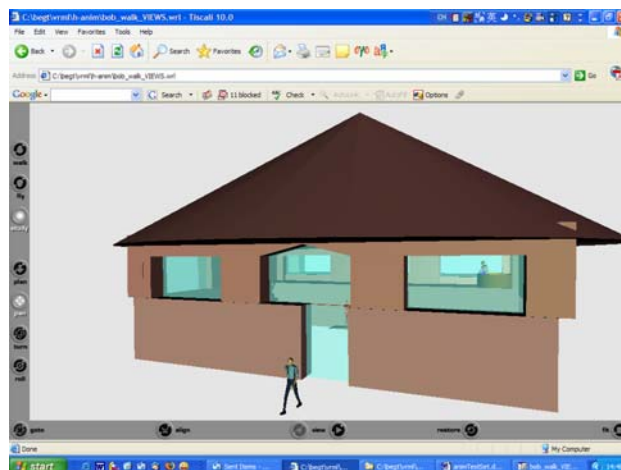


Figure 7.37: The output animation of “John left the gym.”

Example 3:

Input: The waiter came to me: “Can I help you, Sir?”

This example involves speech modality, lip synchronisation, and automatic camera placement for the avatar’s point-of-view (i.e. the first person “me” in the input sentence). Figure 7.38 shows the output of Connexor syntax parser for this sentence. The text in quotation marks is sent to TTS engine to generate a .wav file in a specified directory. The remaining part (with HTML tags removed and semantic marking added) as in Figure 7.39 is then processed by the semantic analyser, and the output of this step is shown in Figure 7.40. The first person pronoun is replaced by a camera ViewPoint in VRML. Next, the `VerbInterpret.java` program replaces “come” with “walk” since it is a human moving action (Figure 7.41).

Analysis of Machine Syntax for English:

#	Text	Baseform	Syntactic relation	Syntax and morphology
1	The	the	det:>2	@DN> %>N DET
2	waiter	waiter	subj:>3	@SUBJ %NH N NOM SG
3	came	come	main:>0	@+FMAINV %VA V PAST
4	to	to	dat:>3	@ADVL %EH PREP
5	me	i	pcomp:>4	@<P %NH PRON PERS ACC SG1
6	:	:		
7	`	'		
8	Can	can	v-ch:>10	@+FAUXV %AUX V AUXMOD
9	I	i	subj:>8	@SUBJ %NH PRON PERS NOM SG1
10	help	help		@-FMAINV %VA V INF
11	you	you	obj:>10	@OBJ %NH PRON PERS ACC
12	,	,		
13	Sir	sir	voc:>10	@VOC %NH N NOM SG
14	?	?		
15	'	'		
16	<s>	<s>		

Figure 7.38: Connexor parser output of “The waiter came to me: ‘Can I help you, Sir?’”

```

1 The the det:>2 @DN> %>N DET
2 waiter waiter subj:>3 @SUBJ %NH N PERS_M NOM SG
3 came come main:>0 @+FMAINV %VA V PAST
4 to to dat:>3 @ADVL %EH PREP
5 me i pcomp:>4 @<P %NH PRON PERS ACC SG1
6 : :
7 <s> <s>

```

Figure 7.39: After removing HTML format and adding semantic marking

```

[EVENT come, class 2.2.1.2.1.4
  agent:(waiter{male})
  spatial/time:(to(CAMERA))
]

```

Figure 7.40: After semantic analysis and WSD

```
[EVENT walk, class 2.2.1.2.1.4
  agent:(waiter{male})
  spatial/time:(to(CAMERA))
]
```

Figure 7.41: After verb replacement

Finally, the animation engine applies the walking animation to a male virtual human, embeds the .wav file of the speech, and synchronises the speaker’s lip movement as described in section 6.2.6, and the camera control module places the default camera at the “front view” viewpoint of the virtual human as discussed in section 6.5. The output animation of the generated VRML file is shown in Figure 7.42.

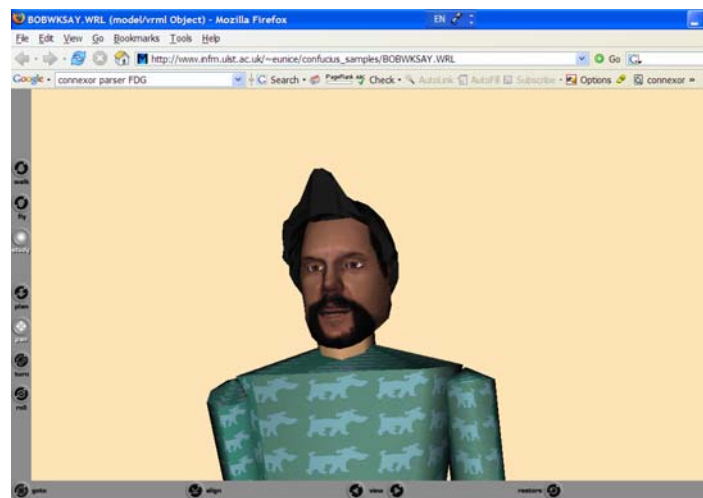


Figure 7.42: The output animation of “The waiter came to me: ‘Can I help you, Sir?’”

7.10 Summary

This chapter described the architecture of CONFUCIUS and explained solutions with respect to the implementation of the knowledge base, NLP, media allocation, 3D animation generation, TTS, and the story narrator. The knowledge base is composed of language, visual knowledge, and cinematic principles. The semantic part of language knowledge in the knowledge base uses LVSR, along with WordNet and the LCS database, to analyse the semantics of language input. The knowledge base provides a solid foundation for integration of the language modality with visual and auditory modalities in intelligent multimodal storytelling. The main NLP solutions addressed were WSD, action representation, and applying lexical knowledge to semantic analysis. The chapter has also discussed object modelling, human animation, collision detection, and application of narrative montage in virtual environments. Examples were given to describe how CONFUCIUS runs from single sentences to 3D animation. Currently, CONFUCIUS is able to visualise single sentences which contain action verbs with visual valency of up to three such as “John left the gym” and “Nancy gave John a loaf of bread”.

As CONFUCIUS is composed of several modules with different tasks to accomplish, an important aspect of the implementation is that it built up an overall framework of intelligent

multimodal storytelling, which makes use of state-of-the-art techniques of natural language processing, text-to-speech, 3D modelling and animation, and integrates multiple diverse components within a complete storytelling framework. The next chapter discusses the subjective and objective evaluation methods and results.

Chapter 8

Evaluation

The evaluation of CONFUCIUS can be organized based on evaluation techniques used. Evaluation techniques in general fall into two categories: subjective techniques, requiring the participation of human subjects, and objective techniques, which do not. Both subjective and objective techniques can be used to provide either global or local evaluation of language visualisation. Objective techniques are further divided into two categories: *diagnostic evaluation* and *adequacy evaluation* (Hirschman and Thompson 1995). Diagnostic evaluation refers to the production of a system performance profile with respect to the possible inputs or test suites. It needs a large amount of data to determine the coverage of the system and fix any faults if found. Adequacy evaluation determines the fitness of a system for a given task and evaluates whether the system does what is required and how satisfactorily the task is carried out.

In this chapter, we design some measures for diagnostic and adequacy evaluations to evaluate the performance of CONFUCIUS on the specific text-to-animation task and discuss the results and findings of the evaluation. The diagnostic evaluation is based on applicable and objective test suites, while the adequacy evaluation includes testing with sets of typical sentences chosen from the specific domain of children's stories. Since working examples of other systems are limited, the evaluation does not include *relative evaluation*, i.e. comparison with other systems.

8.1 Subjective evaluation of animation generation

We evaluate CONFUCIUS' language visualisation in a questionnaire experiment which was conducted on a laptop computer where subjects viewed the animation and completed an HTML questionnaire.

8.1.1 Subjects

Twelve untrained subjects participated in this experiment, 8 females and 4 males, none of which were computer scientists. All subjects participated on a voluntary basis. The mean age of the participants was 32.7 with a range from 24 to 45 years. All of the subjects reported that they had had some Web experience and only one reported some experience with virtual chat rooms.

After an introduction to CONFUCIUS and a skill building session to teach the subject how to control the viewpoint to navigate the virtual world, the subject was given several 3D animations to play with and to become familiar with the Cortona VRML browser

controls. During the experiment, asking for help on viewpoint control was allowed if the subjects couldn't get a specific view they wanted.

8.1.2 Questionnaires

After exploring each 3D animation, subjects were asked to complete a questionnaire (see Appendix G) either rating (on a 5-point scale) agreement for the animation at the word level and sentence level, or selecting the closest text from four candidates to describe the animation, which we call *comprehension measures*, i.e. multiple choice tests at both word and sentence level. Figure 8.1 shows a screenshot of the evaluation questionnaire which is an HTML page with snapshots linking to 3D animations generated by CONFUCIUS.

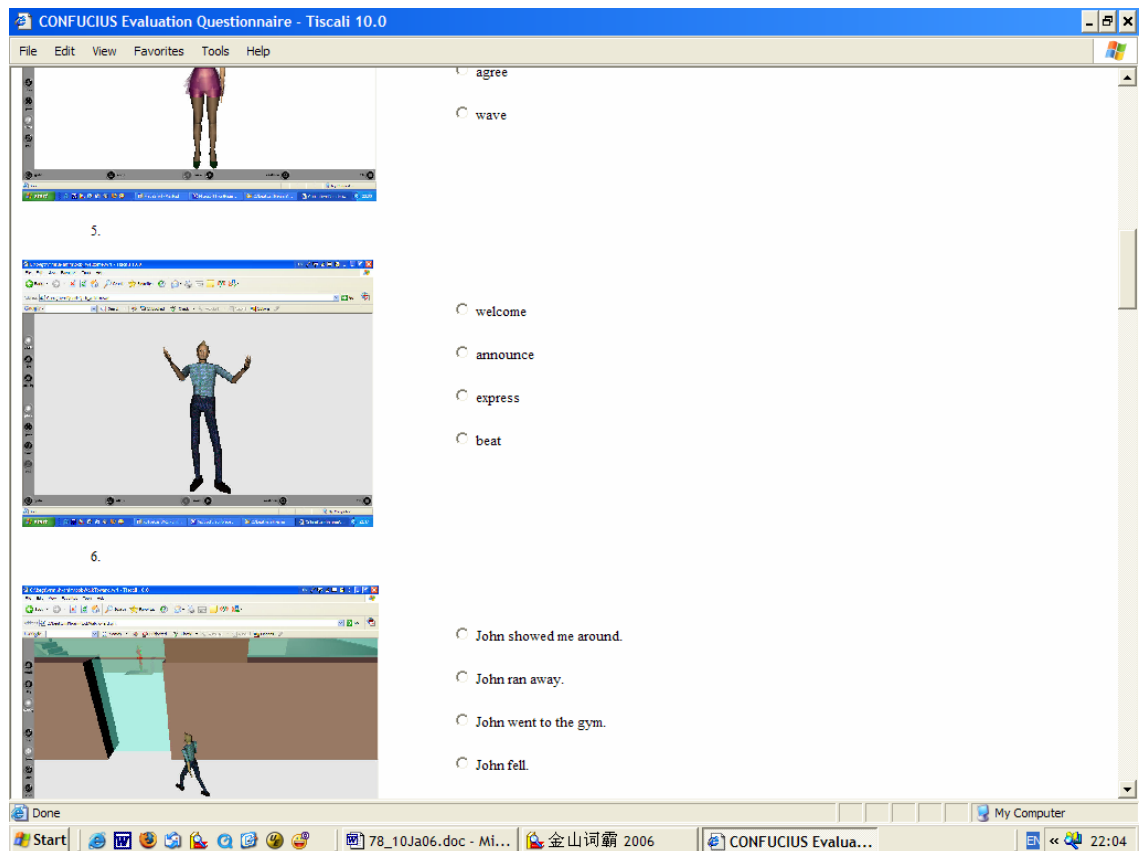
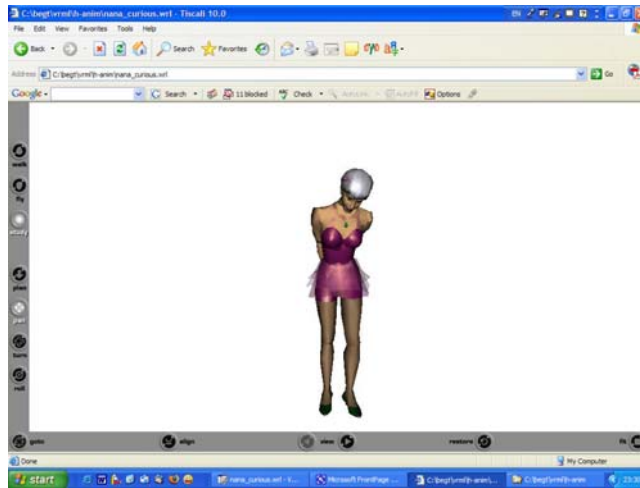


Figure 8.1: A screenshot of the evaluation questionnaire

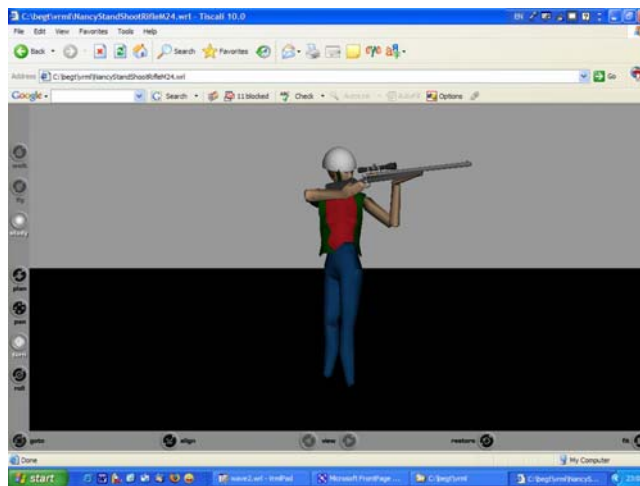
The evaluation questionnaire contains two parts: agreement rating and comprehension measurement. Both parts include two levels of measures: word level and sentence level measures. In the agreement score part, participants are given output animation and a word or a sentence and asked to give a subjective score of the animation quality rating between 1-5 (5: Excellent, 4: Good, 3: Average, 2: Poor, 1: Terrible), indicating how well the animation expresses the word or sentence given (see Figure 8.2 and 8.3). In the comprehension measure part, participants are given an output animation and asked to choose the closest word or sentence from four alternatives to describe this animation (see Figure 8.4 and 8.5).



Please rate the animation indicating how well the animation expresses the word “curious”.

5. Excellent
4. Good
3. Average
2. Poor
1. Terrible

Figure 8.2: An example of word-level agreement rating



Please rate the animation indicating how well the animation expresses the sentence “Jane shot the bird.”

5. Excellent
4. Good
3. Average
2. Poor
1. Terrible

Figure 8.3: An example of sentence-level agreement rating



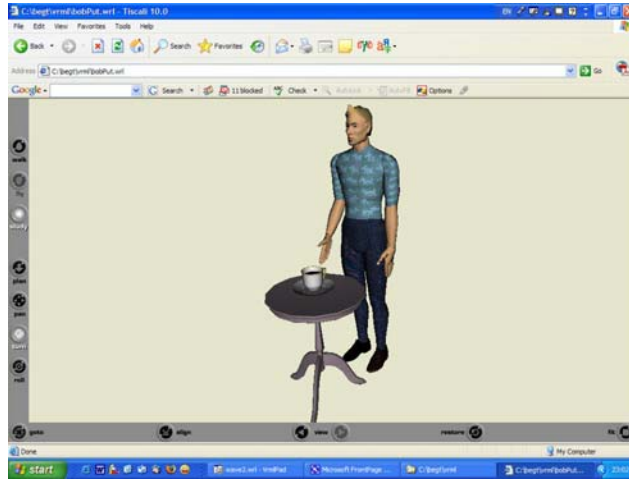
Please choose the closest word to describe the animation.

- A. show
- B. tell
- C. give
- D. sell

Figure 8.4: An example of word-level comprehension measurement

Eighteen animations were given for evaluation. Nine are for word level measures and nine for sentence level measures. For language animation applications, discriminating between *modifiers* and *heads*, visually observable words and visually unobservable words is important since modifiers and visually unobservable words usually have less value in language

visualisation. Heads are carriers of modifiers and hence are more important than modifiers in visualisation, though modifiers may play a more crucial role in identification. The word level measures can be complemented by sentence level measures to evaluate language visualisation applications. The evaluation of the animation quality is accomplished by analyzing participants' replies to the questionnaire.



Please choose the closest sentence to describe the animation.

- A. John picked up the cup.
- B. John put a cup on the table.
- C. John served tea.
- D. John chose the cup.

Figure 8.5: An example of sentence-level comprehension measurement

Our analyses were qualitative in nature, and participants' impressions were gleaned from a survey conducted at the end of the questionnaire. Subjects were given the opportunity to comment freely on each animation and how they thought it could be improved, from which we identify crucial issues that could guide us in future directions. Though the subjective judgement is not perfectly correct due not only to human factors but also to limited evaluation tasks provided and to the inherent vagueness and ambiguity of input text, it is useful to evaluate the quality of computer-generated animations.

8.1.3 How to choose candidate words?

The three non-target candidate words are chosen according to the visual semantic based verb ontology that was discussed in Chapter 5, section 5.2.4. Therefore, the criteria for choosing verb candidates include end effector, visual valency, Level-Of-Detail (LOD), and Levin's classes. Verb candidates of the comprehension measurement are usually chosen from different Levin classes and with same end effectors. The generated animation should be adequate for users to differentiate a specific verb/verb type from other verbs with same end effector(s) and visual valency. It may not be fine-grained enough to differentiate verbs in the same hypernym tree on different LODs (e.g. "stumble"- "walk").

The word level measures in the evaluation questionnaire only include verbs since they are the most important part-of-speech in 3D animation generation. Other part-of-speech measures (noun, adjectives, prepositions and spatial relationships) are integrated in the sentence level measures, e.g. the animation "John put a cup on the table" integrates the preposition "on".

In particular, there is no need to evaluate animation for nouns because the animation quality completely depends on the object modelling which is not a main focus of CONFUCIUS.

8.1.4 Results

Table 8.1 gives results for the comprehension measures in the animations 1-10. The rows represent specific animations on the questionnaire HTML page. The second column is the number of mismatches between the animation and text. The third column is the error rate for each animation, calculated by dividing the number of mismatches by the number of subjects. The total number of mismatches for all the animations is 10, and the mean value of the error rate is 8.33%. The mean error rate is calculated by dividing the total number of errors by the total number of animations evaluated by all subjects. Table 8.2 gives results for agreement rating in animations 11-18. It includes sum of scores for all 12 subjects, mean values, and standard deviations for each animation. The average agreement score is 3.82 (between 1-5).

Animation#	No. of mismatches	Error rate %
1	3	25.00
2	0	0
3	2	16.67
4	0	0
5	2	16.67
6	0	0
7	1	8.33
8	0	0
9	1	8.33
10	1	8.33
Total	10	
Mean		8.33

Table 8.1: Results of the comprehension measures for animations 1-10

Animation#	Sum of scores	Mean score	Standard deviation
11	55	4.6	0.55
12	36	3.0	0.71
13	41	3.4	0.55
14	43	3.6	1.14
15	46	3.8	1.10
16	45	3.8	0.48
17	58	4.8	0.45
18	43	3.6	0.53
Total	367		
Mean		3.82	

Table 8.2: Subjective agreement rating results for animations 11-18

Typical comments given by the participants are: “it was interesting” and “useful”. In the interview with the subjects after they completed the evaluation questionnaire, we found that they tended to compare the automated generated animations with computer graphics movies like “Toy story” and “A bug’s life” and games, which were created by professional human artists

and motion capture techniques. This might explain why the agreement score was not higher. For future version improvement, some participants suggested more intelligent camera control which saves time for adjusting the viewpoint, and more realistic facial expressions. The inclusion of changing clothing for characters was also suggested by some subjects.

8.1.5 Comparison with computer games

We also compare CONFUCIUS generated animations with computer games. The games that were studied included action games, on-line roleplaying games and virtual environment chat rooms (activeworlds.com). Typical character animations in action games and VR chat rooms include “walk”, “run”, “turn” (left/right), “turn around” (180 degree), “wave”, “jump”, “spin”, “joy”, “agree”, “blow”, “kiss”, “karate”, “kung-fu”, “kick”, “look at” (eye gaze following the camera/viewer), “fall”, “stand up”, “aim”, “shoot”, “hold (weapon)”, “crawl”, “kneel”. Some games have very good quality character animations and extensive coverage of fighting animations. Since CONFUCIUS is an automatic animation generating system, we do not expect CONFUCIUS to generate more realistic animations than those games which are created by a team of professional computer artists. However, CONFUCIUS provides an XML-based facility for non-professional users to add their own animations and to expand its animation library. This facility makes CONFUCIUS more flexible than action games.

It is difficult to measure how effective CONFUCIUS' language visualisation is. In the interviews subjects commented that CONFUCIUS was easy to use and to extend by adding new objects and animations. However, when starting to simulate more complex situations, it becomes difficult to coherently interconnect all needed props and human behaviours. One solution to minimize this point is to build complex objects which contain more interaction information.

8.2 Diagnostic Evaluations

We use glass-box testing to examine the NLP components of CONFUCIUS separately since we are more concerned with the internal operation of NLP to identify component limitations and deficiencies. These components are syntactic parser, semantic analysis, and anaphora resolution. The measurement tests the coverage of general linguistic phenomena, both syntactic and semantic, as listed in Table 8.3. Diagnostic evaluation uses the test suite given in Appendix H that covers these phenomena to identify limitations, errors and deficiencies, which may be corrected or improved by future work. Some phenomena which are important measurements for language generation, such as *agreement*, are not relevant for the text-to-animation task, while other phenomena like verb types and lexical semantics are vital to text-to-animation.

8.3 Syntactic parsing

We employ a test suite of exemplary input sentences to enumerate most of the elementary linguistic phenomena in the input domain and their most likely and important combinations, and to determine the coverage of CONFUCIUS' grammar, i.e. the FDG that the Connexor Machine parser is based on. We use the test suite of HORATIO (Michiels 1994) with 8 ungrammatical sentences removed. The original HORATIO test suite has 178 sentences. The test suite we use (see Appendix H) includes 170 sentences. These sentences were parsed by the Connexor parser, and the output was judged by two human judges with linguistics background. Since the parser is clearly not intended to account for the disambiguation behaviour of language users and there is variance between different human judges, we regard uncertain cases and inter-judge inconsistencies as ambiguity rather than errors. A parsing is regarded as incorrect only if both judges agree. The result gives an accuracy of 84.7% (i.e. 26 incorrect parses out of 170 sentences). We identify four major types of parser deficiency.

<i>Syntactic phenomena</i>	<i>Semantic phenomena</i>
Types of utterances	Underspecification (vagueness, ambiguity)
Verbs types and forms	Lexical semantics (e.g. polyseme, WSD)
Adjectives and Adverbs	Contextual reasoning
Complementation	Anaphoric reference
Modification	Quantification, quantifier scope
Agreement	Countability
Coordination	Collective/distributive readings
Negation	Nominalizations
Word Order	

Table 8.3: General linguistic phenomena

8.3.1 Subordinate clauses without conjunctions

The Connexor parser is unable to handle subordinate clauses without conjunctions. For example, the dependency tree of “He is sure I will tell them what to read”, as shown in Figure 8.6, is broken, while “He is sure that I will tell them what to read” is parsed correctly.

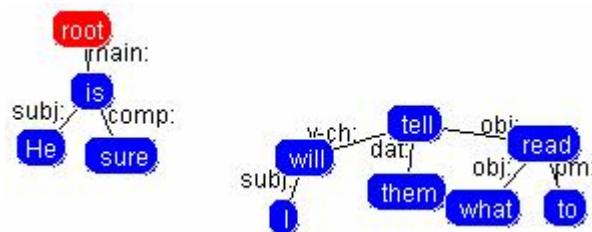


Figure 8.6: Dependency tree of “He is sure I will tell them what to read”

8.3.2 PP attachment

In the example “Have you read the letter to the teacher about the library?” given in Figure 8.7, the PP “to the teacher” can be interpreted as either modifying “read” or modifying “letter”. We regard these as ambiguity. However, in Figure 8.8 “The man reading a book in the library is a teacher”, the PP “in the library” should modify “read” rather than “book” as in the parser’s analysis. We treat these as errors.

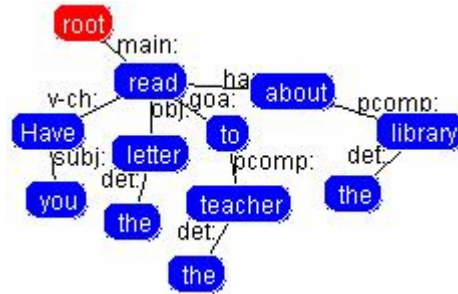


Figure 8.7: Dependency tree of “Have you read the letter to the teacher about the library?”

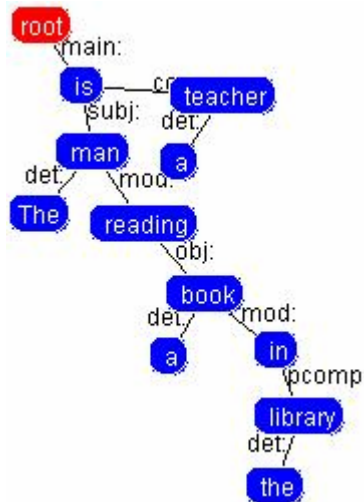


Figure 8.8: Dependency tree of “The man reading a book in the library is a teacher.”

8.3.3 Coordination and ellipsis

Coordinations are complex phenomena in NLP because they break the *normal* pattern of sentence constructions by introducing many kinds of ellipsis as in the following sentences in the test suite:

Mary teaches linguistics and John mathematics.

Mary is and John wants to be in the library.

Mary is in and John wants to be in the library.

Mary went to the library and John to the workshop.

The test has shown the parser is deficient in handling these phenomena. The parser is able to handle coordinations like “John and the students want to put off the workshop” shown in Figure 8.9 but unable to deal with the above four sentences as shown in Figure 8.10.

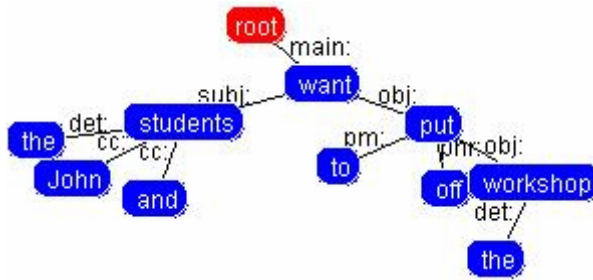


Figure 8.9: Dependency tree of “John and the students want to put off the workshop.”

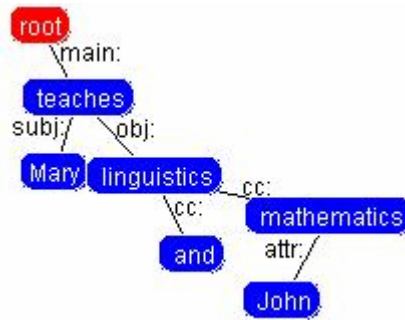
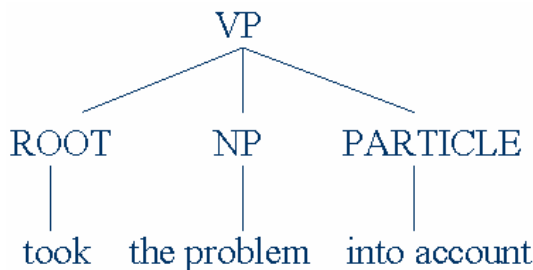


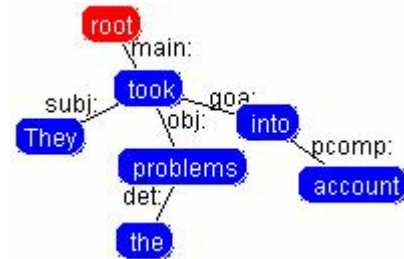
Figure 8.10: Dependency tree of “Mary teaches linguistics and John mathematics.”

8.3.4 Word order and discontinuous constituents

A constituent is called discontinuous if it is interrupted by other constituents, e.g. “take ... into account” in “they took the problem into account”. They can not be captured by a context-free Chomsky grammar. Take for example the typical permutation in (ROOT, NP, PARTICLE) \Leftrightarrow (ROOT, PARTICLE, NP). Figure 8.11 gives an example of (ROOT, NP, PARTICLE), showing its syntactic tree and Connexor’s output of its dependency tree. The Connexor parser can deal with discontinuous constituents interrupted by a simple NP like the case in Figure 8.11. However, Connexor failed to parse discontinuous constituents if the interrupting NP is too complex, as shown in Figure 8.12. In this case the permutation (ROOT, NP, PARTICLE) \Rightarrow (ROOT, PARTICLE, NP) is necessary in order to get the correct dependency tree as shown in Figure 8.13.



A. Syntactic tree of “took the problem into account”



B. Dependency tree of “They took the problems into account.”

Figure 8.11: An example of (ROOT, NP, PARTICLE)

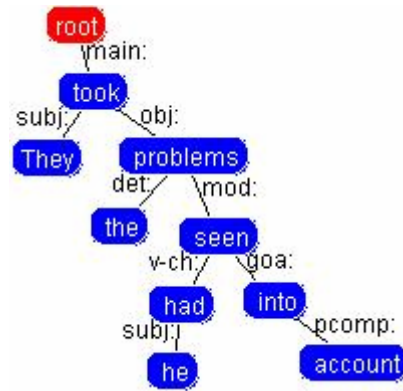


Figure 8.12: “They took the problems he had seen into account.”

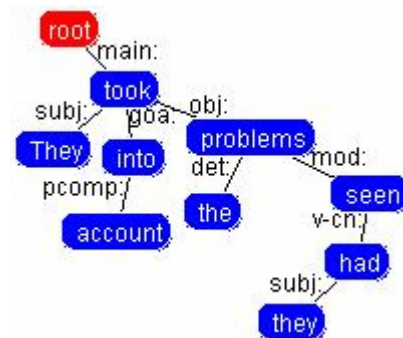


Figure 8.13: “They took into account the problems they had seen.”

8.4 Semantic analysis

To test the accuracy of the semantic analysis component for verb Word Sense Disambiguation (WSD), we use a test set of 40 single sentences (see Appendix J) containing the most frequently used verbs in the British National Corpus (BNC 2004). The sentences containing these verbs are from *Alice in Wonderland*. In order to rule out the effect of the syntactic parser’s performance, we transform some sentences which cannot be parsed correctly by the Connexor parser to those that can be handled by it, e.g. change “It was at the great concert given by the Queen” to “The great concert is given by the Queen” for the verb “give”. CONFUCIUS gives promising results on WSD (70% accuracy, 28 correct word senses out of 40 verbs) with regard to the dataset it is tested on.

Resnik (1997) proposed selectional preferences — a statistical generalization of selectional constraints in WSD to consider probability of a semantic class occurring in a given argument position. He used a training set of 800,000 words of parsed, non-sense tagged text from Brown corpus to train a selectional preference model, and his test set is 200,000 words of parsed (by Penn tree bank folks), and sense-tagged (by WordNet group) Brown corpus text. He took 100 verbs that selected most for objects in the training corpus to calculate accuracy and compared with baselines. Figure 8.14 shows the accuracy of these WSD approaches. One baseline is 26.8% accuracy by randomly assigning a word sense to each occurrence of an ambiguous word. The other baseline is 58.2% accuracy by choosing the most frequent sense

which requires sense-labeled training corpus. Resnik’s statistical approach achieved a 44% accuracy.

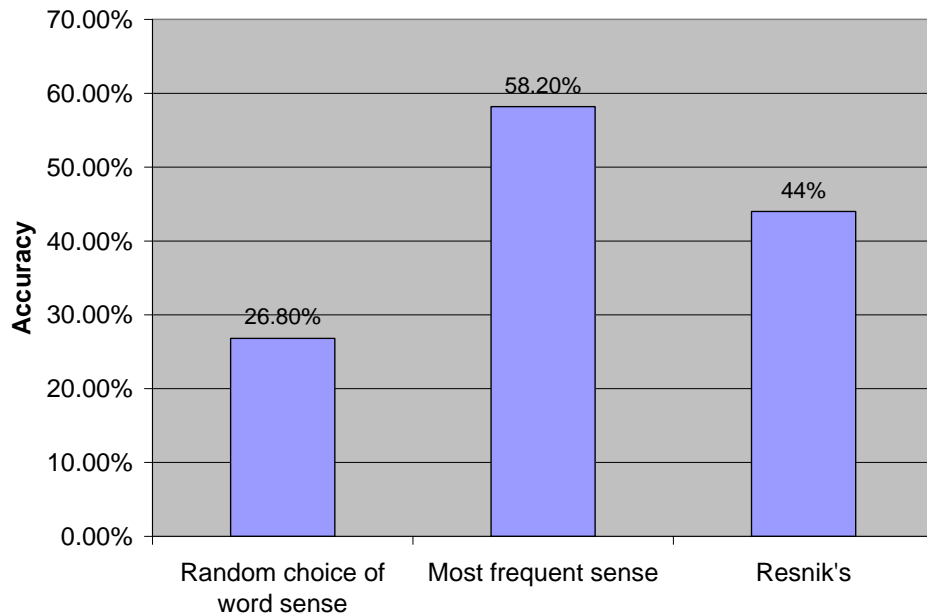


Figure 8.14: Comparing Resnik’s WSD approach with baselines

We identify and localize two major deficiencies of CONFUCIUS’ semantic analyser:

(1) The semantic analyser does not support phrasal verbs such as “go on”, “make up”, “put on”, and “write down”. Since high-frequency verbs are more likely to add a preposition or adverb and compose a meaning different from the original verb, the overall performance will be improved if phrasal verbs are covered. This deficiency can be overcome in future versions of CONFUCIUS by using WordNet’s corresponding facilities, i.e. phrasal verbs have distinct lexical entries in WordNet.

(2) Some failures are due to mistakes in the existing verb lexicons used for semantic analysis. For instance, the semantic analyser is unable to identify the sense of “know” in “I didn't know it was your table”, but it is able to identify it in “I didn’t know *that* it was your table”. This is caused by the verb entry (Figure 8.15), with the sense “be cognizant or aware of a fact or a specific piece of information; possess knowledge or information about”, specifies a subordinate clause with an obligatory conjunction “that” (*italics* in Figure 8.15), which should be optional.

8.5 Anaphora resolution

It is helpful to use adequacy evaluation, i.e. corpus-based evaluation, to test the coverage of linguistic phenomena in the storytelling domain, using typical children’s stories, because the test suite in diagnostic evaluations, like the one we used in section 7.3, sometimes does not reflect the distribution of linguistic phenomena in the actual application domain. Corpus-based evaluation metrics reflect the nature of the task of CONFUCIUS, i.e. language visualisation in the storytelling domain. We use this method to test the accuracy of the anaphora resolution

component (i.e. JavaRAP). A set of 20 paragraphs are randomly selected from *Alice in Wonderland* which includes 105 third person pronouns and reflexive pronouns (see Appendix I). The results show 46 errors out of the 105 references, giving an accuracy of 56.2%, which is close to the accuracy (57.9%) declared by the developer (Qiu et al. 2004).

```
(
  :DEF_WORD "know"
  :CLASS "29.5.b"
  :WN_SENSE (("1.5" 00333362 00333754)
            ("1.6" 00401762 00402210))
  :PROPBANK ("arg0 arg1-PRD(that)")
  :THETA_ROLES ((1 "_exp_prop(that)")
  :LCS (be perc (* thing 2) (at circ (thing 2) (* nil 27))
        (know+ingly 26))
  :VAR_SPEC ((2 (human +)) (27 (thing -) (cform fin)))
)
```

Figure 8.15: An incorrect entry of the verb “know” in the LCS database

8.6 Summary

The evaluations for an intelligent multimodal storytelling application like CONFUCIUS integrate features not only present in evaluation of NLP systems but also in evaluation of graphic systems: the quality of the visualisation and audialisation, e.g. intelligibility, accuracy, fidelity, appropriateness of style; the usability of facilities for expanding graphic library, for controlling input language; the extendibility to new application domains; and cost-benefit comparisons with human animation creation performance. Even with small-scale experiments the practical problems of natural language understanding and animation generation were brought to the surface.

In this chapter we evaluated CONFUCIUS’ animation generation by subjective evaluation, syntactic parsing by test-suite based diagnostic evaluation, and anaphora resolution and semantic analysis by corpus-based adequacy evaluation. CONFUCIUS gives promising results on WSD (70% accuracy) with regard to the dataset it is tested on. The error rate of comprehension measures of animation (8.33%) is quite low. Although the agreement score of CONFUCIUS’ generated animation is 3.82, between “good” (4) and “average” (3), we believe this fact is simply because of subjects’ expectation and comparison with human-made animations. Clearly, there is room for improvement in both the NLP component (e.g. to handle phrasal verbs) and animation engine (e.g. to improve facial expressions and increase clothing varieties).

Automatic conversion of natural language to 3D animation is a new area of NLP, and currently, there are no objective, reliable and publicly acceptable benchmarks and evaluation metrics available for these applications. The evaluation methods used in this chapter can serve as a testbed for text-to-animation applications.

Chapter 9

Conclusion and future work

Virtual storytelling concerns various research areas in NLP and animation generation. In this chapter we conclude by first summarising the research completed in this thesis. We then compare the research with other related work. Finally, potential future directions and applications of the research are explored.

9.1 Summary

In this thesis, we have investigated the problems and solutions for automatic conversion of natural language to 3D animation and reviewed a range of systems, from multimodal storytelling systems, automatic text-to-graphics systems, to embodied agents and virtual humans. Previous work in the areas of language and multimodal semantic representations, temporal relations, computational lexicons, linguistic ontology, virtual human standards, and nonspeech audio was discussed. Lexical Visual Semantic Representation (LVSR) was proposed as a necessary semantic representation between 3D visual information and syntax, and we discussed using LVSR to represent visual semantics of verbs, nouns, adjectives, and prepositions. Temporal information of various semantic relations is encoded in interval relations.

Based on visual and auditory semantics, a verb ontology was proposed. We introduced the notion of *visual valency* and use it as a primary criterion to categorise event verbs for language visualisation. Various lexicon-based approaches used for WSD were discussed. The context and the senses of the ambiguous verb are analysed using hypernymy relations and word frequency information in WordNet and thematic roles in LCS database. We proposed a methodology to extract common sense knowledge on default arguments of action verbs from WordNet to solve the underspecification problem and meet the needs of explicit information required in language visualisation.

We then discussed various issues on computer animation such as virtual human and 3D object animation, animation of facial expressions and lip synchronization, autonomy, collision detection, and automatic camera placement. We proposed the approach of multiple animation channels to blend non-exclusive animations and present overlapping interval relations. In 3D object modelling, object-oriented models were used to encapsulate object-related information to decentralise the control of animation engine. In virtual human animation, we used general-

purpose virtual human characters and their behaviours which follow the H-Anim standard and balance between computational efficiency and accuracy to produce believable human motions.

Having examined the problems and solutions in NLP and 3D animation generation, we implemented an intelligent multimedia storytelling interpretation and presentation system — CONFUCIUS, which automatically generates multimedia presentations from natural language sentences. It employs several temporal media such as 3D animation, speech and nonspeech sound for the presentation of stories. We explained solutions for implementation of media allocation, knowledge base, NLP, 3D animation generation, and the story narrator. The knowledge base is composed of language, visual knowledge, and cinematic principles. The semantic part of language knowledge in the knowledge base uses LVSR, along with WordNet and the LCS database, to analyse the semantics of language input. The main NLP solutions addressed are word sense disambiguation, action representation, and applying lexical knowledge to semantic analysis. We also discussed object modelling, human animation, collision detection, and the application of narrative montage in virtual environments.

Finally, we conducted an evaluation experiment and identified some deficiencies of the current version of CONFUCIUS. We also evaluated the syntactic parsing by test-suite based diagnostic evaluation, and anaphora resolution and semantic analysis by corpus-based adequacy evaluation. CONFUCIUS gives promising results on WSD (70% accuracy) with regard to the dataset it is tested on, and the error rate of comprehension measures of animation (8.33%) is quite low.

Specifically, the main contributions made in the research of this thesis are:

(1) Visual Semantic Representation. Existing multimodal semantic representations within various intelligent multimedia systems represent the general organization of semantic structure for various types of inputs and outputs and are usable at various stages such as media fusion and pragmatic aspects. However, there is a gap between high-level general multimodal semantic representation and lower-level semantic representation that is capable of connecting meanings across modalities. Such a lower-level meaning representation—Lexical Visual Semantic Representation, which connects language modalities to visual modalities, is introduced.

(2) Automatic animation generation. We use an object-oriented method to organize visual/auditory knowledge. 3D object models encapsulate not only their intrinsic visual properties, nonspeech auditory information (auditory icons), but also manipulation hand postures, describing possible interactions with virtual humans. For instance, a gun model includes the visual knowledge of its size, shape, color, the auditory knowledge such as its firing sound, and possible human interaction knowledge, such as where (spatial sites) and how (hand postures) to hold or trigger. This method decentralizes animation/audio control by storing information of object interaction & sound effects in the objects. Object-specific computation is released from the main animation/audio control.

Additionally, we combined precreated and dynamically generated (procedural) animation facilities into a unified mechanism which controls simultaneous animations by multiple animation channels. This approach allows the intelligent storytelling to take advantage of procedural animation effects in the same manner as regular animations, adding an additional level of flexibility and control when animating virtual humans.

(3) Language ontology based on visual semantics. We investigated relations between concepts and multiple modalities, verb/adjective taxonomy and visual semantics, and categorised verbs and adjectives from a visual/audio semantic perspective. In particular, the notion of visual valency and LOD for language visualisation were introduced. The constructed set of verbs and adjectives categories for visual semantics provide a solid framework for further application in natural language visualisation.

(4) A linguistically-based approach concerning lexical semantics of sound emission verbs and audio representable adjectives was introduced. We investigated the relations between concepts, entity properties and the audio modality, and proposed a verb/adjective taxonomy based on audio semantics. The methodology can serve as a framework for researchers in auditory display.

(5) We use an automatic word sense disambiguation approach for mapping verbs to LCS entries using frequency information of WordNet senses, thematic grids and lexical-semantic representations from the LCS database (Dorr and Olsen 1997). This considerably improves the precision of verb sense disambiguation.

(6) CONFUCIUS is an overall framework of intelligent multimedia storytelling, using 3D modelling/animation techniques with natural language understanding technologies to achieve higher level animation generation. It integrates and improves novel theories and techniques in the areas of natural language processing, intelligent multimedia presentation, and 3D graphics.

9.2 Relation to other work

The theoretical and practical context of this research finds its relation to other work, such as existing intelligent storytelling and text-to-graphics systems reviewed in Chapter 2, Jackendoff's LCS (Jackendoff 1990), Levin's verb classes based on semantic/syntactic correlations (Levin 1993) and virtual human simulation (Kallmann and Thalmann 2002). The themes across these works vary from NLP to virtual reality. In this section we compare this research and CONFUCIUS to the different approaches and systems.

9.2.1 Comparison with previous systems

Direct comparisons of performance between various language visualisation, multimodal storytelling, and virtual human systems discussed in Chapter 2 is difficult given the variances in application domain, input/output, and grading criteria. Here we compare CONFUCIUS with

related systems in terms of NLP, I/O modalities, quality of graphics, and automation/intelligence.

Figure 9.1 presents a comparison showing features of various related systems (see Table 2.1). We compare the NLP, multiple modalities, quality of graphics, and automation/intelligence of these systems and rate their overall performance based on information in the literature. Besides the systems listed in the figure, many other practical applications of intelligent multimedia interfaces using virtual agents have been developed in domains such as intelligent tutoring, retrieving information from a large database, real estate presentation, cultural heritage, and car exhibition.

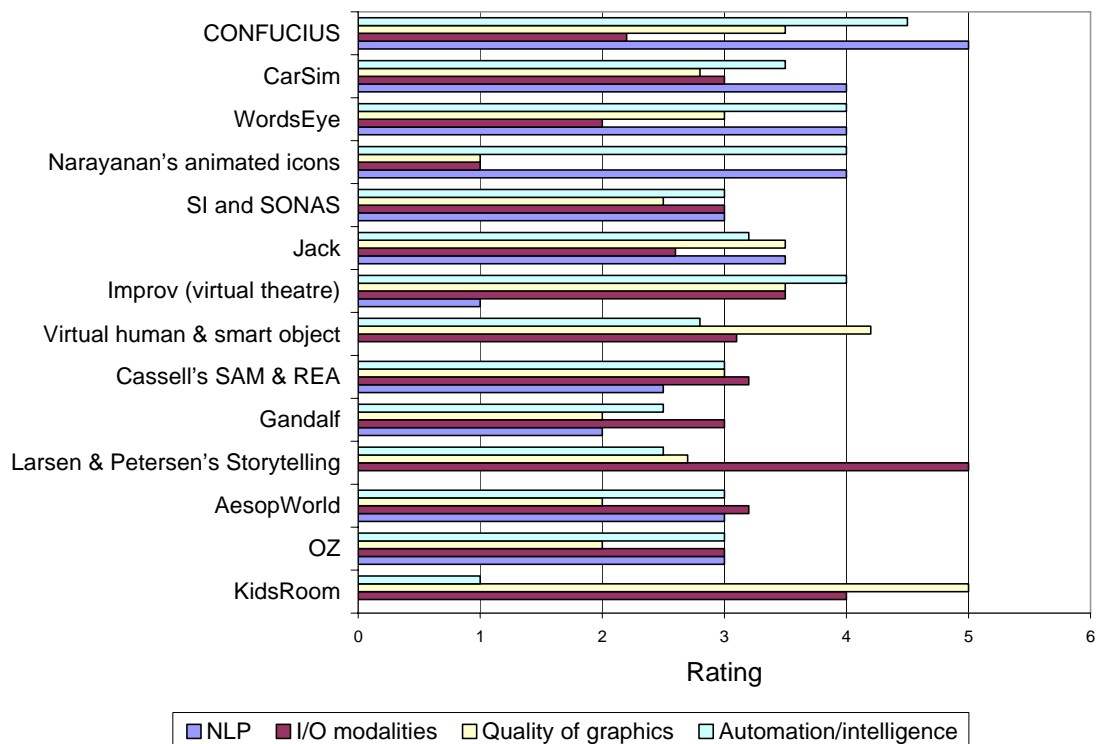


Figure 9.1: Comparison of related systems

The most relevant work in connection with our research is WordsEye (Coyne and Sproat 2001) and CarSim (Dupuy et al. 2001), both of which are text-to-graphics conversion systems. No direct comparison can be carried out because they work in different application domains and evaluation metrics. The goal of this research is identical to that of WordsEye, except that we create 3D animations rather than static 3D scenes from short descriptions. The number of 3D objects WordsEye uses for text-to-graphic conversion, 12000, is much bigger than used in CONFUCIUS', 50, which gives WordsEye more freedom on visualisation tasks. WordsEye integrates NLP resources such as the Collins' dependency parser and WordNet. The interpretation of a narrative is based on an extension of case grammars (semantic frames) and a good deal of inferences about the environment. WordsEye does not address real world stories. The narratives cited as examples resemble imaginary fairy tales, and all the cited texts appear to have been invented by the authors.

CarSim creates 3D animations of car accidents from written texts. CarSim interprets the texts using information extraction techniques. CarSim has been applied to a real world corpus of 87 reports written in French and Swedish for which it can synthesize 35% of the texts, and its visualiser can reproduce approximately 60% of manually created templates. Because of its application domain, the 3D models needed in CarSim are restricted and hence the performance is less affected by the resource of 3D models than CONFUCIUS.

Figure 9.1 also shows that systems with high graphic quality or rich multimodal interfaces usually provide little NLP, and those with more NLP have fewer interface channels. Our research in the form of CONFUCIUS overcomes the limitations of previous systems and achieves the best overall performance. This is mainly due to its NLP and high automation in animation generation. CONFUCIUS provides a methodology of semantic representation called LVSR and novel language ontology based on multiple modalities. Current state of the art techniques in natural language processing and speech synthesis, automatic 3D animation design, media design and coordination are utilized and incorporated in CONFUCIUS.

9.2.2 Comparison of LVSR and LCS

CONFUCIUS' LVSR is based on Jackendoff's (1990) LCS and is adapted to the task of language visualisation. LVSR introduces finer ontological categories of concepts and adds basic human actions as EVENT predicates since LCS' event predicates are too coarse for character animation generation. Table 9.1 shows the difference in ontological categories of concepts between LCS and LVSR. LVSR distinguishes OBJ (non-animated Thing) and HUMAN (animated, articulated Thing), both of which belong to *Thing* in LCS classification. In LVSR HUMAN can be either human being or any other articulated, animated character (e.g. animal) as long as its skeleton hierarchy is defined in the graphic library. OBJ can be props or places (e.g. buildings). Moreover, LVSR adds the category TIME which is solved in LCS by adding a temporal feature to PLACE predicates.

<i>Semantic representation</i>	<i>Ontological categories</i>
LCS	Thing, Event, State, Action, Place, Path, Property, and Amount
LVSR	OBJ, HUMAN, EVENT, STATE, PLACE, PATH, PROPERTY, TIME, and AMOUNT

Table 9.1: Comparison of conceptual categories of Jackendoff's LCS and LVSR

These differences are primarily for the purpose of generating humanoid character animation, and they provide a finer selection restriction facility. Since most of story animation concerns humanoid characters, Jackendoff's original LCS is not suitable for the diversity of human actions. For instance, the `Event cause` in LCS is overloaded by including both phrasal causations (e.g. `cause`, `force`, `prevent`, `impede`) and lexical causatives (e.g. `push`, `break` (vt.), `open`, `kill`). A direct consequence of generalizing lexical causatives to `[Event cause]` is

indistinctness between action verbs. LVSR solves this problem by distinguishing each distinct human action as an `EVENT` predicate, and hence has a distinct animation model in the graphic library. Therefore, LVSR provides not only a finer ontological category system but also a richer representation of events.

9.2.3 Comparison of interval algebra and Badler's temporal constraints

Previous temporal representation, analysis and reasoning in syntax (e.g. tense and aspect) and pragmatics is at the sentence level, while research on lexical semantics takes few temporal relations into consideration. All temporal relations research within natural language processing is limited within the language modality itself and does not take other modalities such as vision into account. Our work brings interval temporal logic, which is significantly more expressive and more natural for representing events and actions, to the visual semantics of verbs at the lexical level and uses this methodology to enhance our compositional visual definition of action verbs for dynamic language visualisation.

Pinhanez et al. (1997) use interval logic in storytelling, but they use it to describe the relationships between the time intervals of events and interactions, i.e. the storyline. Table 9.2 shows a comparison with Badler's (Badler et al. 1997) temporal constraints for actions in the technical orders (instruction manuals) domain. We can use interval logic to represent all the five constraints they put forward. Their constraints are compositional (e.g. jointly parallel deals with three actions), and all the constraints are disjunctions of several interval relations. They also consider other non-temporal factors such as dominancy of action (e.g. while parallel). We claim that interval relations are more flexible and suitable for general purposes since they are 'minimal' relations of time intervals. For domain-specific applications such as technical instructions, their specific temporal representation may work well.

<i>Badler's temporal constraints (technical orders domain)</i>	<i>Interval relations</i>
Sequential	{p,m}
Parallel	{s,s ⁻¹ ,≡}
Jointly parallel	(act1 {s,s ⁻¹ ,≡} act2) {p,m} act3
Independently parallel	{f,f ⁻¹ ,≡}
While parallel	act_dominant {s ⁻¹ ,f ⁻¹ ,≡} act_indominant

Table 9.2: Comparison of interval algebra and Badler's temporal constraints

9.2.4 Comparison of visual semantic ontology and Levin's classes

In many ways CONFUCIUS' verb ontology discussed in Chapter 5 is related to that of Levin (1993). However, our point of departure and underlying methodology are different. We categorise verbs from the visual semantic perspective since language visualisation in CONFUCIUS provides independent criteria for identifying classes of verbs sharing certain

aspects of meaning, i.e. semantic/visual correlations. A visual semantic analysis of event verbs has revealed influences in a taxonomic verb tree. Various criteria ranging from visual valency, somatotopic effector, to LOD are proposed for classifying verbs from the language visualisation perspective.

The relation to Levin's classes is not one-to-one. Figure 9.2 shows examples of one-to-many and many-to-one relations to Levin's verb classes. For instance, the verb "cut" in "Carol cut the whole wheat bread" and "Whole wheat bread cuts easily" belongs to the single Levin class "verb of cutting", while the first "cut" is a three visual valency verb belonging to CONFUCIUS' class 2.2.1.3.2 and the second "cut" is a two visual valency verb in the class 2.1.2. In the second case, both the verbs "bring" and "give" are in CONFUCIUS' class 2.2.1.3.1, involving three visual roles, but "bring" belongs to the Levin's class "verbs of sending & carrying" and "give" to "verbs of change of possession". It is obvious that Levin's classes focus more on language semantics, while our categories stress visual semantics.

Example sentences	CONFUCIUS' verb categories of visual semantics	Levin's verb classes
"Carol cut the whole wheat bread." "Whole wheat bread cuts easily."	2.2.1.3.2, visual valency=3 2.1.2, visual valency=2	Verbs of cutting
	↑	1 to N
"Nancy brought the book to John." "Nancy gave the book to John."	2.2.1.3.1 visual valency=3	Verbs of sending & carrying Verbs of change of possession
	↑	N to 1

Figure 9.2: Relation of CONFUCIUS' verb ontology and Levin's verb classes

9.2.5 Comparison of action decomposition and scripts

The action decomposition structure that we discussed in Chapter 4, section 4.5 can be regarded as an extension of scripts. Compared with scripts our method does not use primitives to define events, instead, we use system-known events whose animations are available in the graphic library to define system-unknown events. Table 9.3 shows the common points and differences between our method and scripts. Both scripts and the action decomposition structure translate high level events to lower level events. Level 1 includes routine events (complex activities) that are either lexicalised to verbs (e.g. "interview") or verb phrases (e.g. "eat out", "see a doctor"). Level 2 are simple action verbs such as "jump", "push". Level 3 is a finite set of universal semantic components (primitives or atomic actions) into which all event verbs/verb phrases could be exhaustively decomposed. In CONFUCIUS' action decomposition, the lowest level is using VHML to specify simple action verbs which are not available in the animation library. The major difference between scripts and CONFUCIUS' action decomposition is in level 2 to 3 translation. CONFUCIUS' decomposition structure focuses on visual presentation of events while scripts focus on language comprehension.

<i>Event levels</i>	<i>Example verbs</i>
1. Routine events, complex activities	rob, cook, interview, eatOut
2. Simple action verbs	jump, lift, give, walk, push
3. Primitive actions	(<i>Script</i>) ATRANS, PTRANS, MOVE
	(<i>VHML</i>) move, rotate

Table 9.3: Comparison of CONFUCIUS' action decomposition and scripts

9.2.6 Commonsense knowledge reasoning

Commonsense reasoning is crucial in solving underspecification problem of natural language for visualising humanoid activities. Previous language-to-vision applications hard-code commonsense knowledge, which is needed for filling in missing/underspecified information when presented in visual modalities, either into the systems' vocabulary (Badler 1997, Coyne & Sproat 2001), e.g. telic roles of objects and default arguments of actions, or into a structure like Schank's scripts (Narayanan et al. 1995), e.g. the prop "gun" in a "robbery" script. In Chapter 5, section 5.5, we have presented a methodology to extract such knowledge from the existing computational lexicon WordNet to meet the needs of explicit information required in language visualisation.

9.3 Future work

Automatic conversion of natural language to 3D animation is one of the most challenging applications of NLP and computer animation. It has not matured enough that it can be used in intelligent storytelling or the animation production process. There are a number of things to be improved and problems to be solved before this work can be widely applied.

With respect to NLP, this research can be extended by investigating how to use visual and audio presentation to cover more verb classes. In this work, only simple verbs are presented by visual and auditory display. Verbs like "marry", for instance, which require stereotypical action sequences of prior experience knowledge within human beings' common sense are either conveyed through speech modality or request user intervention for specifying animations. Future research needs to be conducted to extend the knowledge base in order to take over users' load on animation specification.

Extracting common sense knowledge from existing lexicons is another future direction. In Chapter 5, section 5.5, a selection algorithm for finding the highest hypernym of default instruments of verbs from WordNet is described. However, finding appropriate hyponyms from lexical knowledge and context is still an open issue. For example, given the default instrument "cutting implement" for a verb sense of "cut", find an appropriate hyponym ("scythe") for the sentence "John cut the crop". Domain information like Figure 5.5 (1) provides a clue to solve the problem.

In terms of temporal representation, one of the limitations of CONFUCIUS' temporal representation is lack of quantitative information, which is due to our adoption of the interval-

based relations: (1) the durations of activities cannot be specified, though repetition of activities could be indicated by defining one repeatable period and specifying its repeat attribute; (2) for overlapping events $x \{o, o^{-1}\} y$, our temporal representation only works when the exact start point of y is unimportant; (3) for events $x \{p, p^{-1}\} y$, it is difficult to relate the distance between the two intervals, i.e. the distance between the end point of x and the start point of y in the case of $x p y$. Future versions of the compositional visual representation may introduce quantitative elements to overcome these limitations, resulting in more precise animation blending of temporal overlapping activities.

In order to accomplish *real* storytelling, current story understanding and presentation must be extended to discourse level, which requires processing to go beyond the sentence, e.g. resolving inter-sentence reference. Further research may also be conducted on narrative theory and extracting emotional information from natural language. A narrative theory, which specifies how storytelling systems should select and order details presented to the user, can make CONFUCIUS' storytelling more realistic. This task of selection and ordering is similar to that in traditional film-editing or animation production. Emotion understanding is another challenge in natural language processing which is crucial for the characters' performance in storytelling.

In addition, porting CONFUCIUS to other languages is feasible by integrating available tools of machine translation into it. The visual and auditory semantic based language ontology described in Chapter 5 is intended to be language-neutral. Interfacing with machine translation tools involves converting texts into LVSR representations that CONFUCIUS' animation engine can interpret.

With respect to 3D graphics, many details could be improved such as physics fidelity, clothing, face deformation for expressions and lip synchronisation, and multiple character interaction. A physics engine is needed to simulate real world naïve physics, such as kinematics, gravity, collision detection and response, friction and dynamics, and to make the virtual world more realistic. Figure 9.3 shows a design of animation generation using a physics engine. Possible candidate physics engines are Novodex SDK (Novodex Physics 2005) and Open Dynamics Engine (ODE, Smith 2005).

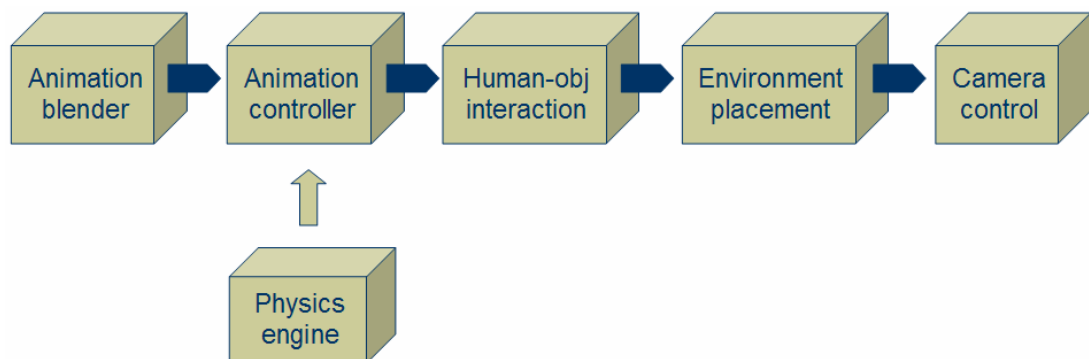


Figure 9.3: The physics engine in animation generation

Moreover, there are various directions in which CONFUCIUS has potential expansions. The first direction is multimodal interaction. Since the story world in CONFUCIUS is modelled by a Virtual Reality language which provides facilities for user interaction in the virtual world, it would be possible to extend the system to interactive storytelling in the future. Second, though CONFUCIUS' storytelling is intelligent as a whole, the characters in the stories do not stand on their own because their personality, response to the world and other characters, and even sense are all decided and described in the input texts, which is the reason that CONFUCIUS is only a story *interpretation and presentation* system. By combining believable agents (Loyall 1997) which have their own sensors, spontaneous response to the environment, individual behaviours, likes and dislikes (emotional properties), once their initial goals are set by the author, the current version of CONFUCIUS may be expanded to have story generation ability. With this ability, it will translate stories even from incomplete input.

From methodology aspect, we have only used symbolic reasoning techniques in this work. Numeric, statistical, and probabilistic approaches, such as neural networks, fuzzy logic, and genetic algorithms may be useful, and we hope to investigate their application to the storytelling domain in the future. Some of these methods have already been used in NLP to understand natural language better, and we believe it is possible to use these approaches to produce interesting and acceptable story presentations.

9.4 Conclusion

The purpose of this research was to investigate the process of mental imagery from a computational perspective, employing theories and resources from linguistics, natural language processing, and computer graphics about human language visualisation. In order to conduct this investigation, an intelligent multimedia storytelling interpretation and presentation system, CONFUCIUS, was implemented. It creates 3D animations with speech and nonspeech auditory display from natural language texts. The benefits of the research includes a novel semantic representation LVSR, linking language modalities to visual modalities; automatic animation generation which combines precreated and dynamically generated (procedural) animation facilities into a unified mechanism; language ontology based on visual and auditory semantics; effective word sense disambiguation approaches; an overall framework of intelligent storytelling, using computer graphics techniques with NLP to achieve high-level animation generation; and a testbed for evaluating text-to-graphics applications.

The evaluation results show that these approaches do contribute in automatic generating virtual worlds from human natural language with little user interaction. Suggestions are made for future work to improve graphic quality, to extend the language knowledge base in order to cover more verb classes, to include discourse level analysis, and to port CONFUCIUS to other languages. Combined with a larger graphic library of 3D models and human animations the

overall outcome of the work has the potential to impact on a wide variety of prospective areas such as virtual reality, computer games, movie production and direction, education, and intelligent multimedia.

Appendices

Appendix A

A VHML Example from Virtual Storyteller

This appendix shows the VHML code of a movie example from the Facial Animation subsystem (VHML Examples 2005). It uses Facial Animation Markup Language (FAML) and Speech Markup Language (SML) to specify synthesized speech and the virtual storyteller's facial expressions.

```
<?xml version="1.0"?>
<!DOCTYPE vhtml SYSTEM "./vhtml-v01.dtd">
<vhtml>
  <p>
    <neutral><pitch range="+150%">A <smile 2 5 5000/>little dog goes
into
    <l_roll 2 4 1200/><nod 2 3 1200/> a saloon in the Wild West, and
    <r_roll 2 6 1000/><nod 2 3 1000/><hl 2 4 1000/> beckons to the
bartender.
    </pitch></neutral>
  </p>

  <p>
    <neutral><speaker gender="male" name="us1">
    <pitch range="+100%"><hl 2 5 800/> <g_left 2 8 1100/>
    <pause msec="3800"/> <hr 2 5 800/> <g_right 2 8 1100/>
    Hey <emph_GST 2 7 1400/><r_roll 2 8 1400/>, bartender, give me a
whiskey.'
    </pitch>
    </speaker></neutral>
  </p>

  <p>
    <neutral>The <smile 2 4 1500/> bartender ignores him.
    </neutral>
  </p>

  <p>
    <neutral><speaker gender="male" name="us1">
    'Hey <emph level="moderate" affect="p">bartender <emph_GST 2 5
1500/>
    <anger 2 2 1500/> </emph>, give me a
    <emph level="moderate" affect="p">whiskey!'</emph>
    </speaker></neutral>
  </p>

  <p>
    <neutral>
    <emph>Still <nod 2 2 500/><r_roll 2 3 500/> </emph>
    <pitch middle="-20%">ignored.</pitch>
    </neutral>
  </p>

  <p>
    <neutral><speaker gender="male" name="us1">
    <anger 2 4 2000/>'HEY <emph_GST 2 6 2000/> <r_roll 2 6 2000/>
    <emph level="strong">BARMAN</emph>!
```

GIMME A **WHISKEY!**'

The bartender takes out his six-shooter and **smiles** **rolls** **shoots** the dog in the **leg**, and the dog runs out the saloon, **howling** in pain.

Three **years** later, the **wee dog** appears **again**, wearing boots, jeans, **chaps**, a Stetson, **gun belt**, and **guns**. He **ambles** **slowly** into the saloon, goes up to the **bar**, leans **over** it, and says to the bartender,

'I'm here to **git** the man that **shot** my **paw**.'

Appendix B

Working Files of Semantic Analysis

Input: "For the Duchess. An invitation from the Queen to play croquet." The Frog-Footman repeated, in the same solemn tone. (From "Alice in Wonderland")

B.1 Analysis of Machine Syntax for English:

#	Text	Baseform	Syntactic relation	Syntax and morphology
1	"	"		
2	For	for		@ADVL %EH PREP
3	the	the	det:>4	@DN> %>N DET
4	Duchess	duchess	pcomp:>2	@<P %NH N NOM SG
5	.	.		
6	<s>	<s>		
1	An	an	det:>2	@DN> %>N DET SG
2	invitation	invitation		@NH %NH N NOM SG @PCOMPL-S %NH N NOM SG @SUBJ %NH N NOM SG
3	from	from	mod:>2	@<NOM %N< PREP
4	the	the	det:>5	@DN> %>N DET
5	Queen	queen	pcomp:>3	@<P %NH N NOM SG
6	to	to	pm:>7	@INFMARK> %AUX INFMARK>
7	play	play	mod:>5	@-FMAINV %VA V INF
8	croquet	croquet	obj:>7	@OBJ %NH N NOM SG
9	.	.		
10	"	"		
11	<s>	<s>		
1	The	the	det:>2	@DN> %>N DET
2	Frog-Footman	frog-footman	subj:>3	@SUBJ %NH <?> N NOM SG
3	repeated	repeat	main:>0	@+FMAINV %VA V PAST
4	,	,		
5	in	in	ha:>3	@ADVL %EH PREP
6	the	the	det:>9	@DN> %>N DET
7	same	same	attr:>8	@A> %>N A ABS

8	solemn	solemn	attr:>9	@A> %>N A ABS
9	tone	tone	pcomp:>5	@<P %NH N NOM SG
10	.	.		
11	<s>	<s>		

B.2 The output file after adding semantic features

```

1 " "
2 For for @ADVL %EH PREP
3 the the det:>4 @DN> %>N DET
4 Duchess duchess pcomp:>2 @<P %NH N PERS_F NOM SG
5 . .
6 <s> <s>

1 An an det:>2 @DN> %>N DET SG
2 invitation invitation @NH %NH N NOM SG
@PCOMPL-S %NH N NOM SG
@SUBJ %NH N NOM SG
3 from from mod:>2 @<NOM %N< PREP
4 the the det:>5 @DN> %>N DET
5 Queen queen pcomp:>3 @<P %NH N PERS_F NOM SG
6 to to pm:>7 @INFMARK> %AUX INFMARK>
7 play play mod:>5 @-FMAINV %VA V INF COGNI
8 croquet croquet obj:>7 @OBJ %NH N NOM SG
9 . .
10 " "
11 <s> <s>

1 The the det:>2 @DN> %>N DET
2 Frog-Footman frog-footman subj:>3 @SUBJ %NH <?> N PERS NOM SG
3 repeated repeat main:>0 @+FMAINV %VA V PAST COMMU
4 , ,
5 in in ha:>3 @ADVL %EH PREP
6 the the det:>9 @DN> %>N DET
7 same same attr:>8 @A> %>N A ABS
8 solemn solemn attr:>9 @A> %>N A ABS
9 tone tone pcomp:>5 @<P %NH N NOM SG
10 . .
11 <s> <s>

```

Appendix C

LCS Notation

This appendix lists notation of the LCS database, including logical arguments and modifiers, and specification of thematic roles.

C.1 Logical arguments

1 = AG, agent Logical subject of CAUSE, LET. If restricted to [-animate], an instrumental subject, as in The hammer broke the vase.

2 = EXP, experiencer Logical subject of GO PERC, BE PERC, ACT_ON PERC, ACT PERC, PATH_ACT PERC.

2 = INFO, information Logical subject of GO COMM, BE COMM

2 = TH, theme Logical subject of everything not mentioned under AG and PERC (8 below). In previous versions of the database, every LCS had either a TH or an EXP. Subsequent addition of the ACT primitive introduced the possibility of AG-only LCSs.

3 = SRC(), source preposition Indicates Path FROM or Path AWAY_FROM. Marks path prepositions or particles indicating source, e.g. from, away from, etc. (ex. John ran *away* from home).

4 = SRC, source Logical argument Paths FROM and Path AWAY_FROM. The SRC role indicates where the TH started its motion (in LOC), what its original state (IDENT) was, or where its original (possibly abstract) location was (in POSS) (ex. John left *the house*).

5 = GOAL(), goal preposition Indicates Position AT (in a Path-like sense) or Paths TO/TOWARD (in any field except Ident). This slot marks path prepositions or particles indicating goals, e.g. to, toward (ex. John ran *to* the store).

5 = PRED(), pred preposition Indicates Paths TO/TOWARD Ident. This slot marks path prepositions or particles indicating goals in the identificational field, e.g. to, toward (ex. John turned into a monkey).

6 = GOAL, goal Logical argument of Path TO (in any field except Ident) or TOWARD (in the LOC field only). The GOAL role indicates the endpoint of motion (in LOC) or the final (possibly abstract) location (ex. John ran *home*).

7 = PERC(), perceived item particle Indicates Paths TO/TOWARD PERC (ex. He looked *into the room*).

8 = PERC, perceived item Logical subject of GO PERC, BE PERC, ACT_ON PERC, ACT PERC, PATH_ACT PERC. The PERC role indicates entities that may be perceived (ex. He saw *the play*).

9 = PRED, identificational predicate Logical argument of GO IDENT, BE IDENT, ACT_ON IDENT, ACT IDENT, PATH_ACT IDENT. A thing/property (in IDENT) or a new state of existence (in EXIST) (ex. We considered him *a fool*).

10 = LOC(), locational particle Indicates Positions AT/IN/ON LOC. This slot marks prepositions preceding static locations, (ex. He lived *in* France).

11 = LOC, locational predicate Logical argument of BE LOC, STAY LOC. A static location, not a source or a goal in the LOC field (ex. The water fills *the box*.)

12 = POSS, possessional predicate Logical argument of BE POSS, Logical subject of GO POSS. POSS is the possessed entity. (ex. This box carries *five eggs*.)

13 = TIME(), temporal particle preceding time Indicates Positions AT/IN/ON TEMP or Paths TO/TOWARD/FROM/AWAY_FROM TEMP. This slot marks prepositions preceding time. Not currently used in the verbs, although maybe in composed structures (ex., John ate *at* nine).

14 = TIME, time for TEMP field Logical argument of BE TEMP, GO TEMP. Temporal argument, not currently used in verb lexicon, but in constant of verb lexicon (ex. John *summered* at the cabin) and in prep lexicon.

27 = PROP, event or state Logical argument of BE CIRC, GO CIRC, STAY CIRC. Non perceptual proposition.

C.2 Logical modifiers

15 = MOD-POSS(), possessional particle corresponding to Positions WITH/IN/OF/FROM POSS and FOR INSTR. Precedes a possessional modifier or monetary unit of exchange, e.g., He loaded the cart *with* hay, He bought it *for* \$5.

16 = MOD-POSS, possessed item modifier, e.g., money, rights. Logical argument of possessive modifier inside WITH/IN/OF/FROM POSS.

17 = BEN(), intentional particle corresponding to Position FOR POSS. Precedes benefactive modifier, e.g., John baked the cake *for* Mary.

18 = BEN, benefactive modifier, e.g., John baked *Mary* a cake. Logical argument of benefactive modifier inside for poss. (associated with collocation "to".)

19 = INSTR(), instrumental particle corresponding to Positions with/by/of/on instr. Precedes instrument modifier, e.g., John broke the glass *with* a stick.

20 = INSTR, instrument modifier and subject of cause, e.g., *The stick* broke the glass; John broke the glass with *a stick* logical argument of instrumental modifier inside with/by/of/on instr.

21 = PURP(), intentional particle corresponding to position for intent. Precedes purpose clause, e.g., He studied *for* the exam, He searched *for* rabbits. (Indicates "because of" or searched item.)

22 = PURP, purpose modifier or reason, e.g., He studied for *the exam*). We currently don't have any uses without particles in our verb lexicon.

23 = MOD-LOC(), location particle corresponding to a Positions AROUND/FROM/DOWN/etc LOC. Precedes locational modifier, e.g., She held the child *in* her arms.

24 = MOD-LOC, location modifier Logical argument of locational modifier inside AROUND/FROM/DOWN/etc LOC. A location that isn't required by the verb but modifies the entire situation. Especially for composed structures.

25 = MANNER() (not -ingly)

26 = reserved for conflated manner component (-ingly)

28 = MOD-PROP, event or state. Non-perceptual proposition not required by the verb.

29 = MOD-PRED(), identificational particle corresponding to Position AS IDENT. Precedes property modifier, e.g., She imagined him *as* a prince.

30 = MOD-PRED, property modifier.

31 = MOD-TIME, time modifier.

32 = MOD-PERC(), perceptual modifier particle.

33 = MOD-PERC, perceptual modifier.

34 = PARTICLE, other particle, more often handled via collocation.

C.3 Thematic role specification

The format for thematic roles is the following: 1. Any theta role preceded by an underscore (_) is obligatory. 2. Any theta role preceded by an underscore (,) is optional. 3. Prepositions inside parentheses indicate that the corresponding phrases must necessarily be headed by the specified prepositions. 4. An empty set of parentheses () indicates that there NECESSARILY must be a prepositional head, but it is left unspecified. If there are no parentheses, then there is no prepositional head.

Example: ag_th,src(from),goal(to)

Here, ag and th are obligatory; the rest are optional.

Although theta roles are theoretically unordered, they are generally specified in a "canonical" ordering that is thought to arise most frequently. In general, surface order reflects the numeric order of thematic roles (e.g. 1 = AG before 2=TH). If numeric order is NOT reflected in the surface order, an :int and :ext operator is encoded in the LCS (as discussed above) indicating that the logical subject is the semantic object, and the logical object is the semantic subject. This parameter can also apply to PP roles. Thus, the _th_loc grid is NOT the same as the _loc_th grid, as seen in the examples below:

THETA_ROLES: "_loc_th"

SENTENCE: "The box holds the ball"

THETA_ROLES: "_th_loc"

SENTENCE: "The water fills the box"

Appendix D

Connexor FDG Notation

This appendix lists all the syntactic, morphological and dependency tags used in the Connexor Machine FDG parser for English.

D.1 English dependency functions

<i>Tag</i>	<i>Explanation</i>	<i>Example</i>
main	main element: main nucleus of the sentence; usu. main verb of the main clause	The Berkeley UNIX <i>mechanism</i> for creating a virtual connection between processes. Sockets <i>form</i> the interface between UNIX standard I/O and network communication facilities.
qtag	tag question	It is cold, <i>isn't it?</i>
v-ch	verb chain: auxiliaries + main verb	If you're running the mess-dos emulator, control-alt-insert <i>will</i> cause a soft boot of the emulator, while leaving the rest of the system running.
pm	preposed marker: grammatical marker of a subordinated clause. The marker (subordinating conjunction) itself doesn't have a syntactic function in the subordinated clause.	Others go further and define software <i>to</i> be programs plus documentation <i>though</i> this does not correspond with common usage.
pcomp	prepositional complement: the head of a nominal construction (NP or non-finite clause or nominal clause) that, together with a preposition, forms a prepositional phrase. Usually a preposition precedes its complement, but also topicalised complements occur.	They are in that red <i>car</i> . She is fond of <i>walking</i> long distances. <i>What</i> are you afraid of?
phr	verb particle: certain preposition-adverb homographs that form a phrasal verb with a verb	She looked <i>up</i> the word in the dictionary.
subj	subject: the head of an NP that agrees in number with the verb in the clause. Often signals the semantic category called agent.	<i>John</i> is in the kitchen.
agt	agent: The agent by-phrase in passive sentences.	The dog was chased <i>by</i> the boys.
obj	object: the head of the other main nominal dependent of transitive verbs (and ditransitive verbs, together with indirect objects)	John saw an <i>apple</i> . John gave him an <i>apple</i> .
comp	subject complement:	John remains a <i>boy</i> . What you see is

<i>Tag</i>	<i>Explanation</i>	<i>Example</i>
	the head of the other main nominal dependent of copular verbs.	what you <i>get</i> . John is <i>foolish</i> .
dat	indirect object: Ditransitive verbs can take three nominal dependents: subject, indirect object, object.	John gave <i>him</i> an apple.
oc	object complement: a nominal category that occurs along with an object for object complementiser verbs.	John called him a <i>fool</i> . John considers him <i>foolish</i> .
copred	copredicative	John regards him <i>as</i> foolish.
com	comitative	Drinking <i>with</i> you is nice.
voc	vocative	<i>John</i> , come here!
ins	instrument	He sliced the salami <i>with</i> the knife.
tmp	time	If you're running the mess-dos emulator, control-alt-insert will cause a soft boot of the emulator, <i>while</i> leaving the rest of the system running.
dur	duration	The OECD praises the relative stability of US unemployment as "remarkable", given the 50 per cent increase in the American US labour force <i>in</i> the past 25 years.
frq	frequency	It <i>often</i> involves the use of CASE tools.
qua	quantity	Singapore says <i>more</i> about this than Hong Kong.
man	manner	If Europe is so wonderful, they argue, why does its job creation record compare so <i>poorly</i> with that of the United States?
loc	location	That exacerbates a key problem <i>in</i> America, the skills gap.
sou	source	Policymakers in both seem to be moving away <i>from</i> the characteristics that defined them.
goa	goal	Virgin is expected to try to move <i>to</i> a full anti-trust trial.
pth	path	He travelled from Tokyo <i>to</i> Beijing.
cnt	contingency (purpose or reason)	The DTI was unable to say last night <i>why</i> the approach for Frances Colliery had been rejected.
cnd	condition	If Europe <i>is</i> so wonderful, they argue, why does its job creation record compare so poorly with that of the United States?
meta	clause adverbial	<i>So far</i> , the OECD has refused to disclose its country-by-country studies.

<i>Tag</i>	<i>Explanation</i>	<i>Example</i>
cla	clause initial adverbial	<i>Under</i> President Clinton, the highly flexible US labour market is becoming more regulated.
ha	heuristic prepositional phrase attachment	Eventually the beam will escape <i>through</i> the partially reflective mirror.
qn	quantifier	IFA Promotion, which represents more than <i>15,000</i> independent financial advisers, commissioned a random poll of new year resolutions.
det	determiner	Nearly <i>a</i> third of East Anglians resolved to stay healthy in 1995.
neg	negator	What a blessing the releases were <i>not</i> in Finnish!
attr	attributive nominal	By <i>Philip Bassett</i> , <i>Industrial</i> Editor.
mod	other postmodifier	Ministers will see the OECD report <i>on</i> the UK labour market as international recognition <i>of</i> their reform <i>of</i> one <i>of</i> the economy's most difficult areas.
ad	attributive adverbial	<i>So</i> much for modern technology.
cc	Coordination: The coordinating conjunction and one coordinated element are linked to the other coordinated element. Multiple coordinated elements are chained together. The upmost element in a chain shows the functional role of the coordinated units.	<i>Jack and Jill</i> bought some pins, <i>nails and needles</i> .

D.2 English morphological tags

<i>Part of speech</i>	<i>Subfeature</i>	<i>Explanation</i>	<i>Example</i>
N		noun	These integrated <i>algorithms</i> are stored on the <i>computer's</i> hard disk.
-- case	NOM	nominative	These integrated <i>algorithms</i> are stored on the <i>computer's</i> hard disk.
	GEN	genitive	These integrated <i>algorithms</i> are stored on the <i>computer's</i> hard disk.
-- number	SG	singular	These integrated <i>algorithms</i> are stored on the <i>computer's</i> hard disk.
	PL	plural	These integrated <i>algorithms</i> are stored on the <i>computer's</i> hard disk.

With nouns, the obligatory tags include 'N' and case. In Conexor FDG, the obligatory tags for nouns include 'N', case, and number.

ABBR		abbreviation	"SODA Manual of Operation", R. C. Brigham and C. G. Bell, School of Elec Eng, U New S
------	--	--------------	---

			Wales, Sydney, NSW (1958)
-- case and number like in nouns			

With abbreviations, the obligatory tags include 'ABBR' and case. In Conexor FDG, the obligatory tags for abbreviations include 'ABBR', case and number.

<i>Part of speech</i>	<i>Subfeature</i>	<i>Explanation</i>	<i>Example</i>
A		adjective	These integrated algorithms are stored on the computer's <i>hard</i> disk, from which they are downloaded into the DSP board's <i>random</i> access memory (RAM).
-- comparison	ABS	absolute	<i>big</i>
	CMP	comparative	<i>bigger</i>
	SUP	superlative	<i>biggest</i>

With adjectives, the obligatory tags include 'A' and degree of comparison.

NUM		numeral	Software can be split roughly into <i>two</i> main types - system software and application software or programs.
	CARD	cardinal	<i>2010</i>
	ORD	ordinal	<i>first</i>
-- number	SG	fraction, singular	<i>one-third</i>
	PL	fraction, plural	<i>two-thirds</i>

With numerals, the obligatory tags include 'NUM' and either 'CARD' or 'ORD'.

PRON		pronoun	<i>Others</i> go further and define software to be programs plus documentation though <i>this</i> does not correspond with common usage.
-- case and related features	NOM	nominative	<i>others</i>
	GEN	genitive	<i>other's</i>
	ACC	accusative	<i>him</i>
	INDEP	the independent genitive form functioning always as head of a noun phrase	<i>theirs</i>
-- number	SG	singular	<i>other</i>
	SG1	singular, first person	<i>me</i>
	SG3	singular, third person	<i>him</i>
	PL	plural	<i>others</i>

	PL1	plural, first person	<i>us</i>
	PL3	plural, third person	<i>them</i>
-- comparison	ABS	absolute	<i>much</i>
	CMP	comparative	<i>more</i>
	SUP	superlative	<i>most</i>
-- other pronoun subfeatures	PERS	personal	<i>us</i>
	DEM	demonstrative	<i>these</i>
	RECIPR	reciprocal	<i>each other</i>
	WH	relative or interrogative pronoun beginning with the letters 'wh' or 'how'	<i>which</i>
	<Interr>	interrogative	<i>why</i>
	<Refl>	reflexive	<i>herself</i>
	<Rel>	relative	<i>which</i>

With pronouns, the only obligatory tag is 'PRON'. The anglebracket tags occur in front of the 'PRON' tag, when relevant. After the 'PRON' tag come the tags 'PERS', 'RECIPR', 'WH', 'DEM', or degree of comparison, when relevant. Next is the place for case, then number, and last, 'INDEP', when relevant.

DET		determiner	If you're running <i>the</i> mess-dos emulator, control-alt-insert will cause <i>a</i> soft boot of <i>the</i> emulator, while leaving <i>the</i> rest of <i>the</i> system running.
-- case	GEN	genitive	<i>whose</i>
-- number	SG	singular	<i>an</i> option
	PL	plural	<i>these</i> options
-- comparison	ABS	absolute	<i>many</i> options
	CMP	comparative	<i>more</i> options
	SUP	superlative	<i>most</i> options
-- other subfeatures of determiners	DEM	demonstrative	<i>this</i> option
	WH	determiner beginning with the letters 'wh' or 'how'	<i>which</i> option

With determiners, the only obligatory tag is 'DET'. In Conexor FDG, the obligatory tags for determiners include 'DET' and number. In the ordering of the morphological tags, the number tag is the last one and the case tag second to last, when relevant.

ADV		adverb	Others go <i>further</i> and define software to be
-----	--	--------	--

			programs plus documentation though this does not correspond with common usage.
-- comparison	ABS	absolutive	<i>far</i>
	CMP	comparative	<i>further</i>
	SUP	superlative	<i>furthest</i>
-- other subfeatures for adverbs	<Ex>	existential <i>there</i>	<i>There</i> are various models of the software life-cycle, and many methodologies for the different phases.
	WH	adverb beginning with the letters 'wh' or 'how'	<i>why</i>

With adverbs, the obligatory tags include 'ADV'. The anglebracket tag again occurs in front of the 'ADV' tag, when relevant.

ING		present participle	The Berkeley UNIX mechanism for <i>creating</i> a virtual connection between processes.
EN		past participle	These <i>integrated</i> algorithms are <i>stored</i> on the computer's hard disk, from which they are <i>downloaded</i> into the DSP board's random access memory (RAM).

The tags EN and ING are used in Conexor FDG Lite for participles in all syntactic functions, whereas in Conexor FDG, they are used only for participles functioning as a verb. In Conexor FDG, participles in nominal functions are classified as adjectives or nouns. Thus, in the above example, the word *integrated* receives the tag EN in Conexor FDG Lite, whereas in Conexor FDG, it is classified as an adjective.

V		verb; used only for finite verbs and infinitives	Others <i>go</i> further and <i>define</i> software to <i>be</i> programs plus documentation though this <i>does not correspond</i> with common usage.
	AUXMOD	modal auxiliary	<i>would</i>
	INF	infinitive	<i>would be</i>
	IMP	imperative	John, <i>come</i> here!
	SUBJUNCTIVE	subjunctive	The casket <i>be</i> brought in.
-- tense	PRES	present tense	<i>are</i>
	PAST	past tense	<i>were</i>
-- person	SG1	singular, first person	<i>am</i>
	SG3	singular, third person	<i>is</i>
-- other	<N+>	noun-verb	<i>India's got..</i>

subfeatures for verbs		combination	
-----------------------	--	-------------	--

With verbs, the obligatory tags include 'V' and one of the following: 'AUXMOD', 'INF', 'IMP', or tense. With tense, person is possible. The subfeature <N+> is placed before the 'V' tag when relevant.

INTERJ	interjection	<i>Hey, so-and-so needs an instruction to do such-and-such.</i>
CC	coordinating conjunction	<i>and</i>
CS	subordinating conjunction	<i>if</i>
PREP	preposition	<i>of</i>
NEG-PART	the negative particle	<i>are not, aren't</i>
INFMARK>	infinitive marker	<i>to do this in order to do that</i>
<?>	mark for unknown word; occurs in front of a part-of-speech tag	<i>mechansim</i>

D.3 English functional tags

<i>Tag</i>	<i>Explanation</i>	<i>Example</i>
@+FAUXV	Finite auxiliary predicator	If you're running the mess-dos emulator, control-alt-insert <i>will</i> cause a soft boot of the emulator, while leaving the rest of the system running.
@-FAUXV	Nonfinite auxiliary predicator	Software can <i>be</i> split roughly into two main types - system software and application software or programs.
@+FMAINV	Finite main predicator	Sockets <i>form</i> the interface between UNIX standard I/O and network communication facilities.
@-FMAINV	Nonfinite main predicator	If you're <i>running</i> the mess-dos emulator, control-alt-insert <i>will cause</i> a soft boot of the emulator, while <i>leaving</i> the rest of the system <i>running</i> .
@SUBJ	Subject	<i>Sockets</i> form the interface between UNIX standard I/O and network communication facilities.
@F-SUBJ	Formal subject	<i>There</i> are various models of the software life-cycle, and many methodologies for the different phases.
@OBJ	Object	If you're running the mess-dos <i>emulator</i> , control-alt-insert <i>will cause</i> a soft <i>boot</i> of the emulator, while leaving the <i>rest</i> of the system running.
@I-OBJ?	Indirect object?	John gave <i>him</i> an apple.

<i>Tag</i>	<i>Explanation</i>	<i>Example</i>
@PCOMPL-S	Subject complement	A statistic that is <i>content-free</i> , or nearly so.
@PCOMPL-O	Object complement	This downloading, or "booting" process of the PC-installed software algorithms occurs as part of the computer's power-up initialisation process in less than 100 milliseconds, making it <i>transparent</i> to the user.
@ADVL	Adverbial	Others go <i>further</i> and define software to be programs plus documentation though this does <i>not</i> correspond <i>with</i> common usage.
@O-ADVL	Object adverbial	She let him walk the <i>streets</i> in the cold and in the rain.
@APP	Apposition	Software can be split roughly into two main types - system <i>software</i> and application <i>software</i> or <i>programs</i> .
@NH	Stray noun phrase	The Berkeley UNIX <i>mechanism</i> for creating a virtual connection between processes.
@VOC	Vocative	<i>John</i> , come here!
@A>	Premodifier of a nominal	These <i>integrated</i> algorithms are stored on the <i>computer's hard</i> disk, from which they are downloaded into the <i>DSP board's random access</i> memory (RAM).
@DN>	Determiner	If you're running <i>the</i> mess-dos emulator, control-alt-insert will cause <i>a</i> soft boot of <i>the</i> emulator, while leaving <i>the</i> rest of <i>the</i> system running.
@QN>	Premodifying quantifier	This downloading, or "booting" process of the PC-installed software algorithms occurs as part of the computer's power-up initialisation process in less than <i>100</i> milliseconds, making it transparent to the user.
@AD-A>	Intensifier	An optical laser works by bouncing photons back and forth between two mirrors, one <i>totally</i> reflective and one <i>partially</i> reflective.
@<NOM-OF	Postmodifying prepositional phrase beginning with <i>of</i>	If you're running the mess-dos emulator, control-alt-insert will cause a soft boot <i>of</i> the emulator, while leaving the rest <i>of</i> the system running.
@<AD-A	Postmodifying intensifier	Compuserve developed the GIF format for graphical images many years <i>ago</i> .
@<NOM?	Postmodifier of a nominal	The Berkeley UNIX mechanism <i>for</i> creating a virtual connection <i>between</i> processes.
@INFMARK>?	Infinitive marker <i>to</i>	Others go further and define software <i>to</i> be programs plus documentation though this does not correspond with common usage.
@<P-FMAINV	Nonfinite clause as preposition complement	The Berkeley UNIX mechanism for <i>creating</i> a virtual connection between processes.
@<P	Other preposition complement	If you're running the mess-dos emulator, control-alt-insert will cause a soft boot of the <i>emulator</i> , while leaving the rest of the <i>system</i> running.

<i>Tag</i>	<i>Explanation</i>	<i>Example</i>
@CC	Coordinating conjunction	Others go further <i>and</i> define software to be programs <i>plus</i> documentation though this does not correspond with common usage.
@CS	Subordinating conjunction	Others go further and define software to be programs plus documentation <i>though</i> this does not correspond with common usage.
@DUMMY	Unspecified	<i>Hey</i> , so-and-so needs an instruction to do such-and-such.

D.4 English surface syntactic tags

There are two parallel tagsets of English surface syntactic tags. The FDG parser outputs a tagset beginning with '%' and the FDG Lite parser outputs a tagset beginning with '&'. The tags and their explanations are listed in the table below. Note that the tagsets differ also in other details besides their prefixes: in the surface syntactic tags of FDG, the letter E stands for 'adverb', therefore '%E>' pro '&>A'.

The surface syntactic tags of FDG Lite for English are allocated through a fast surface analysis: on a 233 MHz Pentium PC running Linux, FDG Lite for English analyses text at the speed of 2,000 words per second. The surface syntactic tags of FDG for English, on the other hand, are allocated based on full dependency parsing of the sentence structure. Consequently, their accuracy is better, their ambiguity is smaller, and the analysis takes more time: on a 233 MHz Pentium PC running Linux, FDG for English analyses text at the speed of 300 words per second.








<i>Tags in FDG Lite</i>	<i>Tags in FDG</i>	<i>Explanation</i>	<i>Example</i>
&>N	%>N	determiner or premodifier of a nominal	<i>These integrated</i> algorithms are stored on <i>the computer's</i> hard disk, from which they are downloaded into <i>the DSP board's</i> random access memory (RAM).
&NH	%NH	nominal head	<i>Sockets</i> form the <i>interface</i> between UNIX standard <i>I/O</i> and network communication <i>facilities</i> .
&N<	%N<	postmodifier of a nominal	The Berkeley UNIX mechanism <i>for</i> creating a virtual connection <i>between</i> processes.
&>A	%E>	premodifying adverb	An optical laser works by bouncing photons back and forth between two mirrors, one <i>totally</i> reflective and one <i>partially</i> reflective.
&AH	%EH	adverbial head (besides adverbs, applies to interjections, prepositions, and the negative particle)	Others go <i>further</i> and define software to be programs plus documentation though this does <i>not</i> correspond <i>with</i> common usage.
&A<	%<E	postmodifying adverb	He knew it well <i>enough</i> .
&AUX	%AUX	auxiliary verb or particle	Software <i>can be</i> split roughly into two main types - system software and application software or programs.
&VP	%VP	main verb in a passive verb chain	These integrated algorithms are <i>stored</i> on the computer's hard disk, from which they are <i>downloaded</i> into the DSP board's

<i>Tags in FDG Lite</i>	<i>Tags in FDG</i>	<i>Explanation</i>	<i>Example</i>
			random access memory (RAM).
&VA	%VA	main verb in an active verb chain	If you're <i>running</i> the mess-dos emulator, control-alt-insert will <i>cause</i> a soft boot of the emulator, while <i>leaving</i> the rest of the system <i>running</i> .
&>CC	%EH>	Introducer of coordination	<i>both</i> Harry and Bill came
&CC	%CC	coordinating conjunction	Others go further <i>and</i> define software to be programs <i>plus</i> documentation though this does not correspond with common usage.
&CS	%CS	subordinating conjunction	Others go further and define software to be programs plus documentation <i>though</i> this does not correspond with common usage.

Appendix E

List of Existing H-Anim Models

This appendix lists some of H-Anim models currently available on the Internet. The names, snapshots of the human models, their authors and URLs are included.

<i>H-Anim models</i>	<i>Names</i>	<i>Authors, URLs</i>
	Nancy	Cindy Ballreich http://www.ballreich.net/vrml/h-anim/nancy_h-anim.wrl
	Baxter Nana	Christian Babski http://ligwww.epfl.ch/~babski/StandardBody
	Y.T. Hiro	Matt Beitler http://www.cis.upenn.edu/~beitler/H-Anim/Models/H-Anim1.1/
	Dilbert	Matt Beitler http://www.cis.upenn.edu/~beitler/H-Anim/Models/H-Anim1.1/dilbert/
	Max	Matt Beitler http://www.cis.upenn.edu/~beitler/vrml/human/max/
	Jake	Matt Beitler http://www.cis.upenn.edu/~beitler/H-Anim/Models/H-Anim1.1/jake/
	Dork	Michael Miller http://students.cs.tamu.edu/mmiller/hanim/proto/dork-proto.wrl


```

        USE l_elbowRotInterp
        USE l_wristRotInterp
        USE r_shoulderRotInterp
        USE r_elbowRotInterp
        USE r_wristRotInterp
        USE whole_bodyRotInterp
        USE whole_bodyTranInterp
    ]
    directOutput TRUE

url "vrmlscript:
function initialize()
{
    for(i=0;i<InvolvedJointNameList.length;i++)
        if (InvolvedJointNameList[i] != 'INACTIVE')
            {
Browser.addRoute(walk_Time, 'fraction_changed', InvolvedJointPtrList[cur
rent_ptr], 'set_fraction');
                for(j=0;j<HumansList.length;j++)
Browser.addRoute(InvolvedJointPtrList[current_ptr], 'value_changed', Hum
ansList[j].joints[i+1], 'set_rotation');
                    current_ptr++;
            }

        for(j=0;j<HumansList.length;j++)
            {
Browser.addRoute(InvolvedJointPtrList[current_ptr], 'value_changed', Hum
ansList[j].joints[0], 'set_translation');
                //this is HumanoidRoot
Browser.addRoute(InvolvedJointPtrList[current_ptr+1], 'value_changed', H
umansList[j].joints[0], 'set_rotation');
            }
Browser.addRoute(walk_Time, 'fraction_changed', InvolvedJointPtrList[cur
rent_ptr], 'set_fraction');
Browser.addRoute(walk_Time, 'fraction_changed', InvolvedJointPtrList[cur
rent_ptr+1], 'set_fraction');
    }
}
"

```

F.2 Nana's joints list

```

joints [
USE hanim_HumanoidRoot      #0
USE hanim_sacroiliac       #1
USE hanim_l_hip            #2
USE hanim_l_knee           #3
USE hanim_l_ankle          #4
USE hanim_l_subtalar
USE hanim_l_midtarsal
USE hanim_l_metatarsal
USE hanim_r_hip            #8
USE hanim_r_knee           #9
USE hanim_r_ankle          #10
USE hanim_r_subtalar
USE hanim_r_midtarsal
USE hanim_r_metatarsal
USE hanim_vl5
USE hanim_vl4
USE hanim_vl3
USE hanim_vl2
USE hanim_vl1              #18
USE hanim_vt10
USE hanim_vt9

```

```

USE hanim_vt8
USE hanim_vt7
USE hanim_vt6
USE hanim_vt5
USE hanim_vt4
USE hanim_vt3
USE hanim_vt2
USE hanim_vt1
USE hanim_vc7
USE hanim_vc6
USE hanim_vc5
USE hanim_vc4           #32
USE hanim_vc3
USE hanim_vc2
USE hanim_vc1
USE hanim_skullbase     #36
USE hanim_l_sternoclavicular
USE hanim_l_acromioclavicular
USE hanim_l_shoulder   #39
USE hanim_l_elbow      #40
USE hanim_l_wrist      #41
USE hanim_l_thumb1
USE hanim_l_thumb2
USE hanim_l_thumb3
USE hanim_l_index1
USE hanim_l_index2
USE hanim_l_index3
USE hanim_l_middle1
USE hanim_l_middle2
USE hanim_l_middle3
USE hanim_l_ring1
USE hanim_l_ring2
USE hanim_l_ring3
USE hanim_l_pinky1
USE hanim_l_pinky2
USE hanim_l_pinky3
USE hanim_r_sternoclavicular
USE hanim_r_acromioclavicular
USE hanim_r_shoulder   #59
USE hanim_r_elbow      #60
USE hanim_r_wrist      #61
USE hanim_r_thumb1
USE hanim_r_thumb2
USE hanim_r_thumb3
USE hanim_r_index1
USE hanim_r_index2
USE hanim_r_index3
USE hanim_r_middle1
USE hanim_r_middle2
USE hanim_r_middle3
USE hanim_r_ring1
USE hanim_r_ring2
USE hanim_r_ring3
USE hanim_r_pinky1
USE hanim_r_pinky2
USE hanim_r_pinky3     #76
]

```

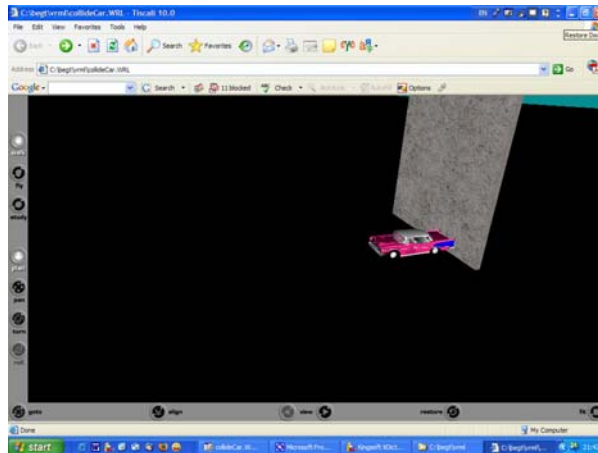
Appendix G

Evaluation Questionnaire

Thank you for completing the evaluation questionnaire of CONFUCIUS. Please click the pictures to see the animation. You can press F5 to play the animation again. You may adjust your view by using the navigation buttons on the left side, or change to another viewpoint from the context menu. To view the context menu, right-click the scene.

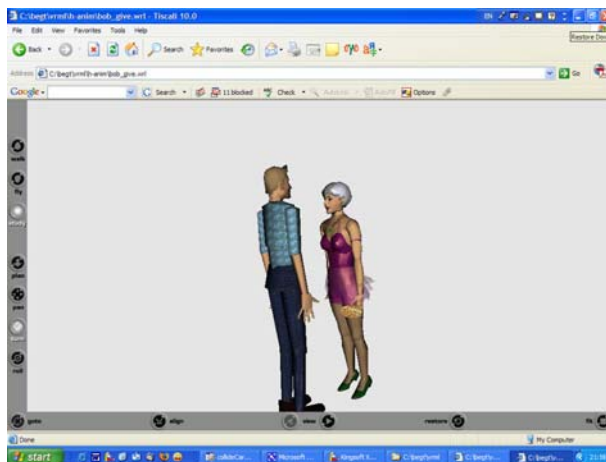
Please choose the closest word/sentence to describe the animation.

1.



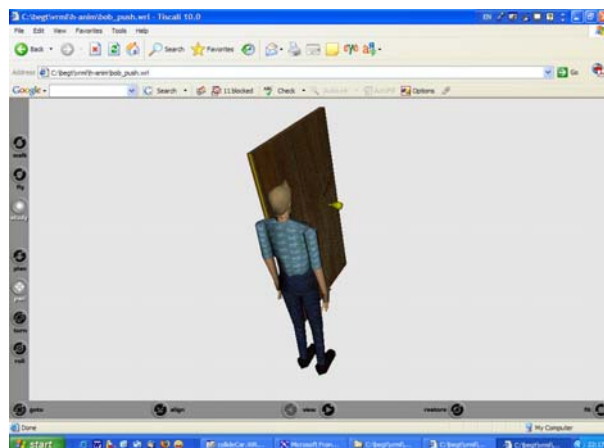
- drive
- hit
- touch
- collide

2.



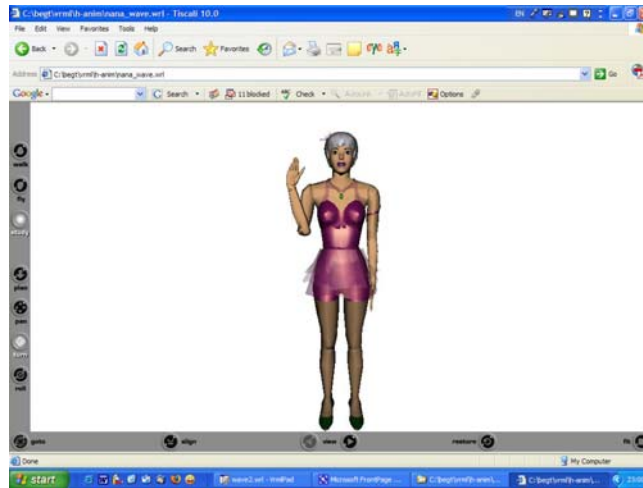
- show
- tell
- give
- sell

3.



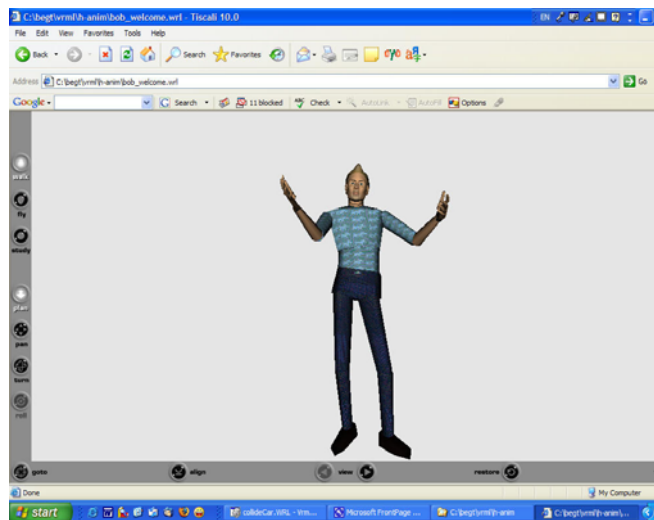
- push
- feel
- carry
- open

4.



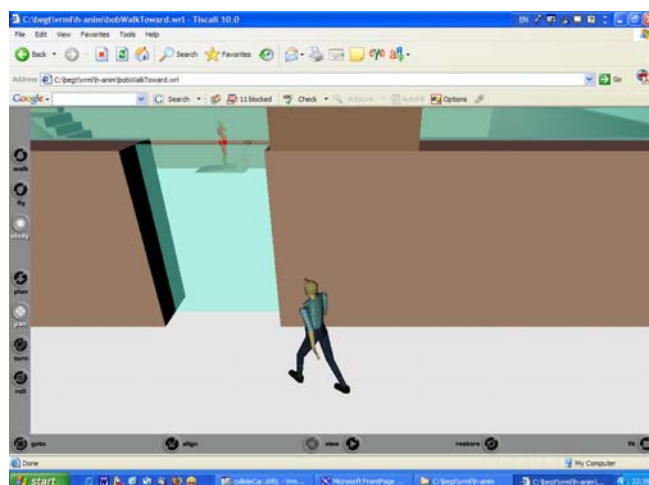
- reach
- point
- agree
- wave

5.



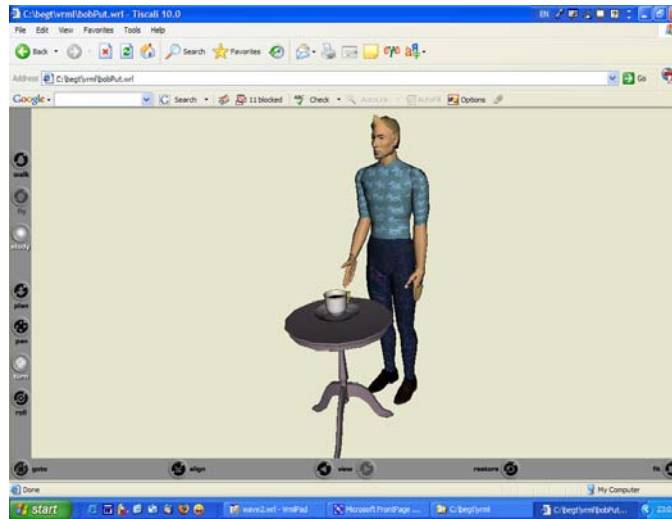
- welcome
- announce
- express
- beat

6.



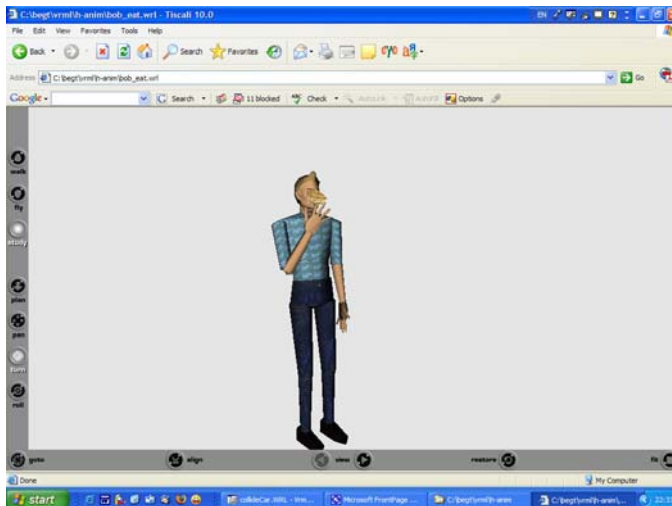
- John showed me around.
- John ran away.
- John went to the gym.
- John fell.

7.



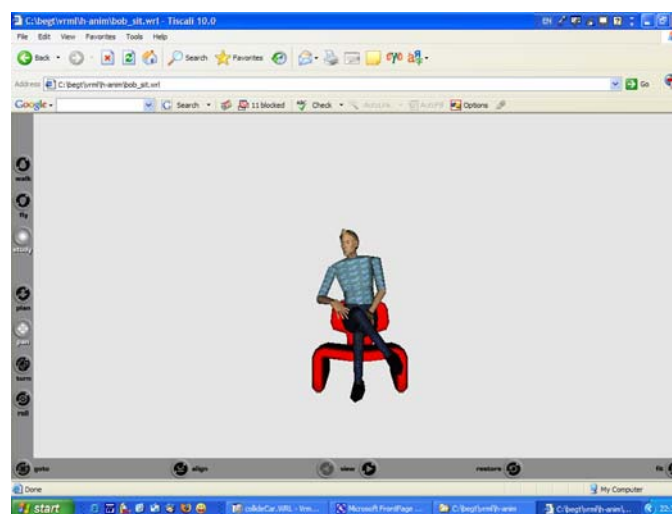
- John picked up the cup.
- John put a cup on the table.
- John served tea.
- John chose the cup.

8.



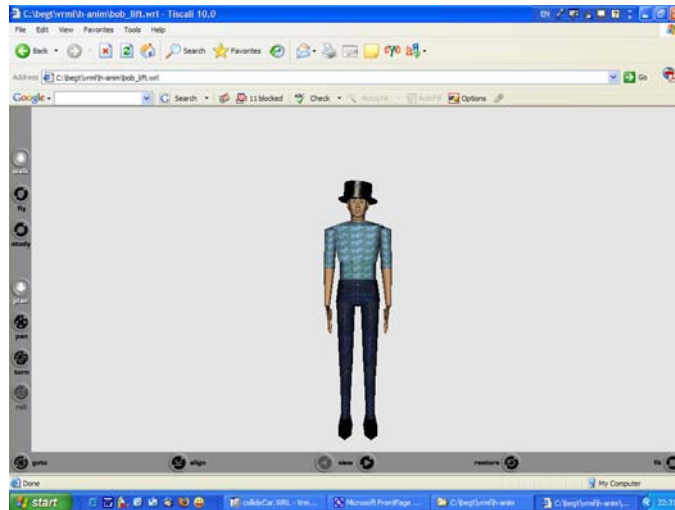
- John showed me the bread.
- John is holding the bread.
- John ate the bread.
- John offered the bread to me.

9.



- John hit the chair.
- John examined the chair.
- John pulled the chair.
- John sat on the chair.

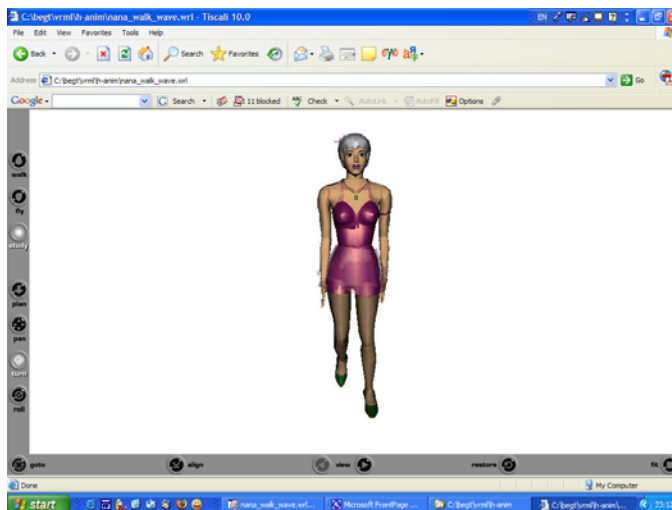
10.



- John wore a black hat.
- John lifted his black hat.
- John carried a black hat.
- John put on his black hat.

Please put down the rate next to the animation indicating if the words/sentences given express the features of the animation.

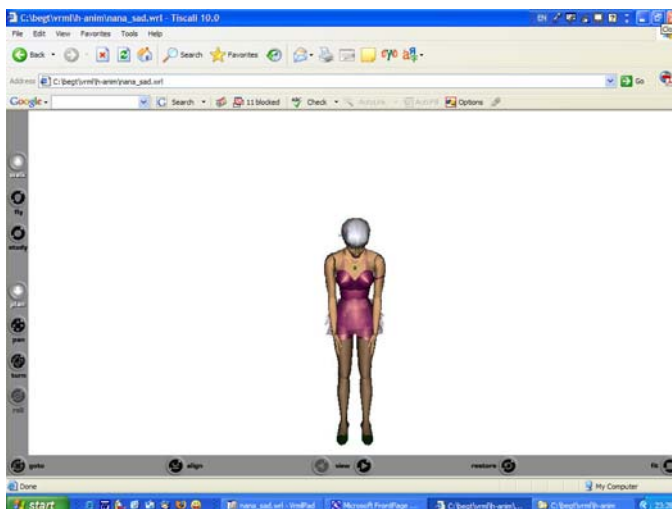
11.



walk & wave

- Excellent
- Good
- Average
- Poor
- Terrible

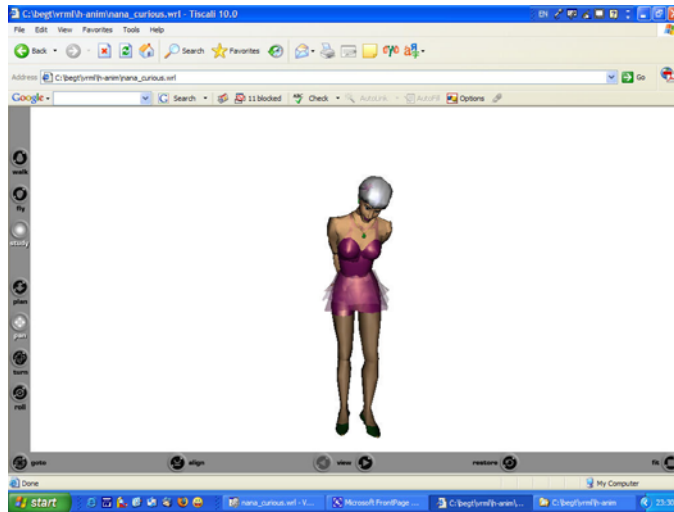
12.



ashamed

- Excellent
- Good
- Average
- Poor
- Terrible

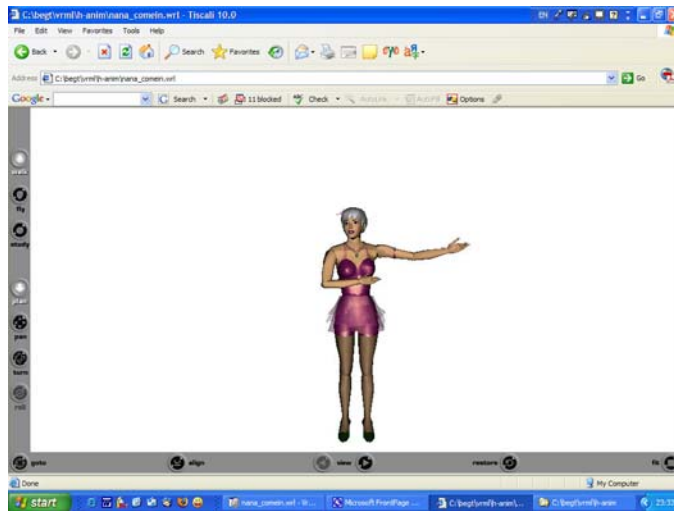
13.



curious

- Excellent
- Good
- Average
- Poor
- Terrible

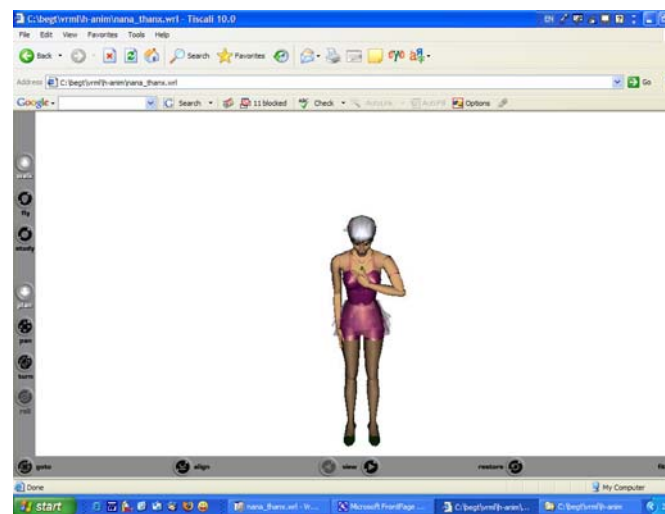
14.



guide

- Excellent
- Good
- Average
- Poor
- Terrible

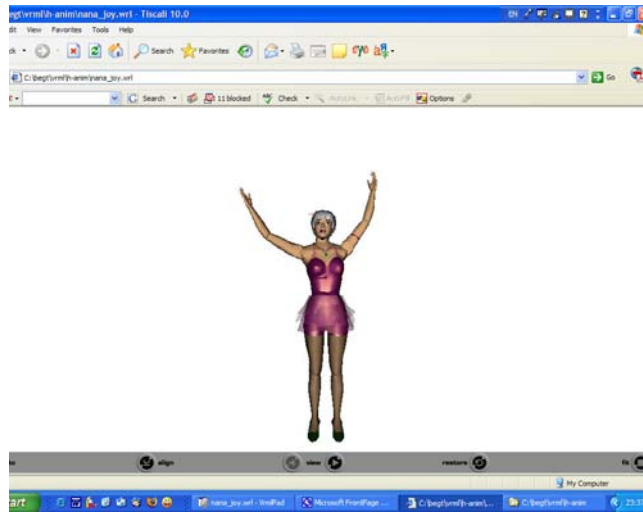
15.



Jane thanked him for his help.

- Excellent
- Good
- Average
- Poor
- Terrible

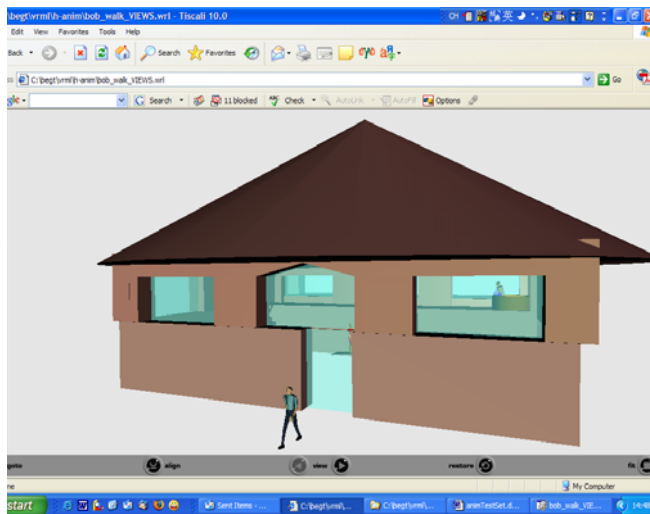
16.



Jane is happy.

- Excellent
- Good
- Average
- Poor
- Terrible

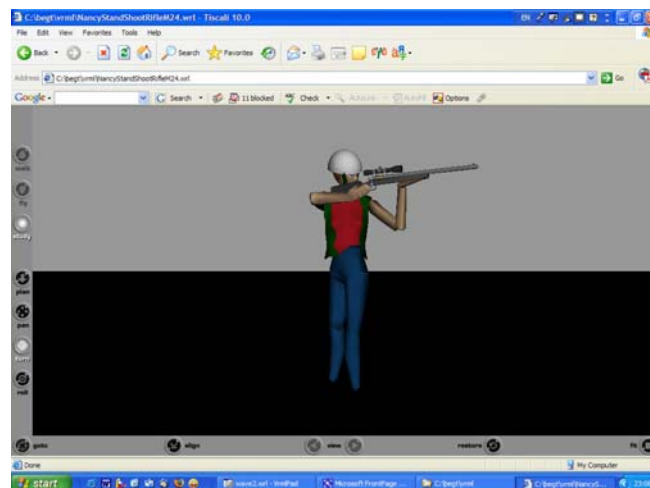
17.



John left the gym.

- Excellent
- Good
- Average
- Poor
- Terrible

18.



Jane shot the bird.

- Excellent
- Good
- Average
- Poor
- Terrible

Please give your comments on CONFUCIUS-generated animations:

⏪
⏩

⏪
⏩

Submit

Appendix H

Test Set for Syntactic Analysis (from HORATIO)

1. They failed.
2. He was eager to back down.
3. Do the facts allow the explanation he gave to the students?
4. They should back up the teacher.
5. They should back the good teachers up.
6. They should back up the teacher they like.
7. The teacher should have been backed up.
8. She must allow that John is a good teacher.
9. She must allow John is a bad teacher.
10. You must allow for the oversimplifications he has made.
11. The teacher allows the boys money for books.
12. He told her that he loved Mary.
13. She told him what to see.
14. John has alienated the students from the teacher.
15. He allowed the students into the library.
16. The students he had allowed into the library were reading books.
17. They are teachers.
18. He is reluctant to go into the library.
19. The problem is that she knows him.
20. We have been in the library.
21. He has become a good teacher.
22. The books belong in the library.
23. The girl went to the library.
24. He brought the books he had liked to the library.
25. He brought to the library the books he liked.
26. He considers the claim she has made an oversimplification.
27. They declared the claim valid.
28. They will decide where to go.
29. They did away with the bad teachers.
30. They want him to kick the bucket.
31. They should pay attention to the problems he saw.
32. Great attention was paid to the problems he had seen.
33. The students had been put at risk.
34. They took the problems he had seen into account.
35. They took into account the problems they had seen.
36. He should take them into account.
37. The workshop will take place in the library.
38. They were shooting the breeze.
39. They allowed her to teach linguistics.
40. She was allowed to teach linguistics.
41. They wanted to teach linguistics.
42. He wanted them to put the workshop off.
43. John tried to teach linguistics.
44. They persuaded her to teach linguistics.
45. She was persuaded to teach linguistics.
46. They expected her to teach linguistics.
47. She was expected to teach linguistics.
48. Mary is expected to be elected.
49. She promised to teach linguistics.
50. She promised them to teach linguistics.
51. She seems to have taught linguistics.
52. It seems that she has taught mathematics.
53. The book seems to have been read by the students.
54. The book was expected to have been read.
55. She is eager to teach.

56. She is easy to please.
57. She is an easy woman to please.
58. The teacher was seen to read a bad book.
59. The students saw John teach mathematics.
60. Teachers avoid reading books.
61. She wants to avoid their reading bad books.
62. They believed him to have killed a student.
63. He was believed to have killed a student.
64. The book seems to be read.
65. The book seems to have been read by the student.
66. The book is believed to have been read.
67. The man is believed to have read the book.
68. Mary tends to be annoyed by John.
69. John tends to annoy Mary.
70. John tries to annoy Mary.
71. Mary tries to be annoyed by John.
72. John wants to appear to be loved by Mary.
73. John appears to want to be loved by Mary.
74. When Mary saw John she told him that she wanted him to meet the teacher.
75. He warned her that she had been seen before she went to the library.
76. If he saw her he must have seen her before she went into the library.
77. The teacher who teaches linguistics is good.
78. The workshop that he wants to put off will fail.
79. The genius a book about whom he has read teaches mathematics.
80. She likes the town in which she lives.
81. She likes the town which she lives in.
82. She likes the town that she lives in.
83. She likes the town she lives in.
84. She likes the town where she lives.
85. The teacher whose books she likes thinks that she is a good student.
86. I know the university which she tells him she knows he wants her to go to.
87. Who knew that John expected her to break down?
88. What might the man have been looking at?
89. On which table has he put the books?
90. Which table has he put the books on?
91. Where did he go?
92. Have you met Mary?
93. Do I know him?
94. Are they the teachers who taught you linguistics?
95. I knew where he wanted to go.
96. You must decide which books the students should read.
97. I told him where to go.
98. He must have been told where to go.
99. Might he have been writing a book?
100. Does he believe her to have gone in for linguistics?
101. Have you read the letter to the teacher about the library?
102. The problem with you is that you know me.
103. Do you back up the decision to give him money?
104. They are easy to teach.
105. John is reluctant to teach linguistics.
106. John is black.
107. John has seen a black dog.
108. He is sure to tell them what to read.
109. He is sure I will tell them what to read.
110. Mary is an easy woman to please.
111. The man reading a book in the library is a teacher.
112. I want to read a book written by a student.
113. He went to the library with Mary.

114. Mary was reading a book about linguistics in the library.
115. The woman is reading a book in the library.
116. I am reading in the library a book the students want me to read.
117. She gave books to the students.
118. She gave the students good books.
119. She gave the students the books she wanted them to read.
120. The teacher took the problems into account.
121. The teacher took into account the problems.
122. The teacher took them into account.
123. The teacher took into account the problems the students had seen.
124. Do you like books about linguistics?
125. The man reading a book in the library is a good teacher.
126. He considers the claim she made an oversimplification.
127. The students were persuaded to read the books in the library.
128. He had been looked down on.
129. Mary has been given a book.
130. A good book has been given to Mary.
131. A book has been given to Mary by the student.
132. The students are expected to read books about linguistics.
133. The teacher was seen to read a book about women.
134. Books should be read.
135. The student was declared a genius.
136. The problems were paid attention to.
137. Great attention was paid to the problems the students had seen.
138. The books they said they liked were put in the library.
139. He had been told where to meet her.
140. He was believed to have killed a bad student.
141. The good books seem to have been read by the students.
142. The teacher whose books I told her I liked knows the university
I have persuaded her to go to.
143. The students like the books the teacher wants them to read.
144. What does the teacher think the student is learning?
145. Who is the man the woman has been looking for listening to?
146. On which table might the man have put the books?
147. Which table might the man have put the books on?
148. I decided what to tell her I believed her to like.
149. With coordination: Mary teaches linguistics and John is learning
mathematics.
150. Mary teaches linguistics and John mathematics.
151. Mary is and John wants to be in the library.
152. Mary is in and John wants to be in the library.
153. Mary went to the library and John to the workshop.
154. The teacher has been given a book and the students a library.
155. John and the students want to put off the workshop.
156. John likes dogs and black cats.
157. He looked at the teacher and the students.
158. She made a valid and true claim.
159. The teacher turned up and broke down.
160. She declares and considers him a genius.
161. She told him where to go and what to see.
162. He can and should see her.
163. He relied on and liked the students.
164. He liked and relied on the students.
165. He backed up and liked the decision to give them money.
166. He liked and backed up the decision to give them money.
167. She likes the books that I have written and you have put into
the library.
168. They had been tripping the light fantastic and shooting the
breeze.
169. They tripped the light fantastic and shot the breeze.
170. They may trip the light fantastic and shoot the breeze.

Appendix I

Test Set for Anaphora Resolution

This appendix lists the test set for anaphora resolution. It includes 20 paragraphs. The number following each paragraph denotes the number of pronoun in the paragraph. The words that JavaRAP failed to resolve are highlighted.

1. The Fish-Footman began by producing from under his arm a great letter, nearly as large as himself. 2
2. At this moment the door of the house opened, and a large plate came skimming out, straight at the Footman's head: it just grazed **his** nose, and broke to pieces against one of the trees behind **him**. 3
3. Alice did not like to be told so. `...' she muttered to herself. 2
4. `There's certainly too much pepper in that soup!' Alice said to herself, as well as she could for sneezing. There was certainly too much of **it** in the air. Even the Duchess sneezed occasionally; and as for the baby, **it** was sneezing and howling alternately without a moment's pause. 4
5. "Please would you tell me," said Alice, a little timidly, for **she** was not quite sure whether it was good manners for **her** to speak first, "why your cat grins like that?"
"It's a Cheshire cat," said the Duchess, "and that's why. Pig!"
She said the last word with such sudden violence that Alice quite jumped; but she saw in another moment that it was addressed to the baby, and not to her, so she took courage, and went on again. 9
6. Alice did not at all like the tone of this remark, and thought **it** would be as well to introduce some other subject of conversation. While she was trying to fix on one, the cook took the cauldron of soup off the fire, and at once set to work throwing everything within her reach at the Duchess and the baby--the fire-irons came first; then followed a shower of saucepans, plates, and dishes. The Duchess took no notice of them even when they hit her; and the baby was howling so much already, that it was quite impossible to say whether the blows hurt it or not. 8
7. Alice glanced rather anxiously at the cook, to see if **she** meant to take the hint; but the cook was busily stirring the soup, and seemed not to be listening, so **she** went on again. 2
8. Alice caught the baby with some difficulty, as it was a queer-shaped little creature, and held out its arms and legs in all directions. The poor little thing was snorting like a steam-engine when **she** caught it, and kept doubling itself up and straightening itself out again, so that altogether, for the first minute or two, **it** was as much as **she** could do to hold it. 9
9. The baby grunted again, and Alice looked very anxiously into its face to see what was the matter with it. There could be no doubt that **it** had a very turn-up nose, much more like a snout than a real nose; also **its** eyes were getting extremely small for a baby. 4
10. Alice had quite forgotten the duchess by this time, and was a little startled when she heard **her** voice close to her ear. 3
11. The duchess squeezed herself up closer to Alice's side as she spoke. Alice did not much like keeping so close to **her**: first, because the duchess was very ugly; and secondly, because she was exactly the right height to rest her chin upon Alice's shoulder,

- and **it** was an uncomfortably sharp chin. However, **she** did not like to be rude, so **she** bore **it** as well as **she** could. 10
12. All this time Tweedledee was trying his best to fold up the umbrella, with himself in **it**: which was such an extraordinary thing to do, that **it** quite took off Alice's attention from the angry brother. But he couldn't quite succeed, and **it** ended in his rolling over, bundling up in the umbrella, with only his head out: and there he lay, opening and shutting his mouth and his large eyes. 11
13. So the two brothers went off hand-in-hand into the wood, and returned in a minute with their arms full of things--such as bolsters, blankets, hearth-rugs, table-cloths, dish-covers, and coal-scuttles. Alice said afterwards she had never seen such a fuss made about anything in all her life--the way those two bustled about--and the quantity of things they put on--and the trouble they gave **her** in tying strings and fastening buttons. 6
14. Alice laughed loud: but she managed to turn **it** into a cough. 2
15. The White Queen only looked at Alice in a helpless frightened sort of way, and kept repeating something in a whisper to herself that sounded like "Bread-and-butter, bread-and-butter", and Alice felt that if there was to be any conversation at all, **she** must manage **it herself**. So **she** began rather timidly: 5
16. Alice was just beginning to say *** when the Queen began screaming, so loud that **she** had to leave the sentence unfinished. *** shouted the Queen, shaking **her** hand about as if **she** wanted to shake **it** off. 4
17. Alice looked at the Queen, who seemed to have suddenly wrapped **herself** up in wool. Alice rubbed her eyes, and looked again. She couldn't make out what had happened at all. Was she in a shop? And was that really -- was **it** really a *sheep* that was sitting on the other side of the counter? Rub as **she** would, **she** could make nothing more of **it**: **she** was in a little dark shop, leaning with **her** elbows on the counter, and opposite to **her** was an old Sheep, sitting in an arm-chair, knitting, and every now and then leaving off to look at **her** through a great pair of spectacles. 12
18. Alice had *not* got: so she contented herself with turning round, looking at the shelves as she came to them. 4
19. Alice said nothing, but pulled away. There was something very queer about the water, **she** thought, as every now and then the oars got fast in **it**, and would hardly come out again. 2
20. The Sheep took the money, and put it away in a box: then she said ... 2

Appendix J

Test Set for Semantic Analysis of Verbs

1. 34 333518 say v
“Well, I'd hardly finished the first verse,” said the Hatter.
2. 40 249540 go v
We quarrelled last March, just before he went mad.
3. 44 220940 get v
The Hatter got any advantage from the change.
4. 46 217268 make v
(The Dormouse began singing in its sleep `Twinkle, twinkle, twinkle, twinkle--' and went on so long that) they had to pinch it to make it stop.
5. 51 191661 see v
(The last time) she saw them, (they were trying to put the Dormouse into the teapot.)
6. 52 185534 know v
I didn't know it was your table.
7. 54 179220 take v
The March Hare took the watch (and looked at it gloomily: then he dipped it into his cup of tea, and looked at it again.)
8. 64 153881 think v
“That's very curious!” she thought. “But everything's curious today. I think I may as well go in at once.”
9. 66 151871 come v
A bright idea came into Alice's head.
10. 76 131417 give v
The great concert is given by the Queen.
11. 90 111058 look v
She looked back (once or twice, half hoping that they would call after her.)
12. 92 108820 use v
I'll try if I know all the things I used to know.
13. 100 98899 find v
She found that she was now about two feet high.
14. 103 94293 want v
They won't walk the way I want to go.
15. 116 77245 tell v
(Let us get to the shore, and then) I'll tell you my history.
16. 125 69978 put v
The Caterpillar put the hookah into its mouth (and began smoking again.)

17. 129 67842 work v

(She felt that there was no time to be lost, as she was shrinking rapidly; so) she set to work at once to eat some of the other bit.

18. 130 67219 become v

(You are old, Father William, and) your hair has become very white;

19. 134 66556 mean v

(Alice glanced rather anxiously at the cook, to see if) she meant to take the hint.

20. 136 64447 leave v

She left it behind.

21. 143 62445 seem v

The cook was busily stirring the soup, and seemed not to be listening

22. 148 62185 feel v

Alice began to feel very uneasy.

23. 154 60879 ask v

'Did you say "What a pity!"?' the Rabbit asked.

24. 163 58152 show v

She showed off her knowledge.

25. 174 54422 try v

Alice tried every door. (She walked sadly down the middle, wondering how she was ever to get out again.)

26. 175 53396 call v

So she called softly after it, ('Mouse dear! Do come back again, and we won't talk about cats or dogs either, if you don't like them!')

27. 189 50092 keep v

(Alice took up the fan and gloves, and, as the hall was very hot,) she kept fanning herself all the time she went on talking.

28. 197 47234 hold v

(After a while she remembered that) she still held the pieces of mushroom in her hands.

29. 203 46145 follow v

(He moved on as he spoke, and) the Dormouse followed him.

30. 205 45487 turn v

She turned to the Dormouse (and repeated her question).

31. 211 43894 bring v

(By the time) she had caught the flamingo and brought it back, (the fight was over.)

32. 214 43740 begin v

The Dormouse began singing in its sleep.

33. 221 41909 like v

Sure, I don't like it.

34. 223 41497 write v

The jury eagerly wrote down all three dates on their slates.

35. 229 40858 run v

(Alice got up and ran off, thinking while) she ran, (as well she might, what a wonderful dream it had been.)

36. 232 40381 set v

They set to work very diligently to write out a history of the accident.

37. 233 40265 help v

She helped herself to some tea and bread-and-butter.

38. 245 38053 play v

(She remembered trying to box her own ears for having cheated herself in a game of croquet) she was playing against herself.

39. 249 37836 move v

“Off with her head!” the Queen shouted at the top of her voice. Nobody moved.

40. 258 36575 hear v

(Because they're making such a noise inside,) no one could possibly hear you.

Appendix K

Publications

List of publications resulting directly from the thesis is presented in this appendix.

Journal/book papers

1. Ma, Minhua and P. McKeivitt (2006) Virtual Human Animation in Natural Language Visualisation. *Special Issue on Research in Artificial Intelligence and Cognitive Science, Artificial Intelligence Review*, Dordrecht, The Netherlands: Kluwer-Academic Publishers. (Accepted)
2. Ma, Minhua and P. McKeivitt (2004) Visual semantics and ontology of eventive verbs. Natural Language Processing - IJCNLP-04, First International Joint Conference, Keh-Yih Su, Jun-Ichi Tsujii, Jong-Hyeok Lee and Oi Yee Kwong (Eds.), 187-196, *Lecture Notes in Artificial Intelligence* (LNAI) series, LNCS 3248. Berlin, Germany: Springer Verlag. Also published in *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, Keh-Yih Su and Jun-Ichi Tsujii (Eds.), 278-285, Resort Golden Palm, Sanya, China, March. (awarded one of best 3 IJCNLP papers).
3. Ma, Minhua and P. McKeivitt (2004) Interval relations in visual semantics of verbs. *Special Issue on Research in Artificial Intelligence and Cognitive Science, Artificial Intelligence Review* (21): 293-316, Dordrecht, The Netherlands: Kluwer-Academic Publishers.

Conference/Workshop papers

4. Ma, Minhua and P. McKeivitt (2005) Presenting Temporal Relations of Virtual Human Actions by Multiple Animation Channels. In *Proc. of the 16th Irish Conference on Artificial Intelligence & Cognitive Science (AICS-05)*, N. Creaney (Ed.), 169-178, Flowerfield Arts Centre, Portstewart, University of Ulster, Northern Ireland, 7-9 September.
5. Ma, Minhua and P. McKeivitt (2005) Lexical Semantics and Auditory Display in Virtual Storytelling. In *Proc. of the 11th International Conference on Auditory Display 2005 (ICAD05)*, E. Brazil (Ed.), 358-363, University of Limerick, Limerick, Ireland, 6-9 July.
6. Ma, Minhua and P. McKeivitt (2005) Animating Virtual Humans in Intelligent Multimedia Storytelling. In *Proc. of the 6th Annual PGNET Conference: The convergence of telecommunications, networking and broadcasting (PGNET 2005)*, M. Merabti and R. Pereira (Eds.), 159-164, Liverpool John Moores University, Liverpool, England, June.
7. Ma, Minhua and P. McKeivitt (2004) Using lexical knowledge of verbs in language-to-vision applications. In *Proc. of the 15th Artificial Intelligence and Cognitive Science Conference (AICS-04)*, L. McGinty and B. Crean (Eds.), 255-264, Galway-Mayo Institute of Technology, Castlebar, Ireland, September.
8. Ma, Minhua and P. McKeivitt (2003) Building character animation for intelligent storytelling with the H-Anim standard. In *Eurographics Ireland Chapter Workshop Proceedings 2003*, M. McNeill (Ed.), 9-15, Coleraine, University of Ulster, Northern Ireland, April.
9. Ma, Minhua and P. McKeivitt (2003) Semantic representation of events in 3D animation. In *Proceedings of The Fifth International Workshop on Computational Semantics (IWCS-5)*, H. Bunt, I. van der Sluis and R. Morante (Eds.), 253-281, Tilburg, The Netherlands, January.

References

- 3D Studio Max (2005) 3D Studio Max. <http://www.autodesk.com/3dsmax>, site visited 19/03/2006.
- Alexa, M., J. Behr, W. Miller (2000) *The Morph Node*. In *Proceedings of Web3d/VRML 2000*, Monterey, CA., 29-34.
- Allen, J. F. (1983) Maintaining knowledge about temporal intervals. In *Communications of the ACM*, 26(11): 832-843.
- Allen, J. F. and G. Ferguson (1994) Actions and events in interval temporal logic. In *Journal of Logic and Computation*, 4(5): 531-579.
- André, E. and T. Rist (2000) Presenting Through Performing: On the Use of Multiple Lifelike Characters in Knowledge-Based Presentation Systems. In *Proceedings of the Second International Conference on Intelligent User Interfaces (IUI 2000)*, Los Angeles, 1-8.
- Arens, Y. and E. Hovy (1995) The Design of a Model-based Multimedia Interaction Manager. In *Integration of Natural Language and Vision Processing, Vol II, Intelligent Multimedia*, P. Mc Kevitt (Ed.), 95-115. Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Arnold, D. J., L. Balkan, S. Meijer, R. L. Humphreys, and L. Sadler (1994) *Machine Translation: an Introductory Guide*. London: Blackwells-NCC.
- Asher, N. and A. Lascarides (1995) Lexical Disambiguation in a Discourse Context. *Journal of Semantics*, 12(1): 69-108.
- Babski, C. (2000) *Humanoids on the Web*. Ph.D. Thesis, Computer Graphics Lab (LIG), Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
- Badler, N. (1997) Virtual humans for animation, ergonomics, and simulation. In *IEEE Workshop on Non-Rigid and Articulated Motion*, Puerto Rico, June, 28-37.
- Badler, N., B. Webber, J. Kalita, and J. Esakov (1991) Animation from Instructions. In *Making them Move*, N. Badler, B. Barsky, and D. Zeltzer (Eds.), 51-93, Cambridge, MA: MIT Press.
- Badler N., C.B. Philips, and B.L. Webber (1993) *Simulating Humans*, Oxford, U.K.: Oxford University Press.
- Badler, N., B. Webber, M. Palmer, T. Noma, M. Stone, J. Rosenzweig, S. Chopra, K. Stanley, H. Dang, R. Bindiganavale, D. Chi, J. Bourne and B. Di Eugenio (1997) Natural language text generation from Task networks. Technical Report, CIS, University of Pennsylvania, Philadelphia, U.S.A.
- Baecker, R., I. Small, and R. Mander (1991) Bringing Icons to Life. In *Proceedings ACM CHI'91*, New Orleans, U.S.A., 1-6.
- Bailey, D., J. Feldman, S. Narayanan and G. Lakoff (1997) Modeling embodied lexical development. *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society (CogSci97)*, Stanford, CA, U.S.A., 19-24.
- Baker, C.F., C.J. Fillmore, and J.B. Lowe (1998) The Berkeley FrameNet project. In *Proceedings of the COLING-ACL*, Montreal, Canada.
- Ballreich, C. (1997) Nancy - 3D Model. 3Name3D. http://www.ballreich.net/vrml/h-anim/nancy_h-anim.wrl Site visited 23/10/2005.

- Beckwith, R., C. Fellbaum, D. Gross and G.A. Miller (1991) WordNet: A lexical Database Organized on Psycholinguistic Principles. In *Lexicons: Using On-line Resources to Build a Lexicon*, U. Zernik (Ed.), 211-231, Hillsdale, NJ: Lawrence Erlbaum.
- Bergen, B., S. Narayan, and J. Feldman (2003) Embodied verbal semantics: evidence from an image-verb matching task. *Proceedings of CogSci*, Boston ParkPlaza Hotel, Boston.
- Bishop, C.M. and M.E. Tipping (1998) A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3), 281-293.
- Blaxxun Contact (2001) <http://developer.blaxxun.com/> Site visited 08/01/2006.
- Bly, S., S.P. Frysinger, D. Lunney, D.L. Mansur, J. Mezrich and R. Morrison (1987) Communicating with Sound. In *Readings in Human-Computer Interaction: A Multi-disciplinary Approach*, R. Baecker and W. Buxton (Eds.), 420-424, Los Altos: Morgan-Kaufman.
- BNC (2004) The British National Corpus. <http://www.natcorp.ox.ac.uk/> Site visited 17/08/2005.
- Bobick, A., S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schtte and A. Wilson (1996) The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment. In *PRESENCE: Teleoperators and Virtual Environments*, 8(4): 367-391.
- Bobrow, D. and T. Winograd (1985) An Overview of KRL, a Knowledge Representation Language. In *Readings in Knowledge Representation*, R.J. Brachman and H.J. Levesque (Eds.), 263-285, California, U.S.A.: Morgan Kaufmann.
- Bolt, R.A. (1987) Conversing with Computers. In *Readings in Human-Computer Interaction: A Multidisciplinary Approach*, R. Baecker and W. Buxton (Eds.), California, U.S.A.: Morgan Kaufmann.
- Brachman, R. and J. Schmolze (1985) An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9(2): 171-216.
- Brøndsted, T. (1999) The CPK NLP Suite for Spoken Language Understanding. *Eurospeech, 6th European Conference on Speech Communication and Technology*, Budapest, September, 2655-2658.
- Brøndsted, T., P. Dalsgaard, L.B. Larsen, M. Manthey, P. Mc Kevitt, T.B. Moeslund and K.G. Olesen (2001) The IntelliMedia WorkBench - An Environment for Building Multimodal Systems. In *Advances in Cooperative Multimodal Communication: Second International Conference, CMC'98, Tilburg, The Netherlands, January 1998, Selected Papers*, Harry Bunt and Robbert-Jan Beun (Eds.), 217-233. *Lecture Notes in Artificial Intelligence (LNAI) series*, LNAI 2155, Berlin, Germany: Springer Verlag.
- Burger, J., and R. Marshall (1993) The Application of Natural Language Models to Intelligent Multimedia. In *Intelligent Multimedia Interfaces*, M. Maybury (Ed.), 167-187, Menlo Park: AAAI/MIT Press.
- Buxton, W., S. Bly, S. Frysinger, D. Lunney, D. Mansur, J. Mezrich, and R. Morrison (1985) Communicating with Sound. In *Proceedings of Human Factors in Computing Systems (CHI-85)*, New York, 115-119.
- Cadoz, C. (1994) *Les realites virtuelles*. Paris, France: Dominos-Flammarion.
- Cassell, J., C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone (1998) Animated conversation: rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Readings in intelligent user interfaces*, M. Maybury and W. Wahlster (Eds.), 582-591, San Francisco, CA. U.S.A.: Morgan Kaufmann Publishers, Inc.
- Cassell, J., J. Sullivan, S. Prevost, and E. Churchill (Eds.) (2000) *Embodied Conversational Agents*. Cambridge, MA: MIT Press.

- Cassell, J., H. Vilhjalmsson and T. Bickmore (2001) BEAT: the Behavior Expression Animation Toolkit, Computer Graphics Annual Conference, *SIGGRAPH 2001 Conference Proceedings*, Los Angeles, Aug 12-17, 477-486.
- Cavazza, M., R. Earnshaw, N. M. Thalmann and D. Thalmann (1998) Motion Control of Virtual Humans, *IEEE Computer Graphics and Applications*, 18(5): 24-31.
- Church, K. (1988) A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, Morristown, NJ, 136-143.
- Collins, M. (1999) *Head-driven statistical models for natural language parsing*. Ph.D. Thesis, Computer Science Department, University of Pennsylvania, Philadelphia, PA.
- Connexor (2003) Connexor Machine Syntax http://www.connexor.com/m_syntax.html Site visited 07/10/2004.
- Cosmo player (2001) http://www.sgi.com/products/evaluation/6.5_cosmoplayer_2.1.5/ Site visited 08/01/2006.
- Coyne, B and R. Sproat (2001) WordsEye: An Automatic Text-to-Scene Conversion System. Computer Graphics Annual Conference, *SIGGRAPH 2001 Conference Proceedings*, Los Angeles, 12-17 Aug., 487-496.
- CSLU (2002) <http://cslu.cse.ogi.edu/toolkit/index.html> Site visited 12/18/2005.
- Cunningham, H., D. Maynard, K. Bontcheva, V. Tablan, C. Ursu, and M. Dimitrov (2002) Developing Language Processing Components with GATE (a User Guide) For GATE version 2.0. Technical Report, User Guide, University of Sheffield, <http://gate.ac.uk/sale/tao/index.html> Site visited 14/08/2002.
- Cyc (1997) Cyc Ontology Guide: Introduction. <http://www.cyc.com/cyc-2-1/intro-public.html> Site visited 14/08/2002.
- Dalal, M., S. Feiner, K. McKeown, S. Pan, M. Zhou, T. Hollerer, J. Shaw, Y. Feng and J. Fromer (1996) Negotiation for Automated Generation of Temporal Multimedia Presentations. In *Proceedings of ACM Multimedia Conference*, Boston, 55-64, Boston: ACM Press.
- DAML_OIL (2001) DAML+OIL Reference Description. <http://www.w3.org/TR/daml+oil-reference> Site visited 23/09/2002.
- Depraetere, I. (1995) On the necessity of distinguishing between (un)boundedness and (a)telicity. *Linguistics and Philosophy* 18, 1-19.
- Dijkstra, E. W. (1971) Hierarchical ordering of sequential processes. In *Acta Informatica* 1(2): 115-138.
- DirectX (2002) Microsoft DirectX: Multimedia technology for Windows-based gaming and entertainment. <http://www.microsoft.com/windows/directx/default.asp> Site visited 25/09/2002.
- Dixon, R.M.W. (1991) *A new approach to English Grammar on semantic principles*. Oxford, U.K.: Oxford University Press.
- Dorr, Bonnie J. and D. Jones (1999) Acquisition of Semantic Lexicons: using word sense disambiguation to improve precision. In *Breadth and Depth of Semantic Lexicons*, Evelyne Viegas (Ed.), 79-98, Norwell, MA: Kluwer Academic Publishers.
- Dorr, B. J. and M. B. Olsen (1997) Deriving Verbal and Compositional Lexical Aspect for NLP Applications. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, Madrid, Spain, 7-12 Jul., 151-158.
- Dowty, D. (1979) *Word Meaning and Montague Grammar*. Dordrecht, The Netherlands: Reidel Publishing Company.
- Dowty, D. (1991) Thematic proto-roles and argument selection. *Language*, 67(3): 547-619.

- Dupuy S., A. Egges, V. Legendre, and P. Nugues (2001) Generating a 3D Simulation of a Car Accident From a Written Description in Natural Language: The CarSim System. In *Proc. of The Workshop on Temporal and Spatial Information Processing*, 1-8, ACL 2001 Conference, Toulouse, 7 July.
- Elliott, R., J.R.W. Glauert, J.R. Kennaway, and I. Marshall (2000) The Development of Language Processing Support for the ViSiCAST Project. In Proceedings of 4th International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2000), Washington, November.
- EMMA W3C Working Draft (2005) <http://www.w3.org/TR/emma/#s1.1>, site visited 22/07/2006.
- EPFL-VRLab (2004) The Virtual Reality Lab at the Swiss Federal Institute of Technology (EPFL). <http://ligwww.epfl.ch/> Site visited 9/4/2005.
- Esmerado, J., F. Vexo and D. Thalmann (2002) Interaction in the Virtual Worlds: Application to Music Performers, Computer Graphics International.
- Ethier, S.J., Ethier, C.A. (2002) 3D Studio MAX in Motion: Basics Using 3D Studio MAX 4.2. Prentice Hall.
- Fabian, P. and J. Francik (2001) Synthesis and presentation of the Polish sign languages. In *Proceedings of 1st International Conference on Applied Mathematics and Informatics*, University of Gabčíkovo, Slovakia, 6-7 September, 190-197.
- Feiner, S. (1985) APEX: An Experiment in the Automated Creation of Pictorial Explanations. *IEEE Computer Graphics and Application* 5(11): 29-37.
- Feiner, S.K. and K.R. McKeown (1991) Automating the Generation of Coordinated Multimedia Explanations. *IEEE Computer*, 24(10): 33-41.
- Feiner, S., Mackinlay, J. and Marks, J. (1992) Automating the Design of Effective Graphics for Intelligent User Interfaces. Tutorial Notes. Human Factors in Computing Systems, CHI-92, Monterey.
- Fellbaum, C. (Ed.) (1998) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.
- Fernando, T. (2003) Finite-state descriptions for temporal semantics. In H. Bunt, I. van der Sluis and R. Morante (Eds.), *Proceedings of The Fifth International Workshop on Computational Semantics (IWCS-5)*, 122-136, Tilburg, The Netherlands, January.
- FestVox (2003) <http://festvox.org>, site visited 17/11/2005.
- Fillmore, C. J. (1968) The case for case. In *Universals in Linguistic Theory*, E. Bach and R. Harms (Eds.), New York: Holt, Rinehart and Winston, 10-88.
- FreeTTS (2004) <http://freetts.sourceforge.net/docs/index.php>, site visited 05/04/2005.
- Gaver, W. (1986) Auditory Icons: Using Sound in Computer Interfaces. *Human-Computer Interaction*, (2): 167-177.
- Gaver, W. (1989) The SonicFinder: An Interface that Uses Auditory Icons. *Human-Computer Interaction*, (4): 67-94.
- Granstrom, B., D. House and I. Karlsson (2002) *Multimodality in language and speech systems*. London, U.K.: Kluwer Academic Publishers.
- Gross, D. and K. Miller (1990) Adjectives in WordNet. *International Journal of Lexicography* 3(4): 265-277.
- Grover, C., J. Carroll and E. Briscoe (1993) The Alvey Natural Language Tools Grammar 4th Release. Technical Report, Cambridge University Computer Laboratory: Cambridge, England. <http://www.cl.cam.ac.uk/Research/NL/anlt.html#PARSE> Site visited 14/08/2002.

- Gustavsson, C., S. Beard, L. Strindlund, Q. Huynh, E. Wiknertz, A. Marriott and J. Stallo (Eds.) (2001) Working Draft of the Virtual Human Markup Language Specification. <http://www.vhml.org/>
- Gutierrez, M., F. Vexo and D. Thalmann (2004) Reflex Movements for a Virtual Human: a Biology Inspired Approach, *Proceedings of the 3rd Hellenic Conference on Artificial Intelligence*, Special Session on Intelligent Virtual Environments, May, Samos, Greece, *Lecture Notes in Artificial Intelligence*, Springer Verlag, 525-534.
- Halliday, M.A.K. (1985) *An Introduction to Functional Grammar*. London: Edward Arnold.
- Halpern, J. Y. and Shoham, Y. (1991) A propositional modal logic of time intervals. In *Journal of ACM*, 38(4): 935-962.
- H-Anim (2001) Humanoid animation working group. <http://www.h-anim.org>, site visited 26/09/2005.
- Havok Physics (2006) <http://www.havok.com/content/view/17/30>, site visited 14/07/2006.
- Hayes-Roth B. and R. van Gent (1997) Story-making with improvisational puppets. In *Proceedings of the First International Conference on Autonomous Agents*, 1-7, Marina Del Rey, CA. U.S.A.
- Hepple, M. (2000) Independence and commitment: assumptions for rapid training and execution of rule-based POS taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong.
- Hirschman, L. and Thompson, H.S. (1995) Chapter 13. Evaluation: Overview of Evaluation in Speech and Natural Language Processing. In R.A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press.
- Huang, X., Acero, A., Adcock, J., Hon, H.-W., Goldsmith, J., Liu, J. and Plumpe, M. (1996) Whistler: A trainable Text-to-Speech system. *Proceedings 4th International Conference on Spoken Language Processing (ICSLP '96)*, Piscataway, NJ, 2387-2390.
- Huang, Z, A. Eliens and C. Visser (2003) Implementation of a scripting language for VRML/X3D-based embodied agents. *Proceeding of the Eighth International Conference on 3D Web technology*, 91 – 100, Saint Malo, France.
- Jackendoff, R. (1987) On Beyond Zebra: The Relation of Linguistic and Visual Information. *Cognition*, 26(2), 89-114.
- Jackendoff, R. (1990) *Semantic Structures*. Current studies in linguistics series, Cambridge, MA: MIT Press.
- Jackendoff, R. (1991) Parts and Boundaries, *Cognition* 41, 9-45.
- Järvinen, J. M. Laari, T. Lahtinen, S. Paaajanen, P. Paljakka, M. Soininen, and P. Tapanainen (2004) Robust language analysis components for practical applications. Coling: Satellite Workshop Robust and Adaptive Information Processing for Mobile Speech Interfaces, Geneva, Switzerland, August, 2004.
- JSAPI (2002) <http://java.sun.com/products/java-media/speech/> Site visited 25/09/2002.
- Jurafsky (1992) An On-line Computational Model of Human Sentence Interpretation. Technical Report UCB/CSD 92/767, Dept. of Computer Science, University of California, Berkeley, CA.
- Jurafsky, D.S. and J.H. Martin (2000) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. New Jersey, U.S.A.: Prentice Hall, Inc.
- Justeson, J. S. and S. M. Katz (1993) Principled disambiguation: Discriminating adjective senses with modified nouns. In *Making Sense of Words, Proceedings of the 9th Annual Conference of the UW Centre for the new OED and Text Research*, 57-73, Oxford, England, September.

- Kallmann, M. and D. Thalmann (2002) *Modeling Behaviors of Interactive Objects for Real Time Virtual Environments*. *Journal of Visual Languages and Computing*, 13(2):177-195.
- Kautz, H. A. and P. Ladkin (1991) Integrating metric and qualitative temporal reasoning. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, 241-246, Anaheim, California, USA, July.
- Kelleher, J., T. Doris, Q. Hussain and S. Ó Nualláin (2000) SONAS: Multimodal, Multi-user Interaction with a Modelled Environment. In *Spatial Cognition*, S. Ó Nualláin (Ed.), 171-184, Philadelphia, U.S.A.: John Benjamins B.V.
- Kingsbury, P., Palmer, M. (2002) From Treebank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Spain.
- Kipper, K., Dang, H.T., Palmer, M. (2000) Class-Based Construction of a Verb Lexicon. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX, USA.
- Kosslyn, S.M. (1994) *Image and Brain: The Resolution of the Imagery Debate*. Cambridge, MA: MIT Press.
- Kshirsagar, S., N. M. Thalmann, A. Guye-Vuilleme, D. Thalmann, K. Kamyab, and E. Mamdani (2002) Avatar Markup Language, *Proceedings of Eurographics Workshop on Virtual Environments*, 169-177.
- Laird J.E. (2001) Using a Computer Game to Develop Advanced AI. *Computer*, 34(7), 70-75.
- Larsen, C.B. and B.C. Petersen (1999) *Interactive Storytelling in a Multimodal Environment*. Technical Report, M.Sc. Thesis, Institute of Electronic Systems, Aalborg University, Denmark.
- Lee, Mark. G. (1994) A Model of Story Generation. M.Sc. Thesis, Dept. of Computer Science, University of Manchester.
- Leech, G. (1981) *Semantics*. Cambridge University Press.
- Lemoine, P., F. Vexo and D. Thalmann (2003) Interaction Techniques: 3D Menus-based Paradigm, *AVIR2003*, Geneva.
- Lenat, D. B. (1995) CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of ACM*, 38(11): 33-38.
- Levin, B. (1993) *English verb classes and alternations: a preliminary investigation*. Chicago: The University of Chicago Press.
- Loyall, A. B. (1997) *Believable agents: building interactive personalities*. Ph.D. thesis, CMU-CS-97-123, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.
- Macleod, C., R. Grishman and A. Meyers (1998) COMLEX syntax reference manual version 3.0. Linguistic Data Consortium.
- Mann, W.C., C.M. Matthiessen and S.A. Thompson (1992) Rhetorical Structure Theory and Text Analysis. In *Discourse Description: Diverse linguistic analyses of a fund-raising text*, W.C. Mann and S.A. Thompson (Eds.), 39-78, Amsterdam: John Benjamins.
- Manuel, D. (1994) *The Use of Art Media Techniques in Computer Visualisations and the Creation of Partially Automated Visualisation Systems*. Master's thesis, University of Exeter, Department of Computer Science, Exeter, EX4 4PT, U.K.
- Marks, J. and Reiter, E (1990) Avoiding Unwanted Conversational Implicatures in Text and Graphics. In *Proceedings of AAAI-90*, Boston, MA, Vol.1, 450-456.
- Marr, D. (1982) *Vision*. San Francisco, U.S.A.: W.H. Freeman.
- Maya (2005) <http://www.alias.com>, site visited 19/03/2006.
- Maybury, M.T. (Ed.) (1993) *Intelligent Multimedia Interfaces*. Menlo Park: AAAI/MIT Press.

- Maybury, M.T. (1994) Research in Multimedia Parsing and Generation. In *Integration of Natural Language and Vision Processing (Volume II): Intelligent Multimedia*, P. Mc Kevitt (Ed.), 31-55, London, U.K.: Kluwer Academic Publishers.
- Maybury, M.T. and W. Wahlster (Eds.) (1998) *Readings in Intelligent User Interfaces*. San Francisco, CA.: Morgan Kaufmann Press.
- McConnel, S. (1996) KTEXT and PC-PATR: Unification based tools for computer aided adaptation. In H. A. Black, A. Buseman, D. Payne and G. F. Simons (Eds.), *Proceedings of the 1996 general CARLA conference*, November 14-15, 39-95. Waxhaw, NC/Dallas: JAARS and Summer Institute of Linguistics.
- Mc Kevitt, P. (Ed.) (1995, 1996) *Integration of Natural Language and Vision Processing (Vols I-IV)*. London, U.K.: Kluwer Academic Publishers.
- Mc Kevitt, P., S. Ó Nualláin and C. Mulvihill (Eds.) (2002) *Language, vision and music, Readings in Cognitive Science and Consciousness*. Advances in Consciousness Research, AiCR, Vol. 35. Amsterdam, Netherlands: John Benjamins Publishing.
- McTear, M.F. (2002) Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys*, Vol. 34(1), 90-169.
- Michiels, A. (1994) *HORATIO: A Middle-sized NLP Application in Prolog*. University de Liege, Liege, Belgium.
- Microsoft Agent (2002) <http://www.microsoft.com/products/msagent/>, site visited 14/08/2002.
- Mihalcea, R. (2003) SemCor <http://www.senseval.org/> Site visited 25/09/2005.
- Miller, G. (1994) Nouns in WordNet: a Lexical Inheritance System. In *International Journal of Lexicography*, Vol. 3(4), 245-264.
- Minsky, M. (1975) A Framework for representing knowledge, In *Readings in knowledge representation*, R. Brachman and H. Levesque (Eds.), 245-262, Los Altos, CA: Morgan Kaufmann.
- Mueller, E.T. (1998) Natural language processing with ThoughtTreasure. New York: Signiform. <http://www.signiform.com/tt/book/> Site visited 06/09/2002.
- Narayanan, A., D. Manuel, L. Ford, D. Tallis and M. Yazdani (1995) Language Visualisation: Applications and Theoretical Foundations of a Primitive-Based Approach. In *Integration of Natural Language and Vision Processing (Volume II): Intelligent Multimedia*, P. Mc Kevitt (Ed.), 143-163, London, U.K.: Kluwer Academic Publishers.
- Narayanan, S. (1997) Talking the talk is like walking the walk: a computational model of verbal aspect. In *COGSCI-97*, Stanford, CA, 548-553.
- Neal, J. and S. Shapiro (1991) Intelligent Multi-Media Interface Technology. In *Intelligent User Interfaces*, J. Sullivan and S. Tyler (Eds.), 11-43, Reading, MA: Addison-Wesley.
- Nenov, V.I. and Dyer, M.G., (1988) DETE: Connectionist/Symbolic Model of Visual and Verbal Association. In *Proceedings of The Connexionist Models Summer School*, CMU, Pittsburgh.
- Novodex Physics (2006) <http://www.ageia.com/> Site visited 14/01/2006.
- Okada, N. (1996) Integrating Vision, Motion and Language through Mind. In *Artificial Intelligence Review*, Vol. 10, Issues 3-4, August, 209-234.
- Oltramari, A., A. Gangemi, N. Guarino, and C. Masolo (2002) Sweetening ontologies with DOLCE. In A. Gómez-Pérez and V. R. Benjamins (Eds.), Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Sigüenza, Spain, October 1-4, 2002, Proceedings, vol. 2473 of *Lecture Notes in Computer Science*. Springer, 2002.

Ó Nualláin, S. and A. Smith (1994) An Investigation into the Common Semantics of Language and Vision. In *Integration of Natural Language and Vision Processing (Volume II): Intelligent Multimedia*, P. Mc Kevitt (Ed.), 21-30, London, U.K.: Kluwer Academic Publishers.

OpenGL (2005) <http://www.opengl.org/> Site visited 25/09/2005.

OWL (2002) Feature Synopsis for OWL Lite and OWL, W3C Working Draft. <http://www.w3.org/TR/2002/WD-owl-features-20020729/> Site visited 21/08/2002.

Parallelgraphics (2001) VRML products. <http://www.parallelgraphics.com/developer/products/> Site visited 29/05/2003.

Pease A., Niles I., and Li, J. (2002) The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, Edmonton, Canada, July 28-August 1.

Perlin K. and A. Goldberg (1996) Improv: A system for scripting interactive actors in virtual worlds. In *Proceedings of SIGGRAPH 96*, New Orleans, LA, 205-216.

Pinhanez, C., K. Mase, and A. Bobick. (1997) Interval scripts: a design paradigm for story-based interactive systems. In *Proceedings of CHI'97*, 287-294, Atlanta, Georgia, USA, March.

Pinon, C. (1997) Achievements in an event semantics. In *Proceedings of Semantics and Linguistic Theory 7*, A. Lawson (Ed.), 276-292, Ithaca, NY: CLC Publications, Cornell University.

Piwiek, P., B. Krenn, M. Schröder, M. Grice, S. Baumann, and H. Pirker (2002) RRL: A Rich Representation Language for the Description of Agent Behaviour in NECA, *Proc. of the AAMAS workshop on "Embodied conversational agents - let's specify and evaluate them!"*, July, Bologna, Italy.

Poser (2005) <http://www.e-frontier.com>, site visited 25/09/2005.

Pustejovsky, J. (1995) *The Generative Lexicon*. MIT Press.

Pustejovsky, J., J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, and G. Katz (2003) TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of the Fifth International Conference on Computational Semantics (IWCS-5)*, H. Bunt, I. van der Sluis, and R. Morante (Eds.), 337-353, Tilburg, Netherlands.

Qiu, L, Min-Yen Kan and Tat-Seng Chua (2004) A Public Reference Implementation of the RAP Anaphora Resolution Algorithm. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*. Vol. I, 291-294.

Quillian, M. (1968) Semantic Memory. In *Semantic Information Processing*, M. Minsky (Ed.), 227-270, Cambridge, MA: MIT Press.

Qvortrup, L. (Ed.) (2001) *Virtual interaction: interaction in virtual inhabited 3D worlds*. London: Springer.

Reichenbach, H. (1947) *The Elements of Symbolic Logic*. London: Macmillan.

Resnik, P. (1997) Selectional Preference and Sense Disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, 52-57, Washington, DC, U.S.A.

Romary L. and H. Bunt (2002) Towards multimodal content representation. In *Proceedings of LREC 2002, Workshop on International Standards of Terminology and Linguistic Resources Management*, Las Palmas.

SALT (2002) <http://xml.coverpages.org/salt.html> Site visited 20/08/2002.

SAPI (2002) <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcesapi/html/ceoriSpeechAPI SAPIVersion50.asp> Site visited 25/09/2002.

- Sanfilippo, A. (1993) LKB encoding of lexical knowledge. In *Inheritance, Defaults, and the Lexicon*, T. Briscoe, V. de Paiva and A. Copestake (Eds.), 190-222, Cambridge: Cambridge University Press.
- Sassnet, R. (1986) *Reconfigurable Video*. Cambridge, MA: MIT Press.
- Schank, R.C. (1972) Conceptual Dependency: A Theory of Natural Language Understanding *Cognitive Psychology* 3(4): 552-631.
- Schank, R.C. (1973) The Fourteen Primitive Actions and Their Inferences. Memo AIM-183, Stanford Artificial Intelligence Laboratory. Stanford, CA. U.S.A.
- Schank, R.C. (1995) *Tell me a story: narrative and intelligence*. Evanston, Illinois: Northwestern University Press.
- Schank, R.C. and Abelson, R. (1977) *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Hillsdale, NJ: Lawrence Erlbaum.
- Schank, R.C., M. Lebowitz and L. Birnbaum (1980) An integrated understander. *American Journal of Computational Linguistics* (6): 13-30.
- Schödl, A. and I.A. Essa (2002) Controlled animation of video sprites. In *Proceedings of the First ACM Symposium on Computer Animation* (held in Conjunction with ACM SIGGRAPH 2002), San Antonio, TX, USA, July.
- Schödl, A., R. Szeliski, D. Salesin, I. Essa (2000) A new take on textures: video based rendering extends two dimensional textures into the temporal domain. In *Computer Graphics World*, October 2000, 18-20.
- Schwanauer, S. and Levitt, D. (Eds.) (1993) *Machine Models of Music*. Massachusetts, MA: MIT Press.
- Schirra, J. (1993) A Contribution to Reference Semantics of Spatial Prepositions: The Visualisation Problem and its Solution in VITRA. In *The Semantics of Prepositions - From Mental Processing to Natural Language Processing*, C. Zelinsky-Wibbelt (Ed.), 471-515, Berlin: Mouton de Gruyter.
- Siskind, J. M. (1995) Grounding Language in Perception. In *Integration of Natural Language and Vision Processing (Volume I): Computational Models and Systems*, P. Mc Kevitt (Ed.), 207-227, London, U.K.: Kluwer Academic Publishers.
- SMIL (2005) Synchronized Multimedia Integration Language, <http://www.w3.org/AudioVideo/> Site visited 16/01/2006.
- Smith, C. S. (1991) *The Parameter of Aspect*. Dordrecht, The Netherlands: Kluwer.
- Smith, R. (2005) <http://www.ode.org/> Site visited 14/01/2006.
- Smith, S. and J. Bates (1989) Toward a theory of narrative for interactive fiction. Technical Report CMU-CS-89-121, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- Speeth, S. (1961) Seismometer Sounds. *Journal of the Acoustical Society of America*, 33, 909-916.
- Srihari, R.K. and D.T. Burhans (1994) Visual semantics: extracting visual information from text accompanying pictures. In *Proceedings of American Association of Artificial Intelligence (AAAI-94)*, Seattle, U.S.A., 793-798.
- Stede, M. (1996) *Machine Translation*. The Netherlands: Kluwer Academic Publishers.
- Stock, O. and the AlFresco Project Team. (1993) AlFresco: Enjoying the Combination of Natural Language Processing and Hypermedia for Information Exploration. In *Intelligent Multimedia Interfaces*, M. Maybury (Ed.), 197-224, Menlo Park: AAAI/MIT Press.
- Su, A. and R. Furuta (1994) *A Specification of 3D Manipulations in Virtual Environments ISMCR'94: Topical Workshop on Virtual Reality*, Proceedings of the Fourth International

Symposium on Measurement and Control in Robotics, 64-68, NASA Conference Publication 10163, November, Houston, Texas.

Taylor, P., Black, A. and Caley, R. (1998) The architecture of the Festival Speech Synthesis system. In *Proceedings 3rd ESCA Workshop on Speech Synthesis*, 147-151, Jenolan Caves, Australia.

Tesniere, L. (1959) *Elements de syntaxe structurale*. Paris: Klincksieck.

Thórisson, K. (1996) Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills. Ph.D. thesis, Media Arts and Sciences, MIT Media Lab, Massachusetts Institute of Technology, U.S.A.

Thomas, N.J.T. (1999) Are theories of imagery theories of imagination? An active perception approach to conscious mental content. *Cognitive Science*, Vol. 23, 207-245.

Tye, M. (1995) *Ten Problems of Consciousness*, Cambridge, Mass: The MIT Press, Bradford Books.

Tye, M. (2000) *Consciousness, Color, and Content*, Cambridge, Mass: The MIT Press.

van Benthem, J. (1983) *The Logic of Time*. Dordrecht, The Netherlands: Reidel Publishing Company.

Vendler, Z. (1967) *Linguistics and Philosophy*. Ithaca, NY: Cornell University Press.

Verkuyl, H. (1993) *A Theory of Aspectuality*. Cambridge: Cambridge University Press.

VHML Examples (2005) http://www.vhml.org/examples/vhml_examples.shtml Site visited 14/01/2006.

VoiceXML (2004) <http://www.w3.org/TR/2004/REC-voicexml20-20040316/> Site visited 04/12/2005.

Vossen, P., L. Bloksma, H. Rodriguez, S. Climent, N. Calzolari, A. Roventini, F. Bertagna, A. Alonge and W. Peters (1998) The EuroWordNet Base Concepts and Top Ontology. EuroWordNet LE2-4003, Deliverable D017, D034, D036, WP5, <http://www.illc.uva.nl/EuroWordNet/corebcs/topont.html> Site visited 15/08/2002.

VRML (2002) VRML archives. <http://www.web3d.org/x3d/vrml/index.html> Site visited 19/12/2005.

W3C (2002) <http://www.w3.org/XML/> Site visited 20/08/2002.

W3C Voice Architecture (2003) <http://www.w3.org/TR/xhtml+voice/>, site visited 7/14/2006.

Wahlster, W. (1998) User and discourse models for multimodal communication. In *Readings in intelligent user interfaces*, M. Maybury and W. Wahlster (Eds.), 359-370, San Francisco, California: Morgan Kaufmann Publishers, Inc.

Wahlster, W., E. André, W. Finkler, H.J. Profitlich and T. Rist (1993) Plan-based integration of natural language and graphics generation. *Artificial Intelligence*, (63): 387-427.

Wahlster, W., N. Reithinger and A. Blocher (2001) SmartKom: Towards Multimodal Dialogues with Anthropomorphic Interface Agents. In *Proceedings of International Status Conference "Human-Computer Interaction"*, G. Wolf and G. Klein (Eds.), 23-34, DLR, Berlin, Germany, October.

Webber, B., N. N. Badler, B. Di Eugenio, C. Geib, L. Levison, and M. Moore (1995) Instructions, Intentions and Expectations, *Artificial Intelligence Journal*, 73, 253-269.

Wilensky, R. (1981) PAM. In *Inside Computer Understanding: Five Programs Plus Miniatures*, R. C. Schank and C. K. Riesbeck (Eds.), 136-179, Hillsdale, NJ: Lawrence Erlbaum Associates.

Wilson E. and A. Goldfarb (2000) *Living Theater: A History*. Columbus, OH: The McGraw-Hill.

Winograd, T. (1972) *Understanding Natural Language*. New York: Academic Press.