

CONFUCIUS:
**An Intelligent MultiMedia storytelling
interpretation and presentation system**

Minhua Eunice Ma

Supervisor: Prof. Paul Mc Kevitt

First year report

Date: 29 September 2002

Presented as a requirement for Ph.D. in
School of Computing and Intelligent Systems
Faculty of Informatics
University of Ulster, Magee
Email: m.ma@ulster.ac.uk

Abstract

How patterns of perception are interpreted in human beings' brains is a perpetual topic in the disciplines of analytical philosophy, experimental psychology, cognition science, and more recently, artificial intelligence. When humans read (or hear, think, imagine) natural language, they often create internal images in their mind. This report presents our progress toward the creation of dynamic imagery by presenting a genus of natural language-stories as animation and other concomitant modalities. The project 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.

Previous research in related areas of language visualisation, multimodal storytelling, intelligent multimedia agents and interfaces, non-speech audio, and cognitive science is surveyed, and corresponding computer systems are explored. Various methods of multimodal semantic representation, and multimedia fusion and coordination in these systems are also reviewed. The objective of the work described in this research report is the development of CONFUCIUS, an intelligent multimedia storytelling interpretation and presentation system that automatically generates multimedia presentations from natural language stories or drama/movie scripts. The storytelling employs several temporal media such as animation, speech and sound for the presentation of stories. Establishing correspondence between language and dynamic vision is the focus of this research. CONFUCIUS explores the areas of natural language understanding, computer animation, and autonomous agents, in which we blend artificial intelligence technology with ideas and insight from traditional arts and media. This work is significant because it brings us closer to the goal of making a more realistic virtual reality world from human natural language.

This report presents progress toward the automatic creation of multimodal presentation, in particular animations, from natural language input. There are three main areas of contribution: multimodal semantic representation of natural language, multimodal fusion and coordination, and automatic animation generation. Existing multimodal semantic representations may represent the general organization of semantic structure for various types of inputs and outputs within a multimodal dialogue architecture and are usable at various stages such as fusion and discourse pragmatics aspects. However, there is a gap between high level general multimodal semantic representations and lower-level representations that are capable of connecting meanings in various modalities. Such a lower-level meaning representation, which links linguistic modality to visual modalities, is proposed in the report. This research also introduces a new method of multimedia fusion and coordination. It will be implemented using VRML and Java. In addition, the work will also advance automatic text-to-graphic generation through the development of CONFUCIUS.

CONFUCIUS will be developed using existing software tools such as Gate and WordNet for natural language processing, 3D Studio Max for object modelling, Microsoft Agent and Poser for humanoid animation.

Keywords: artificial intelligence, multimedia generation, multimedia presentation, storytelling, story visualisation, natural language processing, language parsing and understanding, visual semantics, language visualisation, language animation, 3D computer graphics, autonomous agents, virtual reality.

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. 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 one year. Also I am grateful to the Intelligent Multimedia Group member Dimitrios Konstantinou, who also works on the Seanchaí intelligent multimedia storytelling platform, from our discussion I receive a lot of information in natural language processing. I want to thank Ted Leath, Pat Kinsella and Bernard McGarry for their technical support. Finally, I thank Sina Rezvani for his friendship.

Contents

ABSTRACT	II
ACKNOWLEDGEMENTS	III
1. INTRODUCTION: THE MOTIVATION FOR CONFUCIUS	1
1.1. OBJECTIVES OF THE RESEARCH	1
1.2. FEATURES OF CONFUCIUS – MULTIMODAL, ANIMATION, INTELLIGENT	2
1.2.1. <i>Multimodal input and output</i>	2
1.2.2. <i>Animation</i>	2
1.2.3. <i>Intelligent</i>	3
1.3. AREAS OF CONTRIBUTION	3
1.4. CONTEXT: SEANCHAÍ AND CONFUCIUS	4
2. LITERATURE REVIEW	5
2.1. MULTIMODAL SEMANTIC REPRESENTATIONS	5
2.1.1. <i>Semantic networks</i>	7
2.1.2. <i>Frame representation and frame-based systems</i>	7
2.1.3. <i>Multimodal semantic representation in XML</i>	9
Speech markup language specifications—VoiceXML and SALT	11
XML representation of semantics--OWL	12
The Semantic Web	13
Multimodal systems using XML-based representation	14
2.1.4. <i>Schank's Conceptual Dependency theory and scripts</i>	15
2.1.5. <i>Event-logic truth conditions</i>	18
2.1.6. <i>X-schemas and f-structs</i>	19
2.2. MULTIMEDIA FUSION, COORDINATION AND PRESENTATION	20
2.2.1. <i>Intelligent multimedia authoring systems</i>	20
2.2.2. <i>Content selection</i>	22
2.2.3. <i>Media preferences</i>	23
2.2.4. <i>Coordination across media</i>	24
2.2.5. <i>Consistency of expression</i>	24
2.3. AUTOMATIC TEXT-TO-GRAPHICS SYSTEMS	25
2.3.1. <i>WordsEye</i>	25
2.3.2. <i>'Micons' and CD-based language animation</i>	27
2.3.3. <i>Spoken Image (SI) and SONAS</i>	27
2.4. MULTIMODAL STORYTELLING	29
2.4.1. <i>AESOPWORLD</i>	29
2.4.2. <i>KidsRoom</i>	30
2.4.3. <i>Interactive storytelling</i>	31
2.4.4. <i>Oz</i>	32
2.4.5. <i>Virtual theater and Improv</i>	33
2.4.6. <i>Computer games</i>	34
2.5. INTELLIGENT MULTIMEDIA AGENTS	34
2.5.1. <i>BEAT and other interactive agents</i>	35
2.5.2. <i>Divergence on agents' behavior production</i>	36
2.5.3. <i>Gandalf</i>	37
2.5.4. <i>PPP persona</i>	38
2.6. INTELLIGENT MULTIMEDIA INTERFACES	39

2.7. NON-SPEECH AUDIO.....	40
2.7.1. <i>Auditory icons</i>	41
2.7.2. <i>Earcons</i>	41
2.7.3. <i>Sonification</i>	41
2.7.4. <i>Music synthesis</i>	42
2.7.5. <i>Non-speech audio in CONFUCIUS</i>	42
2.8. MENTAL IMAGERY IN COGNITIVE SCIENCE	43
3. MULTIMODAL SEMANTIC REPRESENTATIONS	46
3.1. VISUAL SEMANTICS IN A GENERAL PURPOSE KNOWLEDGE BASE.....	47
3.2. BASE CONCEPTS--EQUIVALENCES ACROSS PART-OF-SPEECH	48
3.3. CATEGORIES OF NOUNS FOR VISUALISATION	49
3.4. VISUAL SEMANTIC REPRESENTATION OF EVENTS—MEANING AS ACTION	52
3.4.1. <i>Categories of events in animation</i>	52
3.4.2. <i>Extending predicate-argument representation to word level</i>	54
Constants, variables, types and their naming schemes	55
Hierarchical structure of predicate-argument primitives	55
Examples of verb definitions in extended predicate-argument model	57
3.4.3. <i>Representing active and passive voice</i>	60
3.4.4. <i>Representing tense and aspect</i>	60
3.5. VISUAL SEMANTIC REPRESENTATION OF ADJECTIVES—MEANING AS ATTRIBUTE	60
3.5.1. <i>Categories of adjectives for visualisation</i>	60
3.5.2. <i>Semantic features of adjectives relating to visualisation</i>	62
3.6. VISUAL SEMANTIC REPRESENTATION OF SPATIAL PREPOSITIONS	64
4. PROJECT PROPOSAL.....	66
4.1. ARCHITECTURE OF CONFUCIUS	66
4.2. INPUT STORIES/SCRIPTS	67
4.3. DATA FLOW OF CONFUCIUS	67
4.4. COMPARISON WITH PREVIOUS WORK	69
4.5. ANIMATION GENERATION.....	69
4.5.1. <i>Animated narrator</i>	71
4.5.2. <i>Synthetic actors</i>	72
Motion animation	72
4.5.3. <i>Default attributes in object visualisation</i>	72
4.5.4. <i>Layout</i>	73
4.6. MULTIMEDIA PRESENTATION PLANNING.....	73
4.7. ISSUES RAISED	74
4.7.1. <i>Size of CONFUCIUS' knowledge base</i>	74
4.7.2. <i>Modelling dynamic events</i>	75
4.7.3. <i>Ungrammatical sentences in natural language input</i>	75
4.7.4. <i>Deriving visual semantics from text</i>	76
4.8. PROJECT SCHEDULE AND CURRENT STATUS	76
5. SOFTWARE ANALYSIS.....	77
5.1. NATURAL LANGUAGE PROCESSING (NLP) TOOLS.....	77
5.1.1. <i>Natural language processing in CONFUCIUS</i>	77
5.1.2. <i>Syntactic parser</i>	77
5.1.3. <i>Semantic inference</i>	78
5.1.4. <i>Text-to-speech</i>	81
5.2. THREE DIMENSIONAL GRAPHIC AUTHORING TOOLS AND MODELLING LANGUAGES	81

5.2.1. <i>Three-dimensional animation authoring tools</i>	81
5.2.2. <i>Three-dimensional graphic modelling language — VRML</i>	82
Using Background node to build stage setting	83
Using interpolators and ROUTE to produce animation	84
Using Viewpoint node to guide users' observation.....	84
5.2.3. <i>Java in VRML Script node</i>	85
5.2.4. <i>Basic narrative montage and their implementation in VRML</i>	86
5.3. USING AUTONOMOUS AGENTS TO MODEL THE ACTORS	88
6. CONCLUSION AND FUTURE WORK	91
REFERENCES	93
APPENDIX A: PROJECT SCHEDULE	101

1. Introduction: the motivation for CONFUCIUS

How patterns of perception are interpreted in human beings' brains is a perpetual topic in the disciplines of analytical philosophy, experimental psychology, cognition 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 (Thomas 1999).

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 is the original motivation of the research on this project. In spite of the long "picture-description" controversy in philosophy during the 1970s (Kosslyn 1994), we will develop and implement a system, CONFUCIUS, which can automatically create imagery by presenting a genus of natural language stories as animation and other concomitant modalities. CONFUCIUS will contribute 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. CONFUCIUS will be an automatic multimedia presentation storytelling system, which integrates and improves state-of-the-art theories and techniques in the areas of natural language processing, intelligent multimedia presentation and language visualisation.

To build CONFUCIUS we should first study the techniques in conventional manual animation. The most successful multimedia storytelling is probably Disney's animations. 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 animation is still a difficult and time-consuming job. An intelligent multimedia storytelling system that can generate animations dynamically to 'tell' stories at run-time will spare much labour on animation direction and creation.

1.1. Objectives of the research

The main aim of this research is to present stories using temporal media (e.g. animation and speech) from natural language stories or drama/movie scripts. The primary objectives of CONFUCIUS are summarized as below:

- ❑ To interpret natural language story or movie (drama) script input and to extract concepts from the input
- ❑ To generate 3D animation and virtual worlds automatically, with speech and non-speech audio
- ❑ To integrate the above components to form an intelligent multimedia storytelling system for presenting multimodal stories

The motivation of this project comes from the domain of the integration of natural language and vision processing (Mc Kevitt 1995a,b, 1996a,b, Maybury 1993,1994, Maybury and Wahlster 1998, Qvortrup 2001, and Granstrom et al. 2002). There are two directions of the 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 visualize 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.

In this project, an intelligent multimedia storytelling system, CONFUCIUS, which presents stories with 3D animation using high image quality (not iconic) will be designed and implemented.

1.2. Features of CONFUCIUS – multimodal, animation, intelligent

1.2.1. Multimodal input and output

As illustrated in Figure 1.1, CONFUCIUS will use natural language input including traditional typed text and a tailored menu that facilitates input of movie/drama scripts in a specific format to generate spoken language (dialogue), animation, and non-speech audio outputs. It gives the audience a richer perception than the usual linguistic narrative. Since all the output media are temporal, CONFUCIUS requires coordination and synchronisation among these output modalities.

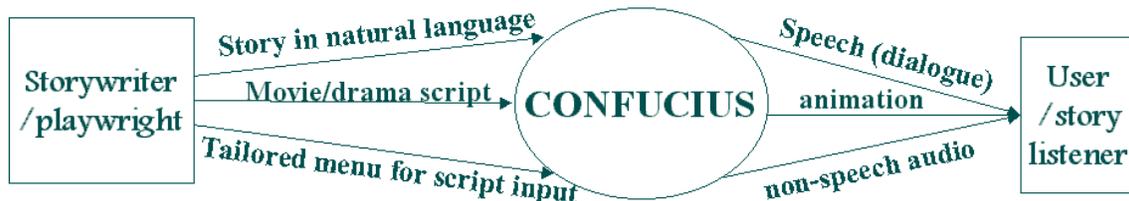


Figure 1.1: Multimodal I/O of CONFUCIUS

Pictures 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 medium alone. In this sense, multimedia storytelling systems may present stories more effectively than oral storytelling and strip cartoons.

As an automatic storytelling system which is inspired by performance arts, the elements of CONFUCIUS correspond to those in conventional theatre arts — *Aristotle’s six parts of a Tragedy* (Wilson and Goldfarb 2000). Table 1.1 shows the corresponding relationship between them, which ensures that CONFUCIUS’ applications in automatic play/cinema direction.

1.2.2. Animation

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. However, except animated conversational agents that emulate lip movements, face expressions and body poses and animated icons, few systems can convert

English text into animation. CONFUCIUS will translate stories expressed in usual typed-in text or script format into a 3D animation presentation with characters' speech and other sound effects. The use of animated characteristics would enable movie makers and drama directors to preview the story by watching the animated effects of actors (protagonists) with props on the stage.

Aristotle's six parts of a Tragedy	Elements of CONFUCIUS	Output modalities of CONFUCIUS
1. Plot	story/play script	
2. Character	actor (protagonist)	Animation
3. Theme (idea)	story/play script	
4. Diction (Language)	dialogue and narrative	speech
5. Music (sound)	non-speech audio	non-speech audio
6. Spectacle	user/story listener	/

Table 1.1: Elements of CONFUCIUS vs. Aristotle's six parts of a Tragedy

1.2.3. Intelligent

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 appropriated presentations based on *presentation knowledge* and *domain knowledge*. The *intelligence* of CONFUCIUS is embodied in the automatic generation of animation with only optional minor user intervention at the beginning of storytelling to help CONFUCIUS set the actors. For example, there is an optional function that will enable users to choose favourite characters in a story before it is told. CONFUCIUS then generates the stage based on scene descriptions, and creates the animation to present the actors' actions and coordinated dialogue in order to present events in the story.

1.3. Areas of contribution

There are three challenges raised by the task of building multimodal presentation of natural language stories in CONFUCIUS. The first challenge is multimodal semantic representation of natural language. This research introduces a new representation method of multimodal semantic representation and especially visual semantic representation. Existing multimodal semantic representations may represent the general organization of semantic structure for various types of inputs and outputs within a multimodal dialogue architecture and are usable at various stages such as fusion and discourse pragmatics aspects. However, there is a gap between high-level general multimodal semantic representations and lower-level representations that is capable of connecting meanings in various modalities. Such a lower-level meaning representation, which links linguistic modalities to visual modalities, is proposed in this report.

The second is one of the requirements for multimedia presentation: to tell stories consistently through multiple media to achieve maximum impact on the human senses, representations of various media need to be fused, coordinated and integrated. This research proposes a new methodology of multimedia fusion and integration.

Third, the work will also make advancement on automatic language visualisation (or automatic conversion of text-to-animation) through the development of CONFUCIUS. Current research on language visualisation suffers from a lack of deep understanding of natural language and inferences, and a lack of richness on temporal media representation. Success in the first two challenges contributes to solve these problems and hence enables CONFUCIUS to interpret stories properly and present them with realistic animated 3D graphics.

All these unique contributions will be implemented and tested in CONFUCIUS using Virtual Reality Modelling Language (VRML) and the Java programming language.

1.4. Context: Seanchaí and CONFUCIUS

The long-time goal of this work is using the methodology presented here to generate 3D animation automatically in an intelligent multimedia storytelling platform called Seanchaí¹. Seanchaí will perform multimodal storytelling generation, interpretation and presentation and consists of *Homer*, a storytelling generation module, and *CONFUCIUS*, a storytelling interpretation and presentation module (see Figure 1.2). The output of Homer could be fed as input to CONFUCIUS. *Homer* focuses on natural language story generation. It will receive two types of input from the user (1) either the beginning or the ending of a story in the form of a sentence and (2) stylistic specifications, and proceeds to generate natural language stories; and CONFUCIUS focuses on story interpretation and multimodal presentation. It receives input natural language stories or (play/movie) scripts and presents them with 3D animation, speech, non-speech sound, and other modalities.

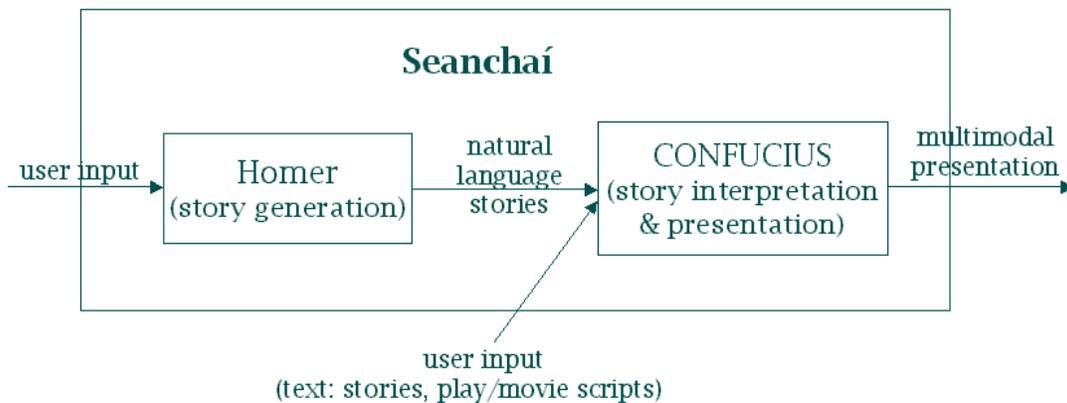


Figure 1.2: Intelligent storytelling platform -- *Seanchaí*

In chapter 2 we investigate related research in the areas of multimodal semantic representation and media coordination, describe previous systems of language visualisation, multimodal storytelling, intelligent multimedia agents and interfaces, and review support from cognitive science. Then we propose the multimodal semantic representation of CONFUCIUS' knowledge base and its intermediate visual semantic representation in chapter 3. We turn next in chapter 4 to the system and unit design of CONFUCIUS and compare it with related work. Next we explore and analyse software for natural language processing, 3D graphics modelling languages and authoring tools which will be used in the development of CONFUCIUS in chapter 5. Finally chapter 6 concludes and discusses areas for further research.

¹ Seanchaí means 'storyteller' in Gaelic.

2. Literature review

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. Furthermore, in the vast majority of non-trivial applications the information needs will vary from user to user and from domain to domain. An *intelligent multimedia presentation* system should be able to flexibly generate various presentations to meet individual requirements of users, situations, and domains. It requires intelligent multimedia systems to have ability of reasoning, planning, and generation. Research in this area initiated during the mid 1980s (Maybury 1993, 1994, Maybury and Wahlster 1998, Qvortrup 2001, Mc Kevitt 1995a,b, 1996a,b, and Granstrom et al. 2002).

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 greatly 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 this chapter previous work in intelligent multimedia applications is described. We investigate the elements of intelligent storytelling that we believe make traditional storytelling (e.g., literature, drama, film, animation) powerful: *plot*, *characters*, and *presentation*. Since plot is already determined in story input of CONFUCIUS, it will not be covered in our discussion. Our research focuses on how to create more believable *characters* and make more realistic *presentations* to tell an immersive story. Toward this goal we explore work in: automatic text-to-graphics systems, multimodal storytelling, intelligent multimedia interfaces, and non-speech audio (for *presentation*), and autonomous agents (for *characters*). We also review the various methods of multimodal semantic representation, and multimedia fusion, coordination and presentation. Finally we investigate the topic of mental imagery from the field of cognitive science.

2.1. Multimodal semantic representations

Multimodal interpretation, realization and integration in intelligent multimedia systems have general requirements of multimodal semantic representations: they should support both interpretation and generation, support any kind of multimodal input and output, and support the variety of semantic theories. A multimodal representation may contain architectural, environmental, and interactional 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). Interactional representation indicates speaker/user's state or other addressees. Frame-based representation and XML representation are traditional multimodal semantic representations. They are common in recent intelligent multimedia applications to represent multimodal semantics, such as CHAMELEON (Brøndsted et al. 2001), AESOPWORLD (Okada 1996), REA (Cassell et al. 2000), Ymir (Thórisson 1996) and WordsEye (Coyne and Sproat 2001) based on frame

representations to represent semantic structure of multimodal content. 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).

There are several general knowledge representation languages which have been implemented in artificial intelligence applications: rule-based representation (e.g. CLIPS (2002)), First Order Predicate Calculus (FOPC), semantic networks (Quillian 1968), Conceptual Dependency (CD) (Schank 1973), and frames (Minsky 1975). 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 an ability to reason about time and space. For the purposes of this project, though, focus will be on 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 (i.e. category (2) in Table 2.1) 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 L₀ (Bailey et al. 1997) is based on x-schemas and f-structures.

Table 2.1 shows categories of major knowledge representations and their typical suitable applications. 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. The new visual semantic representation *extended predicate-argument representation* which will be proposed later in the report also belongs to this category. All of them could be used to represent visual semantics in movement recognition or generation applications as shown in the table.

categories	knowledge representations	suitable applications
(1) general knowledge representation and reasoning	rule-based representation	expert systems
	FOPC	
	semantic networks	lexical semantics
	frames	multimodal semantics
	XML	
(2) physical knowledge representation and reasoning (inc. spatial /temporal reasoning)	CD	dynamic vision (movement) recognition and generation
	event-logic truth conditions	
	x-schema and f-structure	
	extended predicate-argument representation	

Table 2.1: Categories of knowledge representations

Figure 2.1 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.

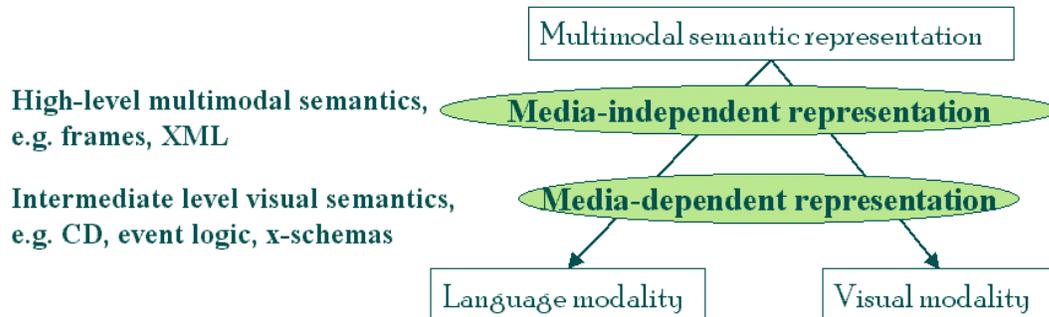


Figure 2.1: Multimodal semantic representations and visual semantic representations

Now we will discuss the above semantic representations respectively in detail.

2.1.1. 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 lexical reference system in which English vocabulary is organized into semantic networks.

2.1.2. Frame representation and frame-based systems

Frames were introduced by Minsky (1975) in order to represent *situations*. It 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 parent of a frame X represents a more general concept than does X (a superset of the set represented by X), and a child of X represents a more specific concept than does X. A collection of frames in one or more inheritance hierarchies is a *knowledge base*. The main features of frame representation are:

1. *Object-orientation*. 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.

2. *Generalization/Specialization*. 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.

3. *Reasoning*. 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, is important to knowledge representation.

4. *Classification*. Given an abstract description of a concept, most knowledge representation languages provide the ability to determine if a concept fits that description, this is actually a common special form of reasoning.

Object orientation and *generalization* make the represented knowledge more understandable to humans, *reasoning* and *classification* make a system behave as if it knows what is represented.

Since frames were introduced in 1970s, many knowledge representation languages have been developed based on this concept. 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-based representations are CHAMELEON (Brøndsted et al. 2001), WordsEye (c.f. section 2.3.1), AESOPWORLD (c.f. section 2.4.1), Ymir (Thórisson 1996) and REA (Cassell et al. 2000).

CHAMELEON is an *IntelliMedia workbench* application. IntelliMedia focusses on computer processing and understanding of signal and symbol input from at least speech, text and visual images in terms of semantic representations. CHAMELEON is a software and hardware platform tailored to conducting IntelliMedia in various application domains. In CHAMELEON a user can ask for information about things on a physical table. Its current domain is a *Campus Information System* where 2D building plans are placed on the table and the system provides information about tenants, rooms and routes and can answer questions like *Where is the computer room?* in real time. CHAMELEON has an open distributed processing architecture and includes ten agent modules: blackboard, dialogue manager, domain model, gesture recogniser, laser system, microphone array, speech recogniser, speech synthesiser, natural language processor, and a distributed Topsy learner, as shown in Figure 2.2a. All modules in CHAMELEON communicate with each other in *blackboard frame semantics* and can produce and read frames through the blackboard (see Figure 2.2b). Frames are coded as messages built of predicate-argument structures following the BNF (Backus Naur Form) definition. Figure 2.3 shows the input frames (speech and gesture), NLP frame, and output frames (speech and laser pointing) for a dialogue, in which the user asks *Who is in this office?* when he points to a room on the 2D building plan, and CHAMELEON answers *Paul is in this office* meanwhile pointing the place on the map with a laser beam.

Ymir (Thórisson 1996) and REA (Cassell et al. 2000) also use similar frame-based representations in their multimodal interaction. 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 could represent the mechanism for performing this "inference" quite well. Frame-based systems that can deal with this kind of knowledge do so by adding a procedural language to the representation. This knowledge is *not* represented in a frame-based way, it is represented as LISP code which is accessed through a slot in the frame (Bobrow and Winograd 1985).

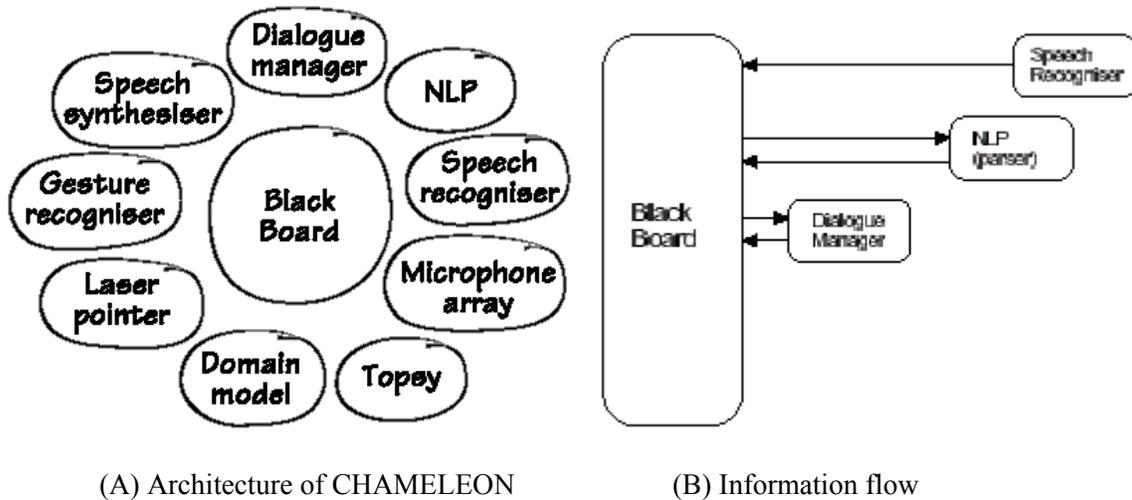


Figure 2.2: Architecture and information flow of CHAMELEON

```
[SPEECH-RECOGNISER
  UTTERANCE: (Who is in this office?)
  INTENTION: query?
  TIME: timestamp]

[GESTURE
  GESTURE: coordinates (3,2)
  INTENTION: pointing
  TIME: timestamp]

[NLP
  INTENTION: query? (who)
  LOCATION: office (tenant Person) (coordinates(X,Y))
  TIME: timestamp]

[SPEECH-SYNTHESIZER
  INTENTION: declarative (who)
  UTTERANCE: (Paul is in this office)
  TIME: timestamp]

[LASER
  INTENTION: description (pointing)
  LOCATION: coordinates(3,2)
  TIME: timestamp]
```

Figure 2.3: Example frames in CHAMELEON

2.1.3. Multimodal semantic representation in XML

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. Its primary purpose is as an electronic publishing and data interchange format. XML documents are made up of *entities* which contain either parsed or unparsed data. Parsed

data is either *markup* or *character data* (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 *application* that reads it. A software module which is used to read XML documents and provide access to their content and structure is called an *XML processor* or an *XML parser*. 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 in principle 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. Figure 2.4 shows two examples of *Speech Synthesis Markup Language* (SSML), and *Speech Recognition Grammar Specification* (SRGS). Example A shows the speech synthesis facility which could be used in synthesizing greeting message and prompt information with different voice features specified in *voice* tags of SSML. Example B shows a simple XML form grammar of SRGS, which recognizes users' speech input of city and state names and stores them in corresponding variables.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<speak version="1.0" xml:lang="en-US"
xmlns="http://www.w3.org/2001/10/synthesis">
  <voice gender="female">Mary had a little lamb.</voice>
  <!-- now request a different female child's voice -->
  <voice gender="female" variant="2">
    It's fleece was white as snow.
  </voice>
  <!-- platform-specific voice selection -->
  <voice name="Mike">I want to be like Mike.</voice>
</speak>
```

(A): An example of Speech Synthesis Markup Language specification

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE grammar PUBLIC "-//W3C//DTD GRAMMAR 1.0//EN"
"http://www.w3.org/TR/speech-grammar/grammar.dtd">

<grammar xmlns="http://www.w3.org/2001/06/grammar"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/06/grammar
http://www.w3.org/TR/speech-grammar/grammar.xsd"
xml:lang="en" version="1.0" root="city_state" mode="voice">

  <rule id="city" scope="public">
    <one-of>
```

```

        <item>Boston</item>
        <item>Philadelphia</item>
        <item>Fargo</item>
    </one-of>
</rule>

<rule id="state" scope="public">
    <one-of>
        <item>Florida</item>
        <item>North Dakota</item>
        <item>New York</item>
    </one-of>
</rule>

<!-- Reference by URI to a local rule -->
<!-- Artificial example allows "Boston, Florida"! -->
<rule id="city_state" scope="public">
    <ruleref uri="#city"/> <ruleref uri="#state"/>
</rule>
</grammar>

```

(B) An example of XML form grammar

Figure 2.4: Examples of Speech Synthesis and Recognition Markup Language

Speech markup language specifications—VoiceXML and SALT

There are some specific standards of XML specially designed for the purpose of multimodal access to the Internet. VoiceXML and SALT are the two universal standards in W3C. Both of them are standards for speech-enabled Web applications.

VoiceXML was announced by AT&T, Lucent, and Motorola. The spoken interfaces based on VoiceXML prompt users with prerecorded or synthetic speech and understand simple words or phrases. Combined with richer natural language processing, multimodal interaction will enable the user to speak, write and type, as well as hear and see using a more natural user interface than today's browsers.

Founded by Cisco, Comverse, Intel, Philips, and Microsoft, SALT (Speech Application Language Tags) is an open standard designed to augment existing XML-based markup languages to provide spoken access to many forms of content through a wide variety of devices and to promote multimodal interaction and enable voice on the Internet. The SALT forum has announced its multimodal access “will enable users to interact with an application in a variety of ways: they will be able to input data using speech, a keyboard, keypad, mouse and/or stylus, and produce data as synthesized speech, audio, plain text, motion video, and graphics. Each of these modes will be able to be used independently or concurrently.” (SALT 2002)

SALT has a wide range of capabilities, such as speech input and output, and call control. The main elements of SALT are (1) <prompt> tag for configuring the speech synthesizer and playing out prompts; (2) <listen> and <reco> tags for configuring the speech recognizer, executing recognition, and handling recognition events; (3) <grammar> tag for specifying input grammar resources; (4) <bind> tag for processing recognized results. These elements are activated either declaratively or programmatically under script running on the client device. In addition, its call control object can be used to provide SALT-based applications with the ability to place, answer, transfer and disconnect calls, along with advanced capabilities such as conferencing. The SALT

specification thus defines a set of lightweight tags as extensions to commonly used Web-based markup languages. It also draws on W3C standards such as SSML and SRGS and semantic interpretation for speech recognition to provide additional application control.

XML representation of semantics--OWL

There are also some semantic markup languages in the XML family in W3C, such as Ontology Web Language (OWL). Published by the W3C's Web Ontology Working Group, OWL is a semantic markup language for publishing and sharing ontologies² on the World Wide Web. It is derived from the DAML+OIL (DARPA Agent Markup Language, Ontology Interchange Language) Web Ontology Language (DAML_OIL 2001) and builds upon the Resource Description Framework (RDF). OWL supports the use of automated tools which "can use common sets of terms called ontologies to power services such as more accurate Web search, intelligent software agents, and knowledge management. It can be used for applications that need to understand the content of information instead of just understanding the human-readable presentation of content. OWL facilitates greater machine readability of web content than XML, RDF, and RDF-S (RDF namespaces) by providing an additional vocabulary for term descriptions." (OWL 2002)

OWL defines basic semantic relations of web ontology. Most of its relations can find their counterparts in WordNet (Beckwith et al. 1991), a lexical reference system in which English vocabulary is organized into semantic networks. Moreover, OWL includes logical information associated with an ontology and hence has more logical inferences embedded within it. Table 2.2 compares some OWL class elements with their counterparts in WordNet. We can see that OWL has finer granularity in its classification, i.e. one relation in WordNet might has several corresponding elements in OWL such as both `subClassOf` and `oneOf` correspond to `hypernym`, and both `sameClassAs` and `samePropertyAs` correspond to `synonym`. In the example parent `inverseOf` child the `inverseOf` relation is more exact than `antonym`.

<i>OWL elements</i>	<i>WordNet relations</i>	<i>Example</i>
<code>subClassOf</code>	<code>hypernym</code>	<pre> person subClassOf mammal <owl:oneOf rdf:parseType="Collection"> <owl:Thing rdf:about="#Sunday"/> <owl:Thing rdf:about="#Monday"/> <owl:Thing rdf:about="#Tuesday"/> <owl:Thing rdf:about="#Wednesday"/> <owl:Thing rdf:about="#Thursday"/> <owl:Thing rdf:about="#Friday"/> <owl:Thing rdf:about="#Saturday"/> </oneOf> </pre>
<code>oneOf</code> (enumerated classes)	<code>hypernym</code> (with fixed cardinality of members)	
<code>sameClassAs,</code> <code>samePropertyAs</code>	<code>synonym</code>	<pre> car sameClassAs automobile </pre>
<code>inverseOf</code>	<code>antonym</code>	<pre> parent inverseOf child </pre>

Table 2.2: Comparison of OWL class elements and WordNet relations

² An *ontology* in terms of the WG charter defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information. Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them.

OWL defines some logical information such as Boolean combinations, set operation, and logical relations that facilitate semantic inference. It allows arbitrary Boolean combinations of classes: `unionOf`, `complementOf`, and `intersectionOf`. For example, citizenship of the European Union could be described as the union of the citizenship of all member states. `TransitiveProperty`, `SymmetricProperty`, `FunctionalProperty`, and `InverseFunctionalProperty` define the transitive, symmetric, function, and inverse function relations respectively.

OWL also provide facilities for coreference resolution: the elements `sameIndividualAs`, (e.g. George Bush `sameIndividualAs` American President) and `differentIndividualFrom`. A reasoner can deduce that *the president* refers to *Bush*, and X and Y refer to two unique individuals if X `differentIndividualFrom` Y. Stating differences can be important in systems such as OWL (and RDF) that do not assume that individuals have only one name.

Cardinality (including *minCardinality*, *maxCardinality*, *cardinality*) is provided in OWL as another convenience when it is useful to state that a property with respect to a particular class. For example the class of *dinks* ("a couple who both have careers and no children") would restrict the cardinality of the property `hasIncome` to a minimum cardinality of two while the property `hasChild` would have be restricted to cardinality 0.

OWL shows that the potential and flexibility of XML can be applied to represent not only multimodal semantics but also lexical semantics for language processing.

The Semantic Web

Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully. Computers can parse web pages for layout and routine processing but they have no reliable way to process the semantics. Machines cannot understand the meaning of the contents and links on a web page. *The Semantic Web* (Berners-Lee et al. 2001) aims to make up for this. The idea is to have data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.

The Semantic Web is not a separate Web but an extension of the current web, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The first steps in weaving the Semantic Web into the structure of the existing Web are already under way. There are three basic components necessary to bring computer-understandable meaning to the current web and hence extend it to the Semantic Web: (1) a structure to the meaningful content of Web pages; (2) a logic to describe complex properties of objects, which is the means to use rules to make inferences, choose courses of action and answer questions; and (3) collections/taxonomy of information, called ontologies. Classes, subclasses and relations among entities are a powerful tool for Web use. Inference rules in ontologies supply further power.

Three important technologies for developing the Semantic Web are already in place: XML, RDF (Resource Description Framework) and OWL. As we discussed above, XML allows users to create arbitrary structure to their documents by adding tags but says nothing about what the structures mean. Meaning is expressed by RDF, which encodes it in sets of triples, each triple being like the subject, verb and object of an elementary sentence. In RDF, a document makes assertions that particular things (e.g. people or web pages) have properties (such as "is a student of," "is the author of") with certain values (e.g. another person or another web page). This

structure turns out to be a natural way to describe the vast majority of the data processed by machines. Subject, verb, and object are each identified by a Universal Resource Identifier (URI), just as used in a link on a Web page (URLs, Uniform Resource Locators, are the most common type of URI). This mechanism enables anyone to define a new concept, a new verb, just by defining a URI for it somewhere on the Web. OWL can be used to improve the accuracy of Web searches—the search program can look for only those pages that refer to a precise concept instead of all the ones using ambiguous keywords. More advanced applications will use ontologies to relate the information on a page to the associated knowledge structures and inference rules.

In the Semantic Web all information is readily processed by computer applications and could be used to answer queries that currently require a human to sift through the content of various pages turned up by a search engine. This is only a simple use of the Semantic Web. In the future, the Semantic Web will break out of the virtual realm and extend into the physical world, e.g. making our consumer electronics intelligent by using the RDF language to describe devices such as mobiles and microwaves.

Multimodal systems using XML-based representation

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 information on facilities such as hotels, restaurants, and theatres. It can understand speech combined with video-based recognition of natural gestures and facial expressions. Users may delegate a task to an interface agent. 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. The word and gesture lattice, the hypotheses about facial expressions, the media fusion results, the presentation plan and the discourse context are all represented in M3L. SmartKom uses unification and an overlay operation of typed feature structures encoded in M3L for media fusion and discourse processing. Figure 2.5 lists the M3L representation in an example in which SmartKom presents a map of Heidelberg highlighting a location of cinemas called *Europa*. The first element in the XML structure describes the cinema with its geo-coordinates. The identifier `pid3072` links to the description of *Europa*. The second element contains a panel element `PM23` displayed on the map and its display coordinates.

```
<presentationContent>
...
  <abstractPresentationContent>
    <movieTheater strucId=pid3072>
      <entityKey> cinema_17a </entityKey>
      <name> Europa </name>
      <geoCoordinate>
        <x> 225 </x> <y> 230 </y>
      </geoCoordinate>
    </movieTheater>
  </abstractPresentationContent>
...
  <panelElement>
    <map strutId="PM23">
      <boundingShape>
        <leftTop>
          <x>0.5542 </x> <y> 0.1950 </y>
        </leftTop>
```

```

        <rightBottom>
          <x>0.9892 </x> <y> 0.7068 </y>
        </rightBottom>
      </boundingShape>
      <contentRef> pid3072 </contentRef>
    </map>
  </panelElement>
  ...
</presentationContent>

```

Figure 2.5: M3L representation of SmartKom

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 software as well. In the Gate natural language processing platform (Cunningham et al. 2002) XML format is used for inter-component communication, every module can parse XML input and generate output in XML format.

2.1.4. Schank's 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, and improve the machine understanding to some degree and present stories more exactly. 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 2.6). 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 2.6: Conceptual Dependency primitives

For example, the sentence: “I gave John a book.” can be depicted in CD theory as shown in Figure 2.7. 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.

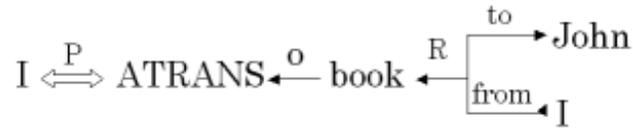


Figure 2.7: Conceptual representation of “I gave John a book.”

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, a lot of inference is still required. An implemented text-to-animation system based on CD primitives (Narayanan et al. 1995) 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—routine ways of responding to greetings, routine ways to go to work every morning, etc.—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 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 some 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 a few 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.

We discuss now two visual semantic representations of simple action verbs, event-logic truth conditions and f-structs. Both of them are mainly designed for verb labelling (recognition). The task of our work is a reverse process to visual recognition, i.e. language visualisation in CONFUCIUS. 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.

2.1.5. 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: lack of conditional selection, i.e. this framework does not provide a mechanism for selection restrictions of the arguments, and 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.³

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

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:

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

³ 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'.

The definition describes the alternate movement of two feet y_1 and y_2 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' as well. 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 individual frame class, such as `contacts()`, `supports()`, and `attached()`. For instance, when one object is supported by another, it usually contacts the supporting object. The other problem is that some primitives in frame-by-frame class are not atomic, i.e. could be described by combinations of others, such as `slideAgainst(x, y)` might be performed by `translatingTowards() \wedge 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 execute motor actions.

2.1.6. 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 (Figure 2.8). 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 2.3 shows the f-structure of one word-sense of *push*, using the *slide* x-schema (Figure 2.8). 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.

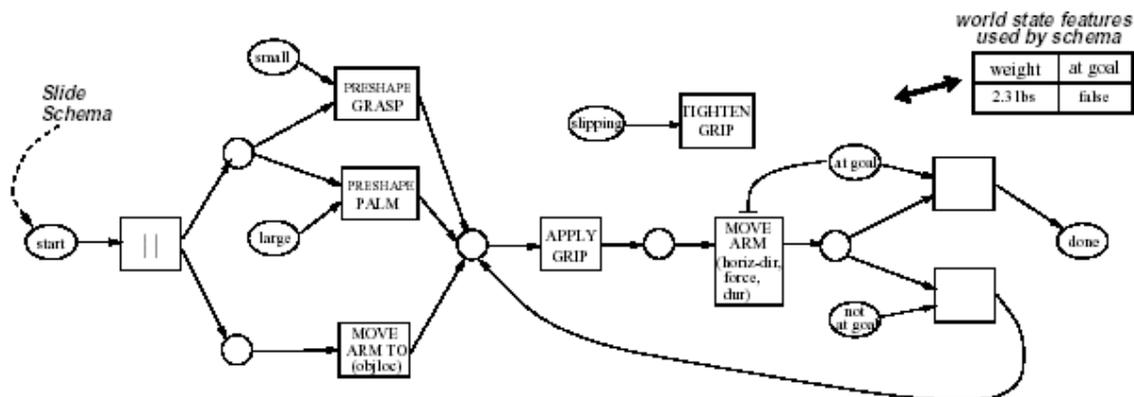


Figure 2.8: *slide* x-schema in Bailey et al. (1997)

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

Table 2.3: 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 simulate 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.

2.2. Multimedia fusion, coordination and presentation

According to Maybury (1994), Maybury and Wahlster (1998) the generation process of a multimedia presentation system can be divided into several co-constraining processes: the determination of communicative intent, content selection, structuring and ordering, allocation to particular media, realization in graphic, text, and/or other specific media, coordination across media, and layout design. We focus on the fusion and coordination of multimedia in this section because 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.

2.2.1. Intelligent multimedia authoring systems

Considerable progress has been made toward intelligent multimedia authoring systems which generate multimedia presentation (in particular visual) for limited domains (e.g. CUBRICON (Neal and Shapiro 1991)), communicative goals (e.g. COMET (Feiner and McKeown 1991a, b), and WIP (Wahlster et al. 1992)), and data (e.g. TEXPLAN (Maybury 1993) and CICERO (Arens

and Hovy 1995)). The advantage of integrating multiple media in output is achieved by effective multimedia coordination.

COMET (COordinated Multimedia Explanation Testbed) (Feiner and McKeown 1991a, b), is in the field of maintenance and repair of military radio receiver-transmitters. It coordinates text and three-dimensional graphics of mechanic devices for generating instructions about the repair or proper use by a sequence of operations or the status of a complex process, and all of these are generated on the fly. In response to a user request for an explanation, e.g. the user selects symptoms from its menu interface, COMET dynamically determines the content of explanation using constraints based on the request, the information available in the underlying knowledge base, and information about the user's background, discourse context, and goals. Having determined what to present, COMET decides how to present it in graphics and text generation. The pictures and text that it uses are not 'canned', i.e. it does not select from a database of conventionally authored text, pre-programmed graphics, or recorded video. Instead, COMET decides which information should be expressed in which medium, which words and syntactic structures best express the portion to be conveyed textually, and which graphical objects, style, and illustrate techniques best express the portion to be conveyed graphically. To communicate between multiple media, COMET uses two facilities: blackboard⁴ for common content description, and bi-directional interaction between the media-specific generators. Therefore, the graphics generator can inform the language generator about the graphical actions it has decided to use, and the language generator can then produce text like "the highlighted knob in the left picture".

Similar to COMET, WIP (Wahlster et al. 1992) is another intelligent multimedia authoring system that presents mechanical instructions in graphics and language for assembling, using, maintaining, or repairing physical devices such as espresso machines, lawn mowers, or modems. It is also supposed to adapt to other knowledge domains. The authors focus on the generalization of text-linguistic notions such as coherence, speech acts, anaphora, and rhetorical relations to multimedia presentations. For example, they slightly extend Rhetorical Structure Theory (Mann et al. 1992) to capture relations not only between text fragments but also picture elements, pictures, and sequences of text-picture combinations. More recent work of the authors focuses on interactive presentations, having an animated agent, called PPP persona, to navigate the presentation (André et al. 1996, André and Rist 2000).

TEXPLAN (Textual EXplanation PLANner) (Maybury 1993) designs narrated or animated route directions in a cartographic information system. It generates multimedia explanations, tailoring these explanations based on a set of hierarchically organized communicative acts with three levels: rhetorical, illocutionary (deep speech acts) and locutionary (surface speech acts). At each level these acts can be either physical, linguistic or graphical. Physical acts include gestures--deictic, attentional or other forms of body language, whilst graphical acts include highlighting, or zooming in/out, drawing and animating objects. A system designed to deliver explanations using each level of communicative act should be capable of explaining physically, linguistically or graphically, depending upon which type of explanation is best suited to the communicative goal. According to this communication theory, presentation agents (c.f. section 2.4) are the best embodiment of physical acts. Linguistic and graphical acts are usually the basic acts in conventional multimedia presentation systems.

4 A *blackboard* is a central repository in which a system component can record its intermediate decisions and examine those of other components. It is typically used for inter-component communication in a system.

CUBRICON (Calspan-UB Research centre Intelligent CONversationalist) (Neal and Shapiro 1991) is a system for Air Force Command and Control. The combination of visual, tactile, visual, and gestural communications is referred to as the unified view of language. The system produces relevant output using multimedia techniques. The user can, for example, ask, "Where is the Dresden airbase?", and CUBRICON would respond (with speech), "The map on the color graphics screen is being expanded to include the Dresden airbase." It would then say, "The Dresden airbase is located here," as the Dresden airbase icon and a pointing text box blink. Neal and Shapiro addressed the interpretation of speech and mouse/keyboard input by making use of an Augmented Transition Network grammar that uses natural language with gesture constituents. CUBRICON includes the ability to generate and recognize speech, to generate natural language text, to display graphics and to use gestures made with a pointing device. The system is able to combine all the inputs into the language parsing process and all the outputs in the language generation process.

CICERO (Arens and Hovy 1995) is a model-based multimedia interaction manager and integration planner. Its aim is to develop the model of an intelligent manager that coordinates and synchronizes the various media in a way that decreases system complexity caused by information overload. It is an application-independent platform tackling how to allocate information among the available media. In other systems, information allocation is usually devolved to the system designer. The general challenge CICERO attempts to respond to is how to build a presentation managing interface that *designs* itself at run-time so as to adapt to changing demands of information presentation.

These projects have studied problems in media design and coordination. The COMET project used a form of temporal reasoning to control representation and coordination whereas Maybury's TEXPLAN enables media realization and layout constraints to influence both content selection and the structure of the resulting explanation. These systems generate multimedia presentations automatically from intended presentation content. They can effectively coordinate media when generating references to objects and can tailor their presentations to the target audience and situation. The approaches of media coordination in these multimedia authoring systems inspire methods of CONFUCIUS' multimedia presentation planning which are discussed in detail in section 4.6.

2.2.2. Content selection

Selecting the content to present a story in traditional animation is an aesthetic task that requires the obvious abstraction or caricature of reality. Stanislavski's⁵ views compare Realism with Naturalism and also point out the principles of content selection in making traditional animations (Loyall 1997, p. 3): "Naturalism implied the indiscriminate reproduction of the surface of life. Realism, on the other hand, while taking its material from the real world and from direct observation, selected only those elements which revealed the relationships and tendencies lying under the surface. The rest was discarded." For instance, if there is a rifle hung on the wall in a description within a novel, film, or cartoon, whatever, traditional storytelling art form; it must be used later in the story in realistic art. This rule lightens our burden of simulating trivial objects in CONFUCIUS' story scenes whilst it requires more intelligence in the content selection module to evaluate the necessity of available ingredients in story input and knowledge base.

⁵ Constantin Stanislavski was regarded by many as the most influential actor and the thinker on acting in the 20th century.

2.2.3. Media preferences

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 have 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.

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 to CONFUCIUS (c.f. Figure 1.1) the system concerns dividing information and assigning 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. 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 (1991b) 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 be presented by both graphics and text (c.f. Figure 2.9 A), and the input is represented using the same formalism, a set of attribute-value pairs (c.f. Figure 2.9 B). The annotation is accomplished by unifying the task grammar (Figure 2.9 A) with the input (Figure 2.9 B). 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. In Figure 2.9, C is the result after unifying A and B.

```
((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 2.9: Functional unification formalism in COMET

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.

2.2.4. Coordination across media

Having solved the problem of content selection (What information should be presented?) and media selection (How to present this information?), we should deal with the integration and coordination problem, i.e. how should the presentation be arranged, *in space* and *in time*? In this section we discuss temporal coordination across media, the problem of space layout will be addressed in section 4.5 Animation generation.

Dalal et al. (1996) considered incorporating a temporal reasoning mechanism to control the presentation of temporal media (animation and speech) by managing the order and duration of communicative acts. Media coordination of CONFUCIUS concerns four issues: (1) temporal coordination between animation and speech (e.g. dialogue and lip movement), (2) cross-references, (3) coordinating voiceover breaks with animation shot breaks, and (4) duration constraints for different media. Similar to static multimedia presentation, the cross-reference in temporal media representation resolves identification of referents. For instance, a voiceover could refer to several characters 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.

2.2.5. Consistency of expression

Consistency of expression is one of the basic requirements for presenting a story realistically. It encompasses the coordination among media and the consistency within one medium. The coordination across media could be between voiceover narration and animation, between background music and situation appearing in animation and so on. Consistency could be within different parts of graphics as one medium. Holistically, consistency of expression requires the plot, the development of the story and the growth of characters, etc. to be coherent. Atomistically, because every character has many avenues of expression, for example an actor has facial expression, body posture, gesture, movement, voice intonation, etc., at every moment all of these avenues of expression must work together to convey the unified message that is appropriate for the character's personality, feelings, situation, and thinking.

2.3. Automatic text-to-graphics systems

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 description can be generated from a linguistic description as in the *Spoken Image* system (Ó Nualláin and Smith 1994). In Nenov and Dyer (1988), a linguistic description of objects is visualized to 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. In this section, recent progress in automatic text-to-graphics (language visualisation) systems is discussed.

2.3.1. WordsEye

WordsEye (Coyné and Sproat 2001) is able to 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 2.10 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 2.10: 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 a version of Michael Collins' (1999) parser. The parser output is then converted to a dependency structure. Then 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 depicitors representing objects, poses, spatial relations, and other attributes. Transduction rules are applied to resolve conflicts and add implicit constraints. Finally, the resulting depicitors 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 2.11 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 the 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.

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 will allow one to depict different statements, such as “John threw the egg” or “Mary tossed the small toy car”.

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 that WordsEye does not cover is coordinating temporal media, e.g. speech and animation, where information is presented over time and needs to be synchronized with other media.

```
(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 2.11: Verb frames of *say* in WordsEye

2.3.2. ‘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 cauldron bubbles, a book pages turn, a letter flies to a mail box. The term MICONs (moving/animated icons) was first coined by Russell 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 CD (as discussed in section 2.1.4). 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 2.12 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 2.12 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. 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.

2.3.3. Spoken Image (SI) and SONAS

In Ó Nualláin and Smith’s (1994) Spoken Image (SI), a 3D dynamic graphic visualisation with an almost photo-realistic image quality 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 can be the following:

```
(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.)

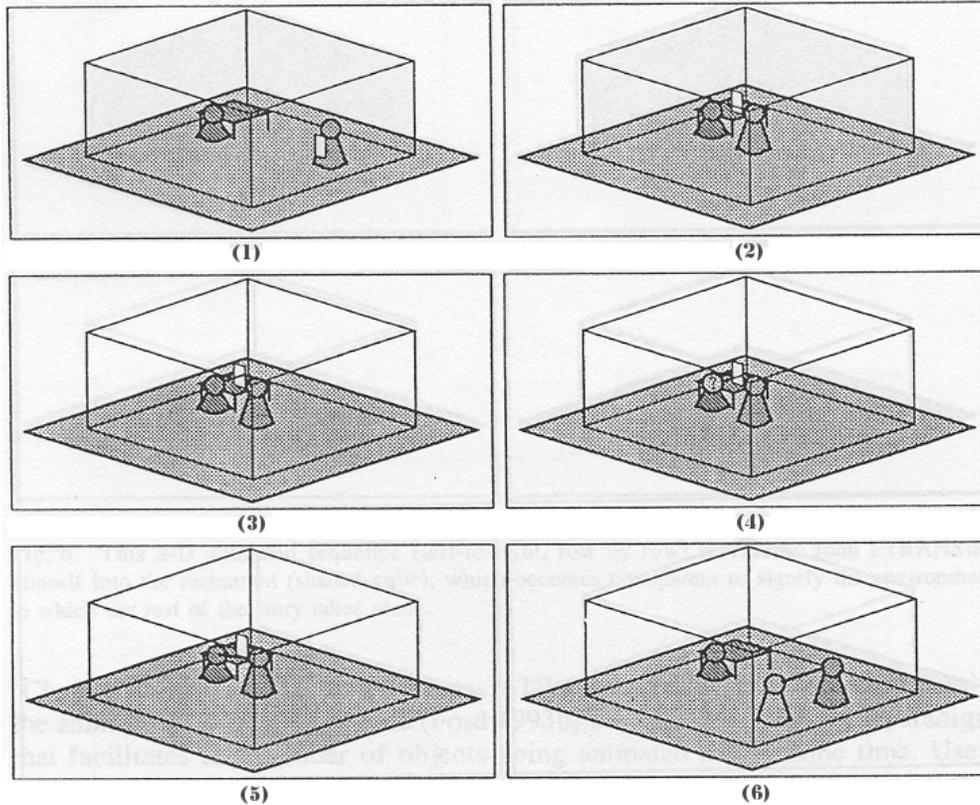


Figure 2.12: An example story: *going to a restaurant*

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 2.13). 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 2.13.

```

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 2.13: Motion classes in SONAS

A deficiency of SI and SONAS is lack of knowledge of naïve physics, 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.

2.4. Multimodal storytelling

In *Tell Me a Story*, 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 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 builds the theory basis of computer storytelling. 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. 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 designer 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.

2.4.1. AESOPWORLD

Different from most storytelling systems that focus more 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, and takes action to execute the plans. He 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 behaviors are shown by 2D graphic displays, a speech synthesizer, 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-synthesized 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.

2.4.2. 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 unusual world for fantasy play. 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 (see Figure 2.14).



(Left) Everyone rows on the correct side and the rock is avoided.

(Right) Room: "Finally, we've come to land. Push the boat towards the trees, onto the sand."

Figure 2.14: KidsRoom

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' behavior (e.g. coaxing them to certain locations) to facilitate its own vision processes.

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*. 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.

This approach of event loop and timers gives ideas of event synchronization in CONFUCIUS. However, 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.

2.4.3. 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 toward 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 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 (see Figure 2.14). 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 2.15 checks a timer first, if the condition is met, move and rotate the viewpoint of the virtual observer (i.e. the user/player). 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.

Although not implemented in the system, Larsen and Petersen intended to use autonomous agents as actors in the story and apply *behavior models* to their action selection. The behavior models give the autonomous actors ‘a life of their own’ in the virtual world and reduce the work of story generation.

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

Figure 2.15: Executable story script of Larsen’s storytelling system

2.4.4. 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 2.16 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.

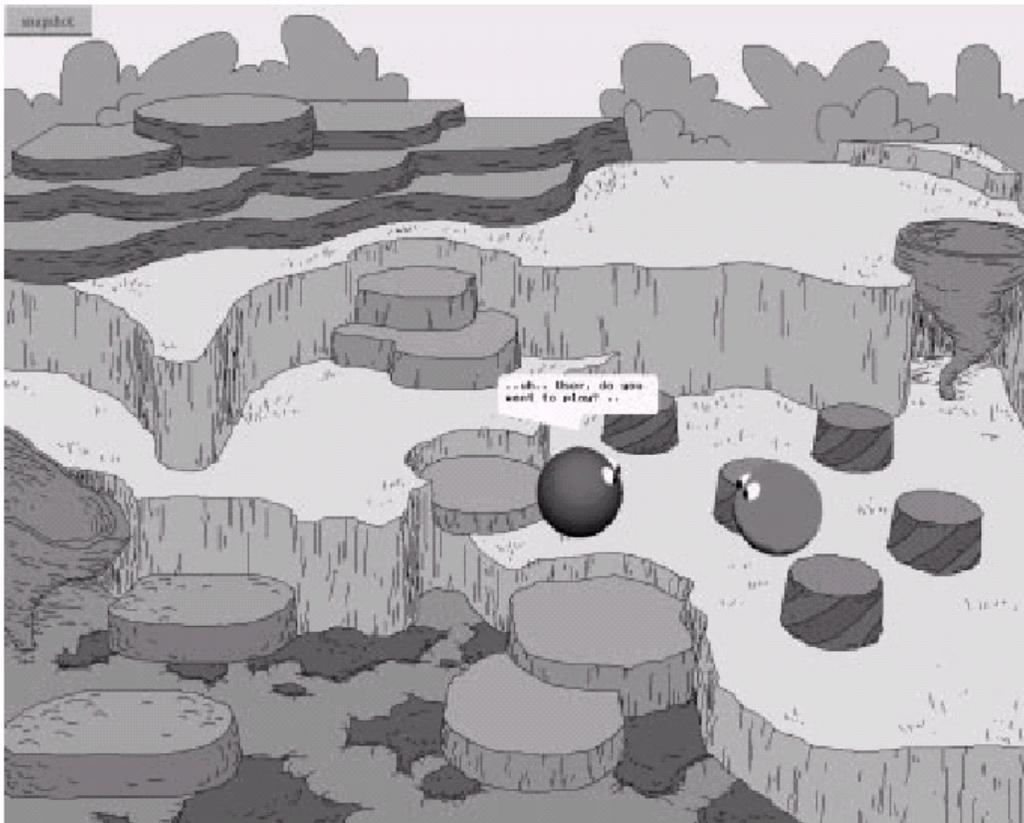


Figure 2.16: The *Edge of Intention* world in OZ project

Primitive actions are specified by names and zero or more 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`, etc. 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 2.16). 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 behaviors such as greeting another agent, sleeping, going to a place in the world, etc. Behaviors 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 behaviors, the subgoals they give rise to, and their behaviors are all written by the story author, and comprise a large part of the personality of the agent.

Oz focusses on simulation of behavior, emotion, and personality of autonomous agents in storytelling which makes characters in a story more believable. Although CONFUCIUS does not need to build autonomous characters on its own initiative because they are defined in the input text, the methods of animating characters in Oz gives ideas on visualization of characters for CONFUCIUS.

2.4.5. Virtual theater and Improv

Hayes-Roth and her colleagues built a system for improvisational theater (Hayes-Roth and van Gent 1997). It includes semi-autonomous agents that can be directed by children to build stories or improvisational experiences. Their agents can interpret the situation, provide choices to the children and cause the agents to perform the chosen behaviors. Her system uses Hap (Loyall 1997) to realize the behaviors (another version uses the Improv system), but Hap agents have no motivational layer, and in fact no autonomy; instead, they are controlled by a script provided by the children or by the children’s direct control using the interactive interface the system provides. In addition to the behaviors provided by Hap, the system also allows speech actions by playing one of 250 prerecorded phrases. This system makes for a powerful “stage” in which children can direct and improvise. Since the agents are not autonomous, they just carry out the children’s direction.

Improv, the work of Perlin and Goldberg (1996) emphasizes agents’ believable movement. Perlin has built an animation system based on ideas from his work in procedural texture synthesis. This approach produces compelling movement in human figures. Originally this work created *puppets* that were controlled by a person choosing which behavior the puppet performed at any given time. More recently it has been progressing toward autonomy, and this animation system has been extended to support scripted agents. The focus of Improv is on providing tools that allow artists to create powerful scripted scenes with these agents.

In brief, 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.

2.4.6. Computer games

Interactive storytelling (or story generation) is 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. Latest 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 with text games. Virtual reality is the most recent technique of interactive action games (Qvortrup 2001). A user who logs into an action game participates in the world, explores it, and might fulfil a task via his 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. Making and marketing includes a storyboard for the gangland adventure *Grand Theft Auto* taken straight from film studios, which leads to a consideration of storytelling. 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, according to 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. The following two areas are what CONFUCIUS is intended to enhance: it will present stories in multiple modalities and embed story understanding to automate animation control.

2.5. Intelligent multimedia agents

Embodied animated agents, either based on real video, cartoon style drawings or model-based 3D graphics, 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 as we discussed in section 1.2.2, creating believable and realistic actors is the crucial task of impressive storytelling. It is practical to turn agents into actors for storytelling because agents' looks, the way they moves and how they expresses their intentions are similar to those of actors 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 both cartoon style agents like *PPP* (André et al. 1996) and Disney animation, and model-based 3D agents like *REA* (Cassell et al. 2000) are

discussed. Some focus on the agents' behaviour model and personality (REA and Disney's characters), others focus on the agents' psychological model (*AESOPWORLD*, c.f. section 2.3.1.) or multimodal human-computer communication as in *Gandalf* (Thórisson 1996).

2.5.1. BEAT and other interactive agents

Cassell et al. (2000) discuss REA (Real Estate Agent), which 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. 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 2.17a.

REA integrates a natural language generation engine (SPUD), and an animator's tool, BEAT (Figure 2.17b), 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 realize written text in spoken language (McTear 2002) BEAT realizes written text in embodied expressive verbal and nonverbal behaviors such as face expression, head nods, gaze, and hand gestures⁶. 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.

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.



(A) User interacting with REA



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

Figure 2.17: REA and BEAT

⁶ This is also a principle of the Disney animators for animating expressions and movements of a character to concomitant dialogue.

2.5.2. Divergence on agents' behavior production

The linkage between agents' speech and behaviors which has been explored in synthesizing 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. Now we will discuss these opinions in detail.

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 2.18 presents a fragment of dialogue in which two animated agents' gesture, head and lip movements, and their inter-synchronization 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 2.18: Dialogue between two autonomous agents

The gestures and facial expressions depend on sentence type (declarative or interrogative), meaning (affirmative or negative), stressed words (italic words in the example) etc. 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 realization 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. You may 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 synthesized animation. Gesture and expressional behavior 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 in CONFUCIUS.

2.5.3. Gandalf

Thórisson (1996) has built a system that addresses many issues in face-to-face communication. His agent, *Gandalf*, is rendered as a computer-animated face and associated hand. Gandalf is the interface of a blackboard architecture called *Ymir* 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 (see Figure 2.19).

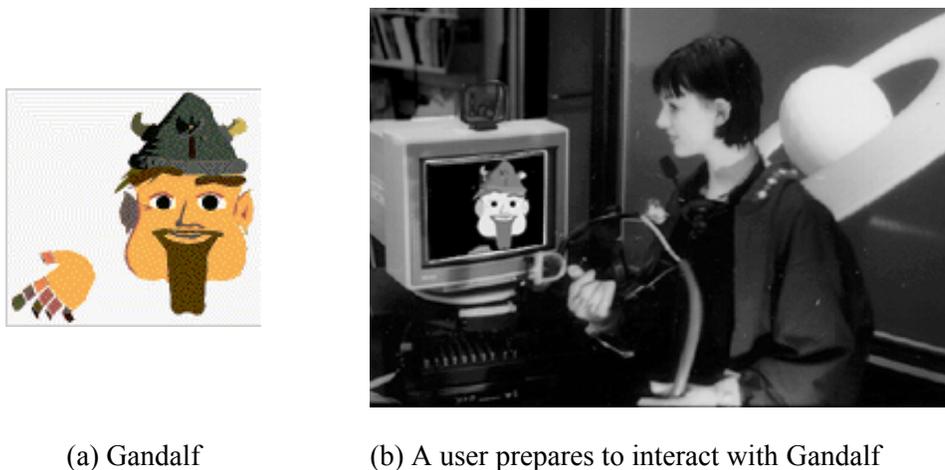


Figure 2.19: Gandalf's interface

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 his 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 also 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 agents.

2.5.4. PPP persona

PPP Persona (Personalized Plan-based Presenter) (André et al. 1996) is an animated presentation agent that can be used for showing, explaining, and verbally commenting textual and graphical output on a multimodal user interface. Figure 2.20 shows an example in one PPP Persona domain of advertising accommodation offers found on the Internet. The system retrieves matching information to the user's request from the Web and creates a presentation script for the PPP Persona which is then sent to the presentation viewer (Netscape Navigator in the example, including a Java interpreter). When viewing the presentation PPP Persona highlights the advantages of the accommodation by means of verbal annotations, i.e. Persona points to part of the picture during a verbal utterance.

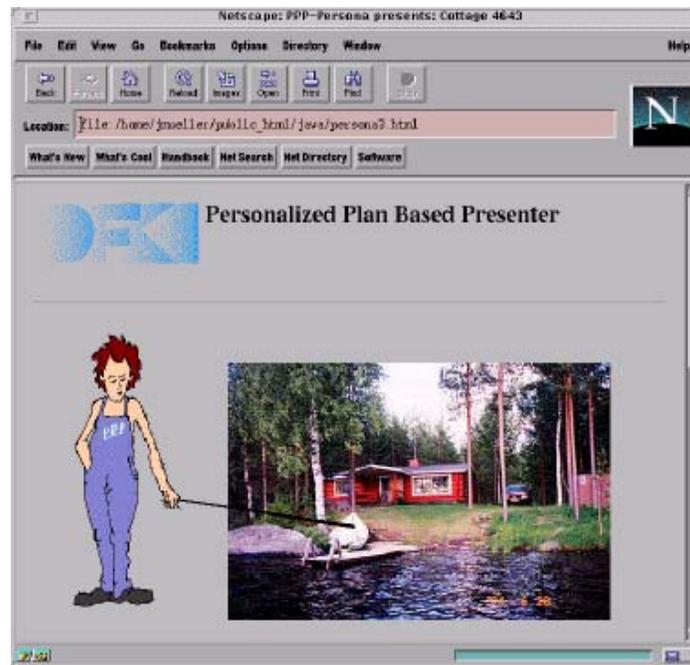


Figure 2.20: PPP persona presents retrieval results from the Internet

According to its functional roles in a presentation the persona must be conversant with a broad variety of presentation gestures and rhetorical body postures. PPP Persona has a built-in mechanism for *action specialization* and *action decomposition*. The concept of action decomposition is complemented by the concept of specialization, i.e. high-level persona actions can be decomposed to lower level ones and down to the basic postures/acts. Using the action specialization and decomposition mechanism applications can request presentation tasks at a high level of abstraction, e.g. the PPP Persona server can find reasonable positions from which pointing gestures are executed.

The work in dynamical interface agent behaviours gives some ideas in creating animated actors in stories and the narrator who speaks over stories in CONFUCIUS, especially on the behavioural aspects, i.e. to produce reasonable behaviours and natural manner such as expression of emotions, behavior planning (e.g. coordination of directions, idle-time and improvised behaviour), and character modelling/rendering (e.g. realization of animated characters).

2.6. Intelligent multimedia interfaces

Typical intelligent multimedia dialogue systems such as Put-that-there (Bolt 1987), AIMI (Burger and Marshall 1993), AIFresco (Stock et al. 1993) and XTRA (Wahlster 1998) discussed in this section parse integrated input and generate coordinated multimodal output. Although the input of CONFUCIUS is unimodal (language modality), the multimedia fusion and coordination methods such as cross-media communication and reference in those systems are important for CONFUCIUS to generate multimodal output. Maybury (1993) gives a general overview of current techniques for intelligent multimedia interfaces. Key issues of intelligent multimedia interface currently include: (1) Coordination of multiple modalities, both in terms of input and output, (2) Planning and realization of multimedia presentations, (3) Media-dependent and media-independent meaning representation languages, (4) Architectures for multimedia interfaces, (5) Discourse and user models for multiple modalities, (6) Qualitative/quantitative measures for evaluation of multimedia interfaces, and (7) Philosophical and psycholinguistic models of multimodal interaction.

“Put-That-There” (Bolt 1987) is one of the first intelligent multimodal interfaces. The interface consisted of a large room, one wall of which was a back projection panel. Users sat in the centre of the room in a chair wearing magnetic position sensing devices on their wrists to measure hand position. Users could use speech, and gesture, or a combination of the two to add, delete and move graphical objects shown on the wall projection panel. Although nearly 20 years old, a video of their work is still an impressive example of an effective multimodal interface because they discovered that by integrating speech and gesture recognition with contextual understanding, neither had to be perfect provided they converged on the user’s intended meaning. In this case the computer responds to users’ commands by using speech and gesture recognition and taking the current context into account.

AIMI (An Intelligent Multimedia Interface) (Burger and Marshall 1993) is aimed at helping users devise cargo transportation schedules and routes. To fulfil this task the user is provided with maps, tables, charts and text, which are sensitive to further interaction through pointing gestures and other modalities. AIMI uses non-speech audio to convey the speed and duration of processes which are not visible to the user.

AIFresco (Stock et al. 1993) is an interactive, natural language centred system for presenting information about Fourteenth century Italian frescoes and monuments. It combines natural language processing with hypermedia to provide an efficient and user-oriented way to browse around through a pre-existing hyper-textual network. The combination of natural language and hypermedia gives rise to an advantage that while prefabricated hypermedia texts can compensate for a lack of coverage in the natural language modules, the on-demand generation of user-tailored texts can help to overcome the disorientation problem in the hypermedia environment. Also, further information about pictures and videos can be accessed by asking questions in natural language combined with direct pointing actions like in AIMI.

XTRA (eXpert TRANslator) (Wahlster 1998), a multimedia interface in the tax form domain, combines language with pointing gesture input. The user and system could both refer to regions

in a tax form, without a pre-definition of pointing-sensitive areas. XTRA generates natural language and pointing gestures automatically, but relies on pre-stored tax forms. It represents not only the linguistic context, but also maintains a data structure for graphics to which the user and the system may refer during the dialogue. In the tax domain, the graphical context corresponds to a form hierarchy which contains the positions and the size of the individual fields as well as their geometrical and logical relationships. Furthermore, connections between parts of the form, e.g. *region30*, and the corresponding concepts in the knowledge base, e.g. *employer1*, are explicitly represented.

Like the intelligent multimedia authoring systems that we discussed in section 2.2, approaches of media coordination and presentation planning in these intelligent multimedia interfaces are helpful for CONFUCIUS' multimedia storytelling.

2.7. Non-speech audio

This section specifically considers the use of non-speech audio in intelligent storytelling. 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.

Bly et al. (1987) mention two dimensions of non-speech audio, the data dimension and the sound dimension. The data dimension consists of the actual data which is being encoded by the sound, i.e. what info is being encoded? The sound dimension describes the actual quality of the sound, i.e. what is it encoded in? This sound dimension may be further divided into sound parameters: waveform, frequency, amplitude, spatial location, duration, timbre, and timing.

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 to synthesized music.

2.7.1. 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 telephone ring. 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 are also a major source of non-speech audio in CONFUCIUS.

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.

Auditory icons are important for CONFUCIUS. Certainly the intuitiveness of this approach to auditory display will result in more wholesome story presentation.

2.7.2. Earcons

Earcons are melodic sounds, typically consisting of a small number of notes, with musical pitch relations (see 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, program error etc. 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.

2.7.3. 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 synthesizing 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: synthesized sound used as an aid to data visualisation (especially abstract quantitative data), for program comprehension, and monitoring performance of parallel programs, etc.

2.7.4. Music synthesis

In synthesized 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 (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 the Serial music techniques, the application of music grammars (notably the General Theory of Tonal Music by Fred Lerdahl and Ray Jackendoff), 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.

2.7.5. Non-speech audio in CONFUCIUS

Figure 2.21 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 synthesized music can generate audio from other modal data; and earcons and synthesized music are melodic sound. Auditory icons and music synthesis will be used as non-speech audio in CONFUCIUS, such as real world sound accompanying animated events in the story being told, and selection of pre-recorded waveforms instead of *real* music synthesis. CONFUCIUS will select music clips from a database according to the story situation rather than synthesizing music on the fly in order to simplify the audio modality design and focus our work on animation and multimodal fusion, as we saw with AESOPWORLD (section 2.4.1) which *plays* music according to the protagonist's emotional state rather than *generating* it.

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 one task in the multimodal fusion section of CONFUCIUS and was discussed in section 2.2.

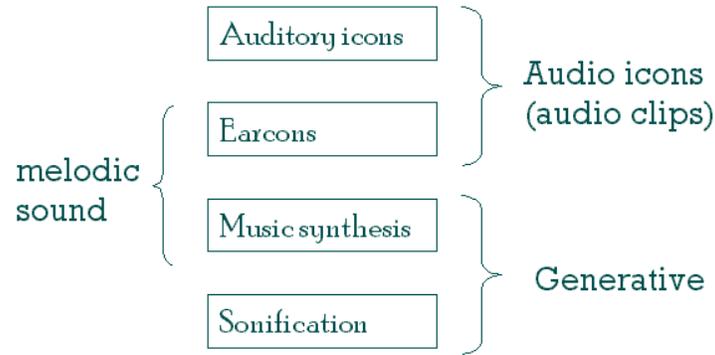


Figure 2.21: Four types of non-speech audio and their common features

2.8. 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 2.22 illustrates the mental architecture and meaning processing widely accepted in cognitive science. Ellipses denote meaning processing and rectangles denote results of each level.

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 experience them and *present* 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 perceptual 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 (discussed in section 2.1.4). Beyond the level of represented

situations in the architecture of human mind is abstract thinking such as discourse-based or symbolic reflection.

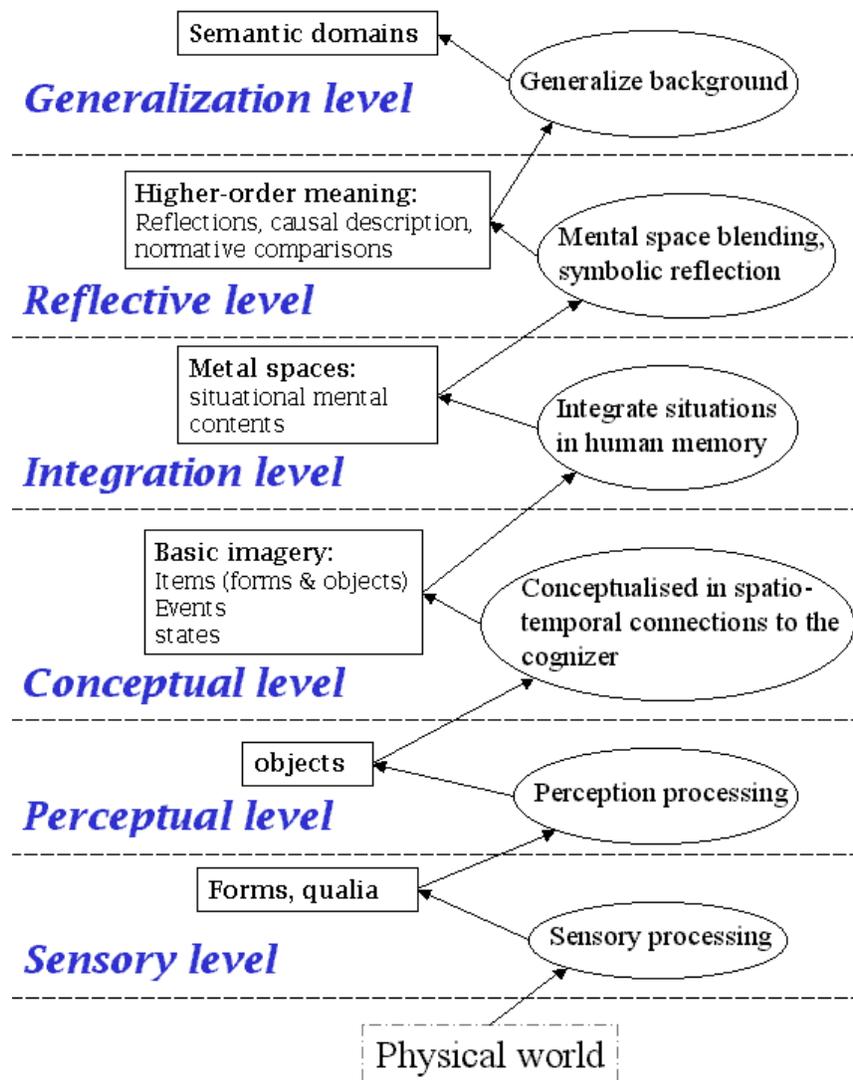


Figure 2.22: Mental architecture and meaning processing

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.

Figure 2.23 depicts the model of the processes of cognition, communication and re-cognition. A language visualisation system like CONFUCIUS 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.

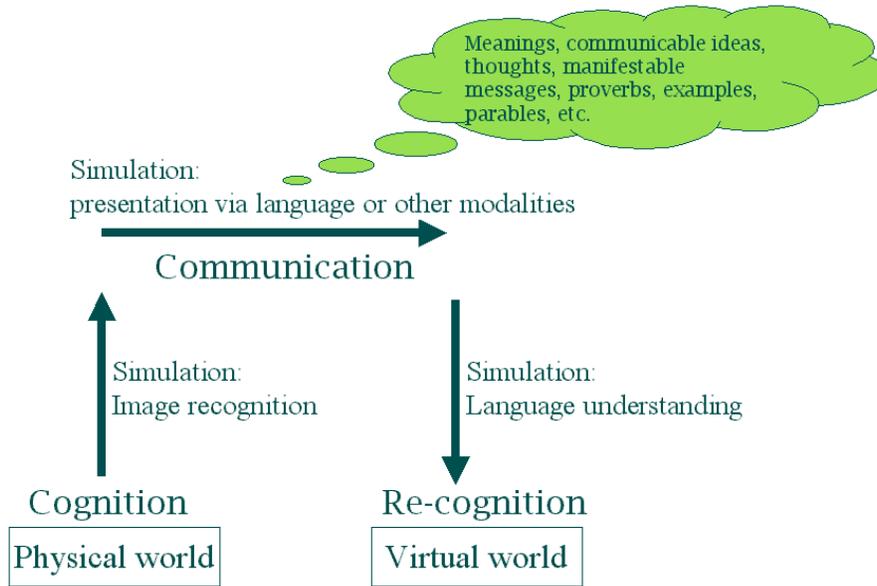


Figure 2.23: Simulation of cognition, communication, and re-cognition

3. Multimodal semantic representations

For the purposes of multimodal interpretation, realization and integration in intelligent multimedia systems, traditional multimodal semantic representations such as frame-based representation and XML representation (see section 2.1) have some common features. They should support both interpretation and generation, support any kind of multimodal input and output, and support the variety of semantic theories. Usually a multimodal semantic representation has four basic components: (1) temporal structures (events), such as dialogue turns/discourse context, gestures, actions on the task, (2) referential structures, i.e. individuals and objects participating in an event, which comprises spatial structures, (3) restrictions on temporal and referential structures, e.g. gesture types, linguistic modifiers, Dialogue acts, and (4) dependency structures (linking events and referential structures), e.g. participant roles (AGENT-SOURCE-GOAL). A multimodal representation may contain architectural, environmental, and interactional information. Architectural representation indicates producer/consumer of the information, confidence, and devices. Environmental representation indicates timestamps, spatial information (e.g. speaker's position, graphical configurations, gestural trajectories). Interactional representation indicates speaker/user's state or other addressees.

Existing multimodal semantic representations within various intelligent multimedia systems may 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 representation that is capable of connecting meanings across modalities. Such a lower-level meaning representation, which links language modality to visual modality, is proposed in this chapter. Figure 3.1 illustrates the multimodal semantic representation of CONFUCIUS. It is composed of language, visual and non-speech audio modalities. Between the multimodal semantics and each specific modality there are two levels of representation: one is a high-level multimodal semantic representation which is *media-independent*, the other is an intermediate level *media-dependent* representation. CONFUCIUS will use an XML-base representation for high-level multimodal semantics and an extended predicate-argument representation for intermediate representation which connects language with visual modalities as shown in Figure 3.1.

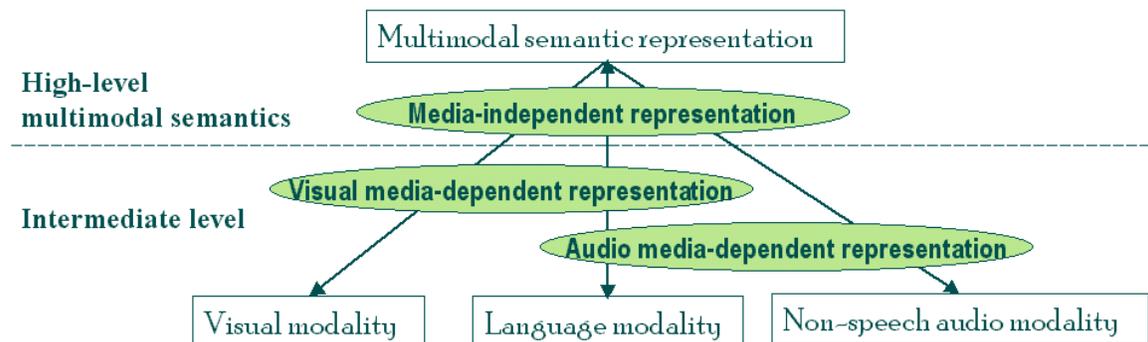


Figure 3.1: Multimodal semantic representation in CONFUCIUS

There is no dearth of language semantic representations for linguistic information. As we discussed in section 2.1, common *linguistic semantic representation*⁷ includes FOPC, semantic networks and frames. Most linguistic semantic representations in natural language processing represent meaning on the sentence level or phrase level, and are used for purposes like question-answering, information retrieval, and image recognition. The framework proposed in this chapter can represent visual semantics on the word level (action verbs, concrete nouns, and adjectives) and is suited for computer graphic generation from natural language input, i.e. the media-dependent representation in Figure 3.1. The representation is suited for implementation in the 3D graphic modelling language VRML. It will be translated to VRML code by a Java program in CONFUCIUS. We also plan to include non-speech audio (c.f. section 2.7) in the media-dependent and media-independent semantic representations. The following subsections focus on the visual knowledge representation of prefabricated objects and events, i.e. the knowledge base, of CONFUCIUS.

3.1. Visual semantics in a general purpose knowledge base

The knowledge representation required for the tasks of CONFUCIUS must provide the capabilities: (i) model both declarative and procedural knowledge, (ii) inference mechanisms (such as classification based inference). Figure 3.2 illustrates the design of the knowledge base of CONFUCIUS, which is also a general design of knowledge base for any intelligent multimedia applications which include both natural language processing and vision processing. It consists of language knowledge, visual knowledge, world knowledge and spatial and qualitative reasoning modules. Language knowledge is used in the natural language processing component to extract concept semantics from text. Visual knowledge consists of the information required to generate moving image sequences. It consists of *object model*, *functional information*, *event model*, *internal coordinate axes*, and *associations between objects*.

The *Object model* has semantic representation of categories (nouns), the *event model* has semantic representation of motions (verbs), and the *internal coordinate axes* are indispensable in some primitive actions of *event models* such as rotating operations, which require spatial reasoning based on the object's internal axes. In VRML, the internal coordinate axes of objects can be represented by a Transform node with a corresponding value of `SFRotation` type.

Here we focus on efficient semantic representation of *event models* in the typical knowledge base (Figure 3.2) for natural language and vision integration systems. The *event model* in visual knowledge requires accesses 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 a man who performs the action “cut”. It also needs *function information* of “knife”, the *internal coordinate axes* information of “knife” and “cake” to decide the direction of the movement. To interpret the verb `wear(x, y)`⁸, the *event model* needs access the *object model* of y, which might be a hat, a ring, a pair of glasses, or shoes, and its *function model* that concerns its typical location (e.g. hat on the head, ring on a finger etc.).

⁷ Here we use “linguistic semantic representation” to differentiate with “multimodal semantic representation”. Typically, without any modifier “semantic representation” means semantics for natural language processing.

⁸ Means x wears y. x is a person or personated character, y is an object.

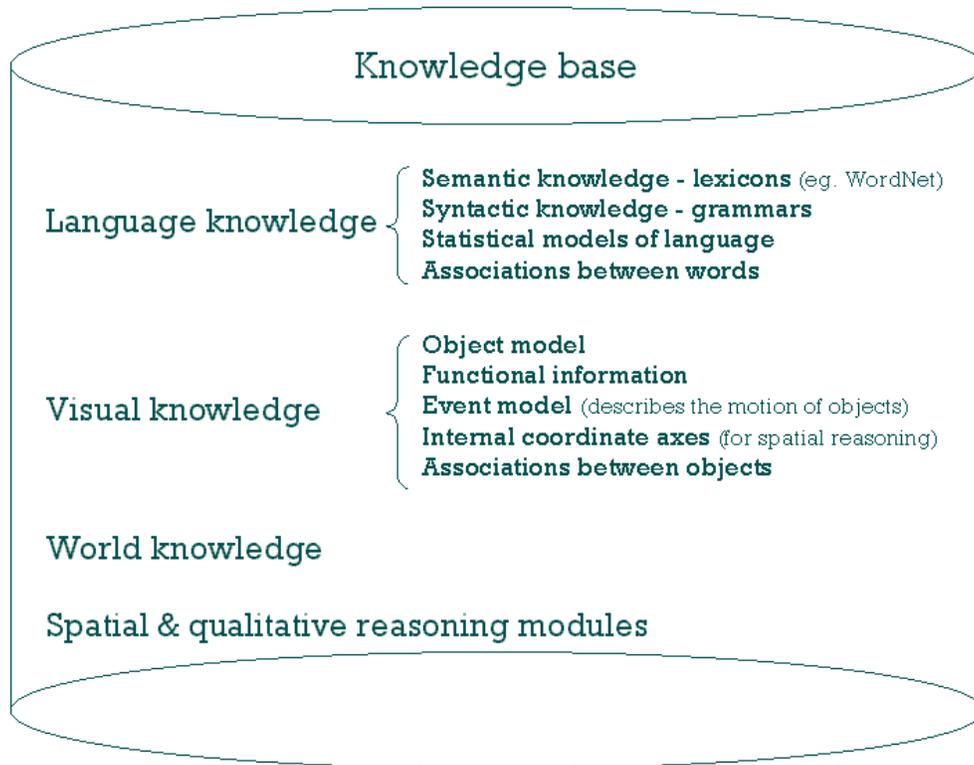


Figure 3.2: Knowledge base of CONFUCIUS

The knowledge base and its visual knowledge representation presented in this section are expected to be used in CONFUCIUS, and it could also be adopted in other vision and natural language processing integration applications.

3.2. Base concepts--Equivalences across part-of-speech

Base concepts are concepts on the top levels in semantic networks which make up the core of the networks. Base concepts can be grouped into coherent semantic clusters, called *top-ontology*, which is typically used to extract semantic distinctions applying to situations cut across parts of speech, i.e. they apply to both nouns, verbs and adjectives. Although the following sections (3.3-3.6) are organized based on part-of-speech, the criteria to differentiate them are actually *base concepts*. Therefore, they are semantic categories rather than syntactic ones. There are many equivalences across parts of speech (called XPOS⁹ relation 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; etc. This taxonomy is necessary because words from different parts of speech can be related in the semantic networks via a XPOS_SYNONYM

⁹ 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.

relation, and the entries in CONFUCIUS' graphic library can be related to any part-of-speech. Therefore, action verbs and action nouns are treated as events in section 3.4, and descriptive adjectives and their corresponding abstraction nouns are treated as attributes in section 3.5, etc.

The hierarchy shown in Figure 3.3 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.

Consider from the prospect of multimodal presentation generation, 1stOrderEntities are suitable to be presented in static visual modalities (still pictures), 2ndOrderEntities are suitable to be displayed in dynamic visual modalities (animation, video), and 3rdOrderEntities are suitable to be expressed in language and other modalities such as non-speech audio since they are unobservable by visual sensor. According to this classification, the following sections can be grouped into three: the 1stOrderEntities cover concrete nouns (section 3.3); *static situations* in the 2ndOrderEntities concern either properties of entities or relations between entities in a 3D space, i.e. adjectives and prepositions (section 3.5 and 3.6); *dynamic situations* in the 2ndOrderEntities cover either events or their action manners, i.e. verbs (section 3.4); and the 3rdOrderEntities are covered as *non-action verbs* in the verb section.

3.3. Categories of nouns for visualisation

In this section we classify nouns from the visual semantic perspective. We first investigate sub-categories of nouns in other semantic top-ontology work and then give our categorization for CONFUCIUS. WordNet 1.5 (Beckwith et al. 1991), one of the most widely used lexical resources, has its semantic networks (*synsets*¹⁰) representing some major semantic clusters per parts-of-speech. Figure 3.4 shows the major semantic clusters of nouns in WordNet.

There are some *hypernym* or *hyponym* relations between some synsets which result in categories overlapping. 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 (included in Figure 3.4 and 3.5).

¹⁰ Set of synonymous word meanings (synset members).

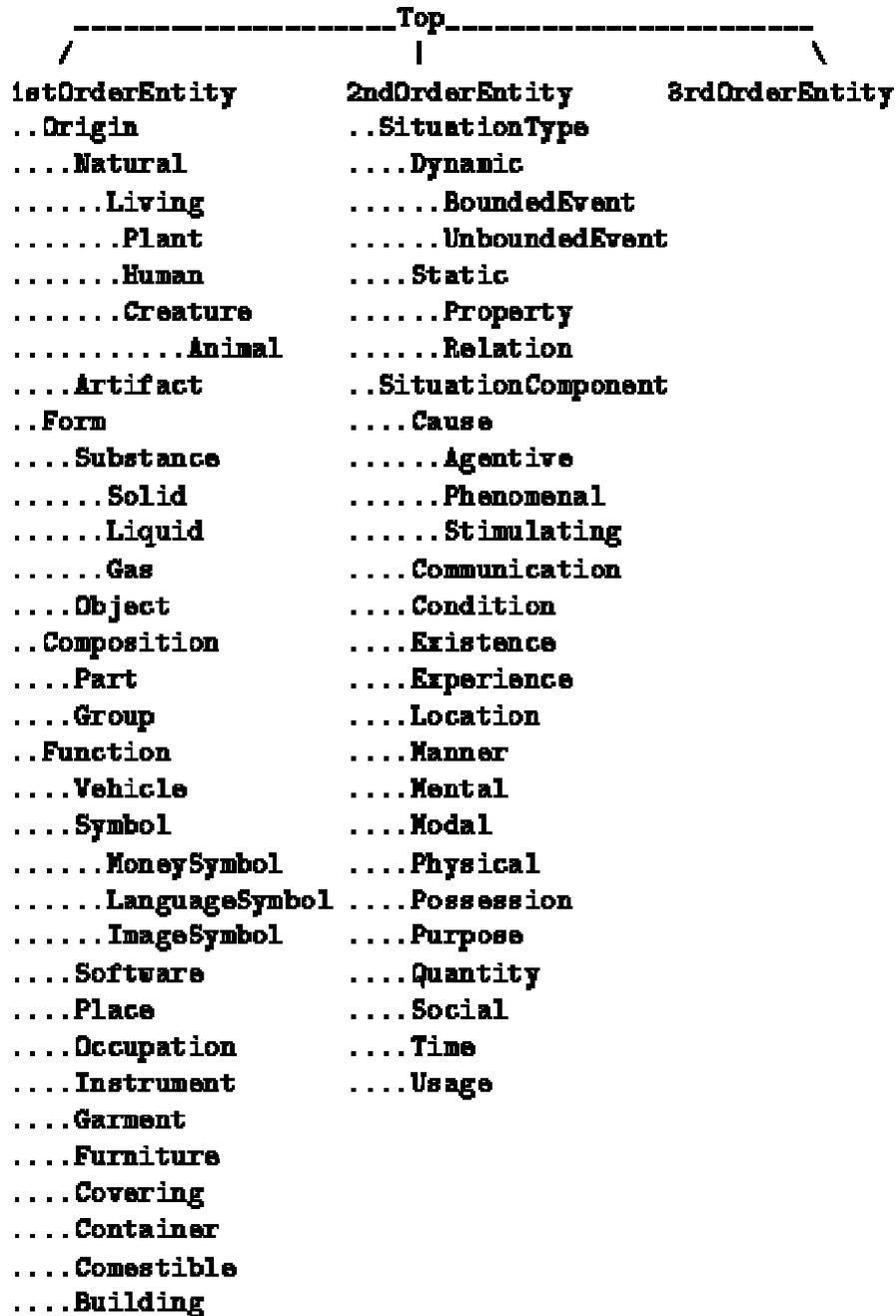


Figure 3.3: Hierachy of Top Concepts in EuroWordNet

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 3.4: Major semantic clusters of nouns in WordNet

There may be one or more synsets in Figure 3.4 which have no hypernym and therefore represent the tops 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 3.5 lists the eleven top noun categories.

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 3.5: Noun Tops in WordNet

Having analysed these eleven noun tops of WordNet, we classify concrete nouns into four top categories (see Figure 3.6) to suit for our visualisation purpose. It shows that visual semantic representation of concrete nouns concerns four issues: (1) the existence of the entity, i.e. its physical features like 3D size and color, (2) its position in a three dimensional space, (3) mass nouns and grouping, and (4) possession.

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*. The first concept group is *event*, i.e. category 1 (event) and 8 (act, humanaction, humanactivity) in Figure 3.5, which is discussed in the next section (section 3.4). The other group concerns *properties, features, or states* of entities, which cannot be seen in the same way as concrete nouns in Figure 3.6, i.e. category 2

(psychological feature), 3 (abstraction), 5 (shape, form), 6 (state), and 11 (phenomenon) in Figure 3.5. They are discussed in section 3.5.

1. entity
2. location, space
3. group, grouping
4. possession

Figure 3.6: Concrete noun categories in CONFUCIUS

3.4. Visual semantic representation of events—meaning as action

Traditional semantic representation of first order predicate calculus is used at sentence/phrase level, 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)` etc., while in semantic representations on physical aspects (see Table 2.1), both 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; event-logic restricts only in movement recognition applications; and x-schema requires learning phase before it can carry out an action. In this section we extend traditional predicate calculus to the word level to represent visual semantics of events (action verbs) and overcome the limitations of previous physical semantic representations.

3.4.1. Categories of events in animation

Like we did for nouns in section 3.3, in this section we classify verbs from the visual semantic perspective. Since verbs are core of events, verb subcategories are significant for visualisation of events. 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 *subcategorization frame*, i.e. possible sets of complements the verbs expect (see Table 3.1). For instance, a verb like *find* subcategorizes for an NP, whereas a verb like *want* subcategorizes for either an NP or a non-finite VP. These possible sets of complements of a verb are also called the *subcategorization frame* for the verb. Here we subcategorize verbs in animation from the visual semantic perspective, as shown in Figure 3.7, albeit the classification has overlays with linguistic subcategorization.

<i>Subcategorization frame</i>	<i>Verb</i>	<i>Example</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 3.1: Some linguistic subcategoriation frames and example verbs

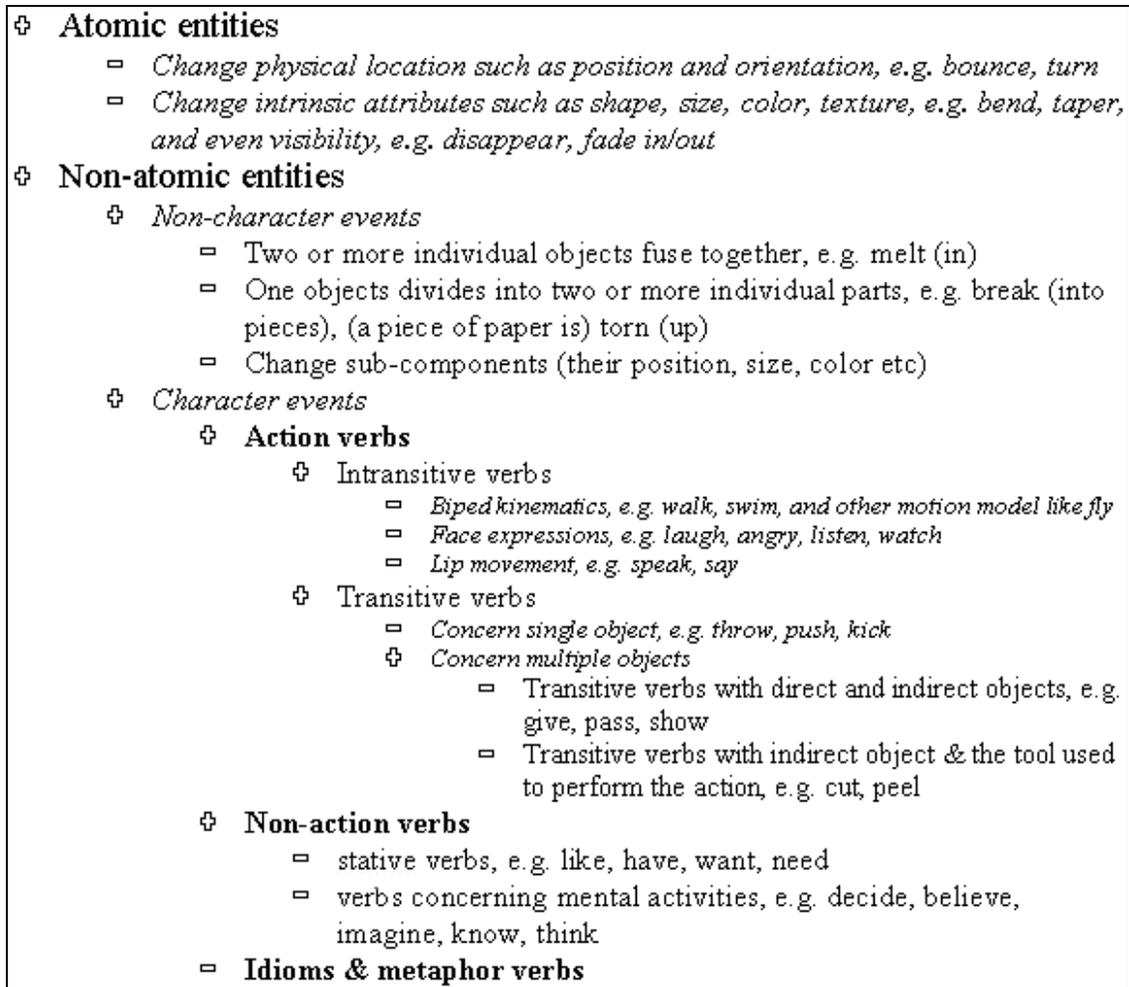


Figure 3.7: Categories of events

First we divide all events into events on atomic entities and events 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 and those changing objects' intrinsic attributes like shape, size, and color. In the non-atomic group, we classify them according to whether they concern a character. Those events based on characters can next be divided based on whether or not they are easily observable. Action verbs are easily observed and hence the major part of events in animation. In its sub-type transitive verb, the movement of the subject who performs the action usually could be modelled by biped kinematics, and the movement (or change) of the objects or tools could be modelled by *event models*. Verbs working on atomic entities can also work on non-atomic objects, for instance, *disappear/vanish* can work on both atomic and non-atomic objects (the Cheshire cat in the following example 1) or component(s) of non-atomic objects like *turn* in the following example 2.

1. 'All right,' said the Cat. And this time it *vanished* quite slowly, beginning with the end of its tail and ending with its grin.
2. He *turned* his head and looked back.

Non-action verbs and verbs with metaphor meanings are not easily observable. They are hard to describe by physical changes and however are common in even simple stories like children's stories. 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. Since CONFUCIUS is multimodal these types of verbs could be presented via speech modality if not obvious to modal visually.

Since the tasks of representing transitive verbs can be divided into two sub-tasks: modelling biped kinematics for the subject, and modelling atomic entities changes for the objects and tools, representing kinematics becomes the main task in visual models. There are two ways to describe a biped's kinematical motion. One method is to model different body parts as distinct objects in *object modelling* (see Figure 3.2), e.g. forearm and upper arm, leg and shin are individual objects which attach together, *event modelling* then transforms these individual objects (body parts) that the motion concerns using some kinematical simulators. This method requires substantial work in event modelling in order to achieve realistic effects. The second method, which is also in vogue, regards the body as a whole non-atomic object consisting of several sub-components which attach to a skeleton system in *object modelling*. *Event modelling* moves the bone (as a parent or a child in the hierarchical skeleton tree) and 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. Inverse kinematics is a system in which the movement of the children is passed back up the chain to the parent. 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. Inverse kinematics models the flexibility and possible rotations of joints and limbs in 3D creatures. This approach gives the best possibilities to produce life-like animation of a character in a story, but does involve effort to set up the *object models* for the characters.

3.4.2. Extending predicate-argument representation to word level

In this section, we propose an action-decomposition structure for presenting visual semantics of action verbs, in which composite actions are recursively replaced with a set of more specific, partially-ordered sub-actions. We extend the predicate-argument representation of verbs in a bottom-up fashion by first examining primitives and then showing how they can be composed to define composite visual semantics (events).

The extended predicate-argument model describes verbs and verb phrases. To define objects, their properties and categories is the task of the object model (in visual knowledge). It may declare an object as an instantiation of a type (its category membership) at first, and give its attributes. The common attributes are position, orientation, size etc. Some complex objects may have other attributes like gender and age etc. for an instance of person. For example, Alice (a six years old girl) is an instance of person, with common attributes like her initial location, orientation (face to which direction), height (size) and specific attributes such as female (gender), 6 (age) etc. as well. The body parts are defined in the prototype of person.

Most of previous semantic representations discussed in section 2.1 suit for certain purposes. FOPC is good for query-answering (esp. a true or false judgement); event-logic truth conditions are suitable for motion recognition; x-schemas with f-structs suit for both verb recognition and performing the action but require training. The extended predicate-argument model discussed here is aimed at automatic generation of animation from linguistic input (language visualisation).

It can also be used to expand FOPC representations to make them workable on lower levels of linguistic input.

Constants, variables, types and their naming schemes

There are a few constants in this framework referring to specific objects which exist in every scene of the virtual world. We use the convention that names of constants in CONFUCIUS are composed of capitals and underscores.

GROUND is a plane in the coordinates (0,0,0) with the length and width which are greater than the space of the visualized scene and has the function of supporting things which otherwise will look like floating in the air. SUN is the default ambient light which illuminates objects in the scene. The constants in CONFUCIUS' knowledge base have different attributes as compared with the term as used in programming languages. Their values can be changed. The name *constant* only refers to their constant existence in the simulated visual world. For example, though it has infinite value for its length and width, one can change size of the GROUND according to the size of the stage for facilitating implementation. Similarly, for the SUN, when the language input describes that "It turned dark." or "in dusk", the brightness of the SUN can be changed.

Unlike in FOPC, some proper nouns such as `Mary` (person's name) are not treated as constants in the framework but just variables (instances of the type `man/woman`). Like in other programming languages, variables in CONFUCIUS can denote names of objects, which is an instance of a type. A variable name is started from a lowercase letter and can be followed by letters (uppercase or lowercase), numbers, underscores and hyphens. Object parts and properties can be referred to by a dot operator, e.g. `alice.righthand` or `alice.height`.

Type is the name of a category. We use the convention that a type name begins with a capital and is followed by letters, numbers, underscores or hyphens. As in WordNet (Beckwith et al. 1991), objects inherit all the properties of their super-concepts (parents). However, attributes such as size, color, position, etc. can be specialised. There are two operations involving type: `type(objName, typeName)` and `aKindOf(subtypeName, parentTypeName)`, e.g. `type(alice, Girl)`, `aKindOf(Girl, Person)`.

Hierarchical structure of predicate-argument primitives

The predicate-argument format we apply to represent verb semantics has a Prolog-inspired nomenclature. Each non-atomic action is defined by one or more subgoals, and the name of every goal/subgoal reveals its purpose and effect. Primitives 1 through 10 below are basic primitive actions in our framework (see Figure 3.8). We do not claim that these ten cover all the necessary primitives needed in modelling observable verbs. 11 and 12 are actually not primitive actions, but they are necessary in processing complex space displacement. In the first ten primitives, 1-3 describe position movement, 4 and 5 concern orientation changes, 6-9 focus on alignment, and 10 is a composite action (not atomic) composed by lower level primitives.

- 1) `move(obj, xInc, yInc, zInc)`
- 2) `moveTo(obj, loc)`
- 3) `moveToward(obj, loc, displacement)`
- 4) `rotate(obj, xAngle, yAngle, zAngle)`
- 5) `faceTo(obj1, obj2)`
- 6) `alignMiddle(obj1, obj2, axis)`
- 7) `alignMax(obj1, obj2, axis)`

- 8) alignMin(obj1, obj2, axis)
- 9) alignTouch(obj1, obj2, axis)
- 10) touch(obj1, obj2, axis) ; for the relation of support and contact
- 11) group(x, [y|_], newObj)¹¹
- 12) ungroup(xylist, x, yList)¹²

Figure 3.8: Primitive actions within CONFUCIUS

Figure 3.9 illustrates the hierarchical structure of the ten primitives. Higher level actions are defined by lower level ones. For instance, alignment operations are composed by `move()` and/or `moveTo()` actions. We will explain these primitives below.

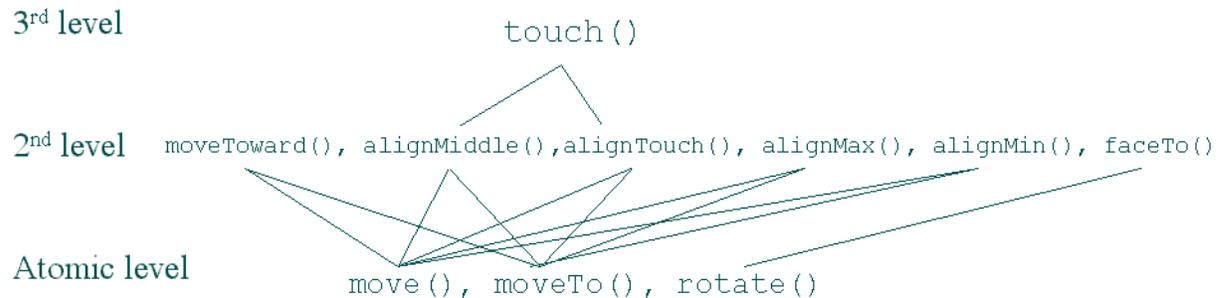


Figure 3.9: Hierarchical structure of primitives

`move(obj, xInc, yInc, zInc)` moves `obj` by designated displacement on the specific axis (axes). Arguments `xInc`, `yInc`, and `zInc` can be a place-holding space (or zero) indicating no displacement on the axis. For example, `move(glasses, _, 10, _)` means ‘move the glasses up 10 units’.

The following primitive action `moveTo()` moves `obj` to a specific position designated by `loc` which is an instance of type `Position`, consisting of a 3D coordinates.

```
moveTo(obj, loc):-
    type(loc, Position).
```

The following primitive action `moveToward()` moves `obj` towards/away from a designated position by some displacement. The second argument is the 3D coordinates of the destination. The third argument can be positive, means *move toward the destination*; or negative, means *move away from the destination*. If the third argument is not given, when called as `moveToward(obj, loc, _)`, the displacement will be a positive random value greater than 0 and less than the distance between `obj` and destination location. This is a second level action, implemented by movement primitive action at the first level (see Figure 3.9).

```
moveToward(obj, loc, displacement):-
    type(loc, Position).
```

¹¹ As is the convention in the programming language Prolog, arguments can be replaced by an underscore if they are undetermined.

¹² `ungroup` element `x` from a list which contains it. `yList` is the rest of the list after deleting `x` from the original list. This is also a basic list operation in Prolog.

`rotate(obj, xAngle, yAngle, zAngle)` rotates `obj` on the designated internal axis (axes). The last three arguments are not external absolute coordinate axes, but the internal coordinate axes of the `obj`, which is defined in visual knowledge->internal coordinate axis (Figure 3.2). It will be shown with CONFUCIUS that using internal axes is more practical than external axes.

`faceTo(obj1, obj2)` is a second level action, also involving an object's internal coordinate axes. It faces `obj1` to `obj2`. This operation concerns not only `obj1`'s internal axes, but also its functional information (Figure 3.2) in which its face is defined. For example, `faceTo(book, alice)`, the book is a cube whose face is defined as book cover. This action rotates this book on its internal axes to make its cover face to `alice`.

Figure 3.10 illustrates alignment actions from 6 through 9. Note that `alignTouch(obj1, obj2, axis)` uses the first object's maximum value and the second object's minimum value along the axis.

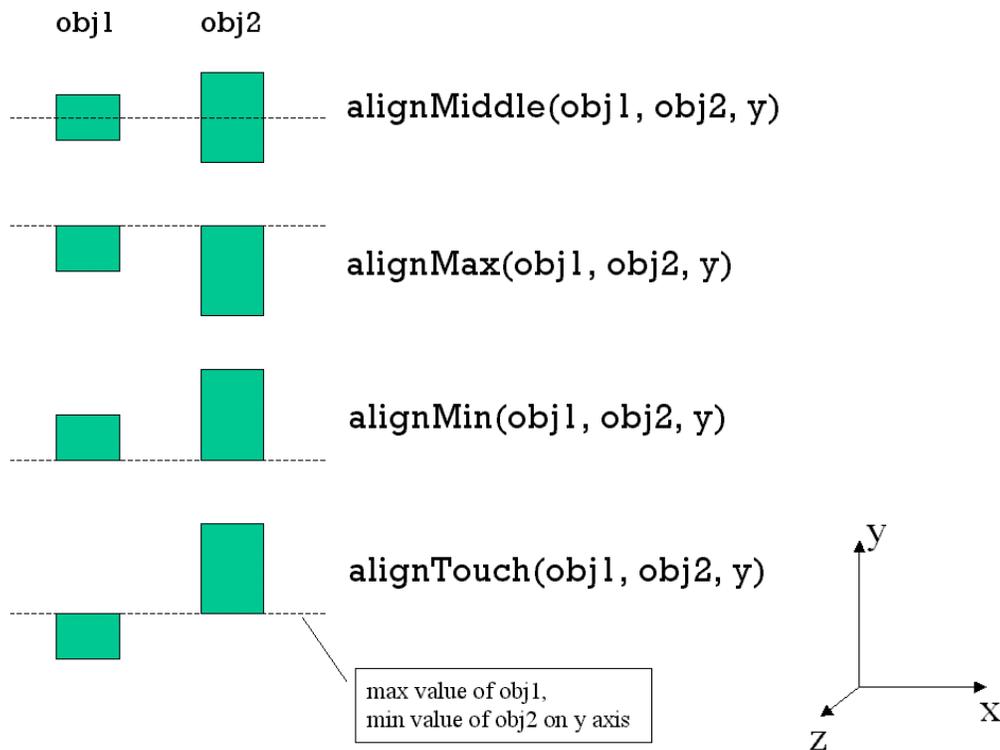


Figure 3.10: Align primitives (on y axis) in a 3D coordinate system

`touch(obj1, obj2, axis)` is a composite action on the third level in the hierarchy. It moves the first argument to the destination where it can touch the nearest face of `obj2`. Figure 3.11 shows what and how `touch()` works given different axes as the third argument.

Examples of verb definitions in extended predicate-argument model

Given below are some examples of visual semantics of verb phrases using the above primitives.

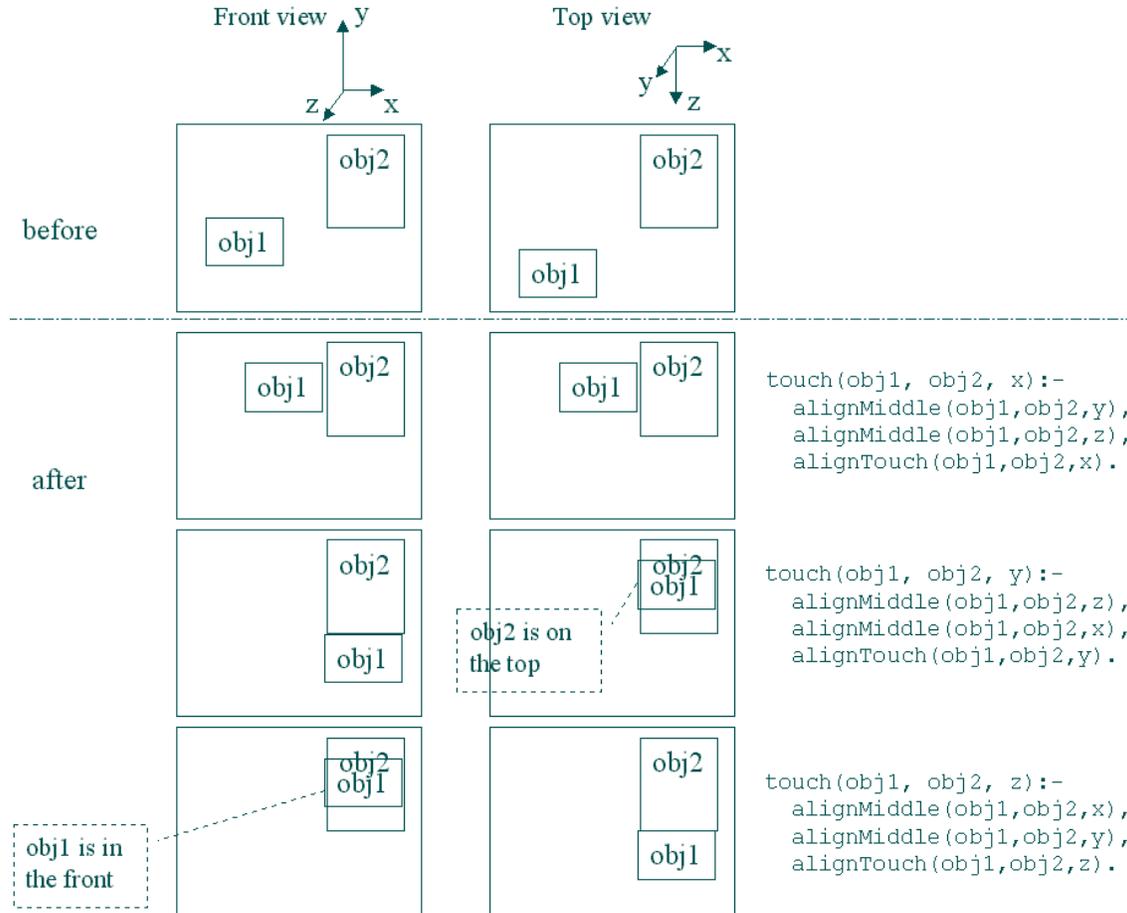


Figure 3.11: touch() along different axes

Example 1, *jump*:

```
jump(x):-
  type(x, Animal),13
  move(x.feet,_,HEIGHT,_),
  move(x.body,_,HEIGHT,_),
  move(x.feet,_,-HEIGHT,_).
```

Other complex joint movements will be modelled by inverse kinematics. If the second subgoal about movement of the man's body is missing, we will see the man's feet lift with some corresponding movements on his legs (by inverse kinematics), but his body keeps at the original height. The reverse subgoal, `move(man,_, -HEIGHT,_)` is not necessary because it will be done by inverse kinematics automatically.

Example 2, *call*:

As in "a is calling b" (verb tense is not considered here because it is at sentence level rather than word level). This is one word-sense of *call* where calling is conducted by telephone. Here is the

¹³ Semantic constraint. Declare an instance of the type 'Animal'. Metaphor use of vegetal or inanimate characters is not considered.

definition of one word-sense of *call* which is at the first level of the visual semantic verb representation hierarchy.

```
call(a):-
    type(a, Person),
    type(tel, Telephone),
    pickup(a, tel.receiver, a.leftEar),
    dial(a, tel.keypad),
    speak(a, tel.receiver),
    putdown(a, tel.receiver, tel.set).
```

The second argument is place-held by an understore because we don't care who B is since he is not present in the scene. The `type(a, Person)` operation is for semantic constraint, to declare that `a` is an instance of the type 'Person'. The variable `tel`, an instance of `Telephone`, is from association between events *call* and the object *telephone* in the knowledge base (see Figure 3.2). `speak()` involves lip movement and coordinates with other modality-speech. `putdown()` refers to the movement of *hangup*.

Here is the definition of *pickup* which is at the second level of the visual semantic verb representation hierarchy:

```
pickup(x, obj, dest):-
    type(x, Person),
    moveToward(x.leftHand, location(obj), location(obj)-location(x)-5),
    touch(x.leftHand, obj, y),
    group(x.leftHand, obj, xhandObj),
    moveToward(xhandObj, dest, _).
```

The three arguments of `pickup()` are subject, object and destination position respectively. `dest` is the 3D coordinates of a location. Here `obj` is a telephone receiver.

Here is the definition of *putdown* which is also at the second level of the visual semantic verb representation hierarchy:

```
putdown(x, obj, dest):-
    moveTo(x.leftHand, dest),
    ungroup(x, obj, x1),
    type(x1, Person).
```

The variable `x` is an aggregation object containing a person and object(s) which are not part of the human body. Here `obj` is a telephone receiver.

There are many complex issues left unconsidered in the example, such as how to put down the receiver in the exact place of the phone set etc. However, this can be achieved by some further operations like as combination of aligns and moves with the calculation of the location of a part of an object.

The above examples show a hierarchical representation that involves multiple levels of visual description and the ability to perform top-down interpretation when necessary right down to the primitive level. The implementation of the visual semantic representation provides a backtracking mechanism similar to Prolog which is convenient and efficient. Constructing a complete event semantics of the visual knowledge base (event model in Figure 3.2) by the above method requires extensive work on verb definitions according to their corresponding semantic knowledge in the

lexicon (Figure 3.2). However, such work is indispensable for an automatic language visualisation system.

3.4.3. Representing active and passive voice

One important difference between active and passive voice is semantic: the subject of an active sentence is often the semantic agent of the event described by the verb (*He* received the letter) while the subject of the passive is often the undergoer or patient of the event (*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 are same for the two sentences, the difference is the parameters (orientation and position) of Viewpoint node to represent the topic in each voice.

3.4.4. Representing tense and aspect

The issue of representing temporal information of events is addressed in this subsection. It is very important for the information that verb tenses and aspect convey. 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.

Verb tense of the language input in CONFUCIUS is interpreted by the temporal reasoning component during natural language processing, and is translated to sequential events. This will be further discussed in section 5.1.

3.5. Visual semantic representation of adjectives—meaning as attribute

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 in CONFUCIUS are discussed.

3.5.1. 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 organized 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 3.12 we classify the category of adjectives according to the perceiving senses they require. The first level is distinguished by the standard 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 could 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 includes human feelings, 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*, etc. 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 CONFUCIUS treats this subcategory of adjectives as noun (section 3.3), and represents the appropriate nouns that they point to in WordNet.

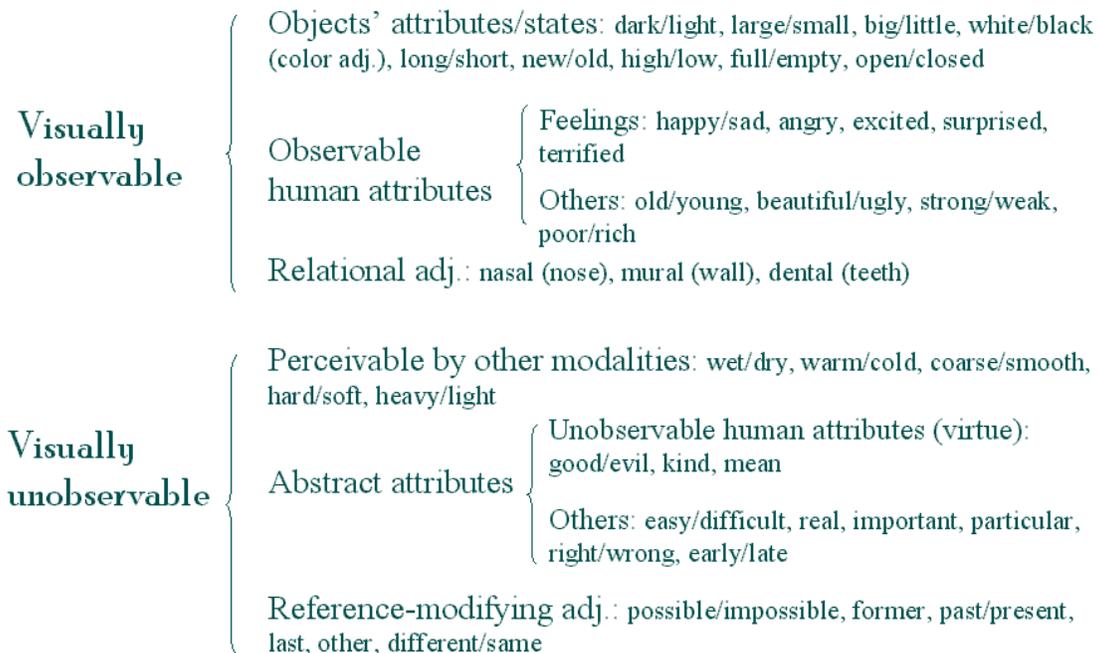


Figure 3.12: Categories of adjectives

There are three types in the unobservable class: the first type is adjectives that can be perceived by other modalities such as haptics, though not always by vision (e.g. wet/dry, warm/cold, coarse/smooth, hard/soft, heavy/light); the second 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* etc. Many refer to the temporal status of the noun (*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*; *the alleged killer* states that *she allegedly killed*.

CONFUCIUS represents 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 is just transmitted to a text-to-speech engine directly and is presented in speech modality; if it appears in the narration part, the natural language processing component judges whether it is presentable visually, and if not, the whole sentence is sent to a text-to-speech engine and is output in the narrator’s speech with the animation presentation (without the information of the unobservable adjective).

3.5.2. 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: 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 organization 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*. The simple answer is to introduce a similarity relation and use it to indicate that the adjectives lacking antonyms are similar in meaning to adjectives that do have antonyms. 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 (see Figure 3.13). Thus, *soggy* is similar to *wet* and *wet* is the antonym of *dry*, so a conceptual opposition of *soggy/dry* is mediated by *wet*.

This strategy in lexical semantics will greatly reduce graphic representation of synonyms. For the example in Figure 3.13, 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 3.2 illustrates lexicalized gradations for SIZE, WHITENESS, AGE, VIRTUE, VALUE, WARMTH and ANGER. In Table 3.2 bolded words are adjectives with bipolar values which are antonyms in the given column.

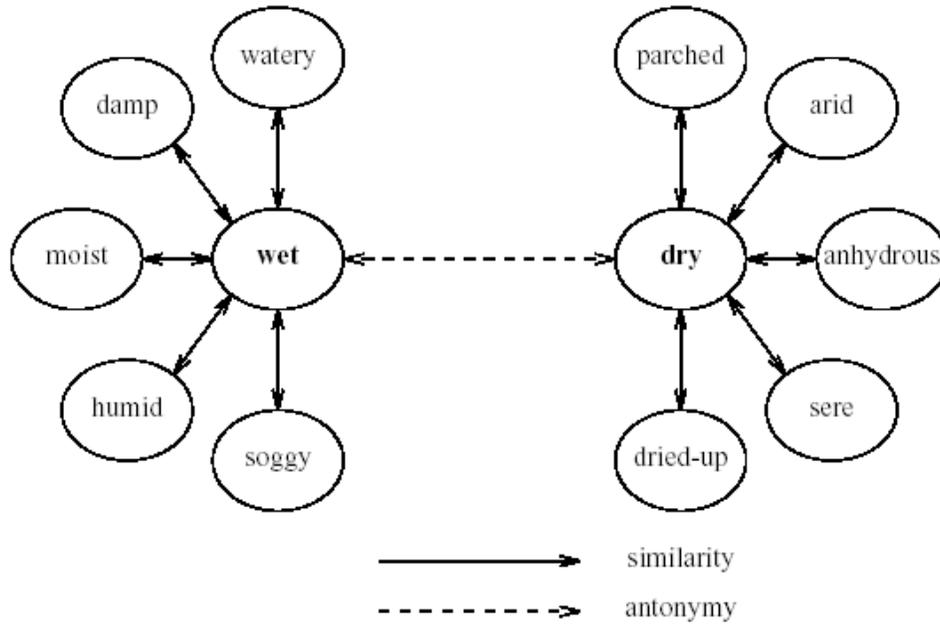


Figure 3.13: Bipolar adjective structure in WordNet

SIZE	WHITENESS	AGE	VIRTUE	VALUE	WARMTH	ANGER
astronomical	snowy	ancient	saintly	superb	torrid	furious
huge	white	old	good	great	hot	wrathful
large	ash-gray	middle-aged	worthy	good	warm	enraged
standard	gray	mature	ordinary	mediocre	tepid	angry
small	charcoal	adolescent	unworthy	bad	cool	irate
tiny	black	young	evil	awful	cold	incensed
infinitesimal	pitch-black	infantile	fiendish	atrocious	frigid	annoyed

Table 3.2: Examples of graded adjectives

Generally, graded adjectives represent a range of points along a linear state machine. This feature gives an automatic visualisation system more control on representing these attributes. Attributes like SIZE, LENGTH and WHITENESS are easy to present comparing with other attributes such as *observable human attributes*. Figure 3.14 shows the representation of gradable observable emotional attributes such as HAPPINESS, SADNESS, and ANGER on Baldi's (CSLU 2001) face. Baldi's expressions could be adjusted by scrollbars. The first face is a neutral face (default values: neutral=1.00, all the others are 0.00), the third is a happy face with happiness=1.00, and the second one is a happy medium with happiness=0.50.

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-aged'. Justeson and Katz (1993) note that the noun context therefore often serves to disambiguate polysemous adjectives.

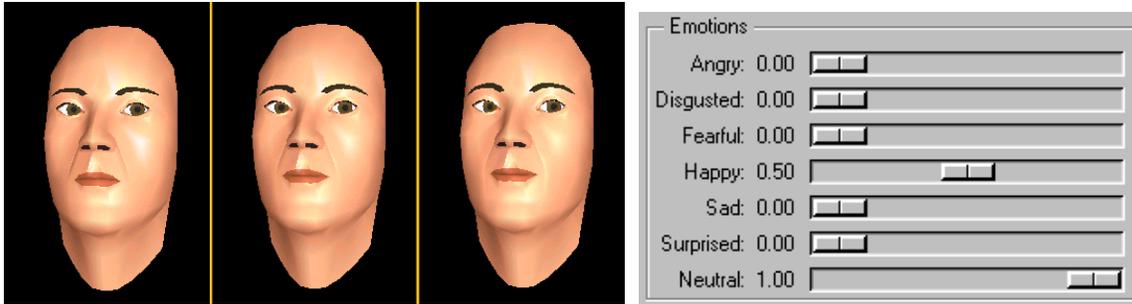


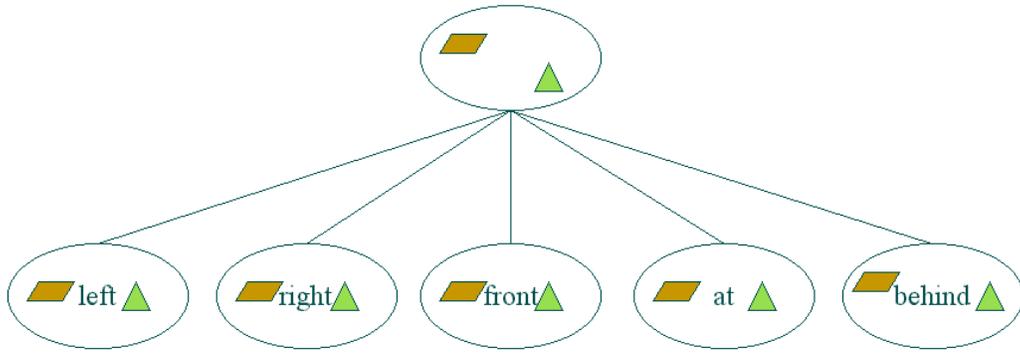
Figure 3.14: Adjustable facial expression of Baldi

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 simply modifies those values above or below their default values.

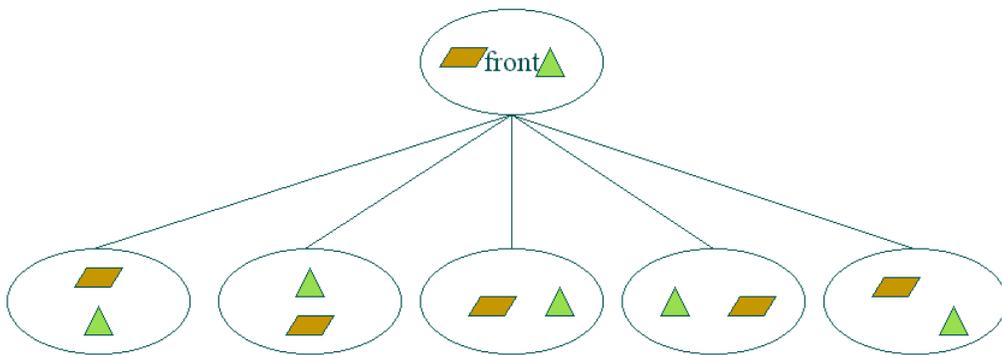
Graded adjectives can also provide subjective descriptions of an object depending on the speaker's individual background and experience. So a person from a remote town may describe a two-storey building as *tall*, whereas another person from a large city would describe it *short*.

3.6. Visual semantic representation of spatial prepositions

A significant portion of visual semantics involves the interpretation of spatial prepositions. Prepositions often bear spatial concepts that are crucial to decide entities' location and movement paths of events in a language visualisation system. The location of an object is only relative to other objects' positions 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 in common sense), and the absolute coordinates of objects are irrelevant. The trick of visualizing 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 configuration, and one geometric configuration could be described by different prepositions. In Figure 3.15a we show different prepositions based on the direction of observation of the two objects, and in Figure 3.15b we show different geometric configurations based on the direction of observation of the two objects with the preposition *front*. Figure 3.15a is relevant for recognition processes of vision systems like VITRA (Schirra 1993), and Figure 3.15b is relevant for visualisation processes of intelligent presentation systems like CONFUCIUS. 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 visualize *in front of the blackboard* properly, the functional information of *blackboard* and surrounding information *classroom/meeting room* and the *audience* need to be considered.



(A) One geometric configuration to many prepositions



(B) One preposition to many geometric configurations

Figure 3.15: The relation between spatial prepositions and geometric configurations

4. Project proposal

In this chapter an intelligent multimedia storytelling system—CONFUCIUS—for translating natural language story/script input to animation is proposed. The core of the development of CONFUCIUS would be the construction of tools to translate natural language input into its dynamic graphic presentation. It integrates natural language processing, concept representation and reasoning, animation generation and other components. There are three main areas of contribution: (1) multimodal semantic representation of natural language, (2) multimedia fusion and coordination, and (3) real-time language visualisation. CONFUCIUS uses a new representation method of visual semantics and new multimedia coordination methods. These will be implemented using VRML and Java. This work will also make advances on automatic language visualisation through the development of CONFUCIUS.

4.1. Architecture of CONFUCIUS

The architecture of CONFUCIUS is depicted in Figure 4.1. The dashed part in the figure includes the prefabricated objects such as characters, props, and animation for primitive actions, which will be used in the *animation generation* module. When the input is a story, it will be transferred to a script by the *script writer*, then parsed by the *script parser* and the *natural language processing* module sequentially. The three components of *Natural Language Processing* (NLP), *Text to Speech* (TTS) and *sound effects* operate in parallel. Their outputs will converge and combine at the *code combination*, which generates a holistic 3D world representation including animation, speech and sound effects.

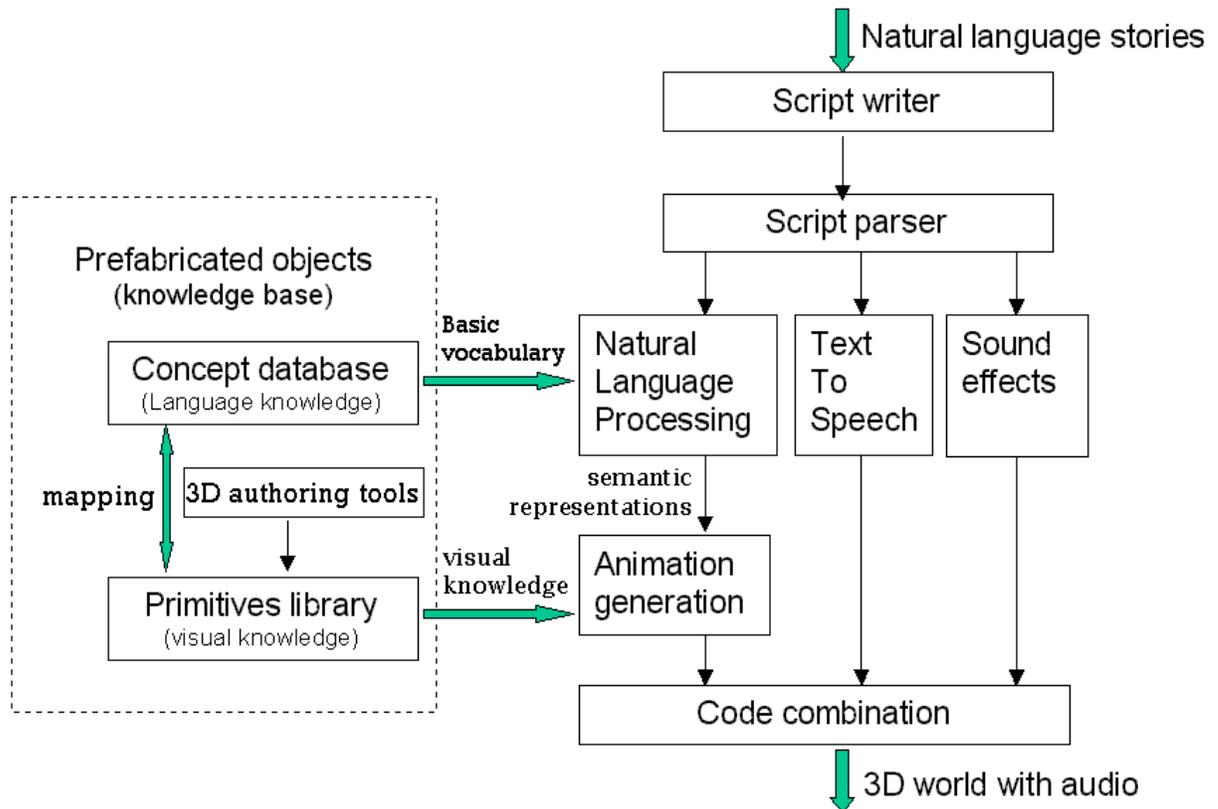


Figure 4.1: System architecture of CONFUCIUS

4.2. Input stories/scripts

The input of CONFUCIUS could be taken from children's stories like "Alice in Wonderland", and the proposed system should represent the semantics in a way that can generate common sense inferences about the stories, present them by animated characters and a human narrator or scenario animation with incidental music matching the plot development, according to user's choice.

The input story can be classified to levels of abstraction. The reason for choosing children's stories as input is because the content in these stories is concrete and tangible, which circumvents some presentation difficulties caused by abstract expressions, e.g. it may be possible to represent the meaning of 'lazy' but has trouble to present more abstract adjective like 'eccentric'. However, a sophisticated storytelling system should be applied at any level of abstraction. On a longer-time scale, it is planned to apply CONFUCIUS to other story domains of increasing complexity and abstraction that require metaphor understanding, such as news and fables etc.

Play scripts are another input for CONFUCIUS. Compared with stories, they are easier to parse because they are partially structured, that is, they have distinct parts for scene description, a set of characters, and dialogue (or monologue). The space created in--and by--a play script is smaller than that of a novel which tells the same story. In addition, scripts often specify technical demands/requirements like sound, lights, costume and sets. This makes the design task of CONFUCIUS much easier for scripts than for stories (or novels).

We plan to use Samuel Beckett's plays, such as "Endgame", "Quad", "What? Where?", "Come and Go" (Beckett 1984), as the script input of CONFUCIUS. The reason we choose Beckett has nothing to do the *existential philosophy* and the *Theatre of the Absurd* in his plays. We choose his play scripts just for technical consideration of directing and producing. Just like he explained why he turned to theatre, "When I was working on *Watt*, I felt the need to create for a smaller space, one in which I had some control of where people stood or moved, above all, of a certain light. I wrote *Waiting for Godot*." (Worton 1994), Beckett's theatrical space is much smaller than other playwrights. His plays typically has a nearly bare stage and few characters with simple movements (has symbolic meaning), and hence give the audience more interpretive challenges than is usually admitted. Statues, reflections, sleepwalkers and silence are teemed with most of his plays. Though what Beckett says outside the texts of his plays is undoubtedly worth considering, it is not the job of CONFUCIUS. These features of Beckett's plays make them well-suited for automatic presentation applications.

4.3. Data flow of CONFUCIUS

The data flow diagram (DFD) of CONFUCIUS is shown in Figure 4.2 which illustrates the data exchange among the main components.

The aim of the *script writer* is to transfer a usual English story to a drama-like script which meets the system's requirements, and send the script to *script parser*. If the input is a script, it goes to the *script parser* directly. The *script parser* then parses the script into several functional parts: actor description, scene description, dialogues, actions, and non-speech audio description which are passed to corresponding processors respectively, e.g. passing dialogues to TTS. The main part, *scene and actor descriptions* and their actions are sent to a *natural language parser*. The function of the *natural language parser* is to parse, mark the input text, and identify characters, objects, and events in the description. After separated processing, VRML codes from the

animation director, *TTS* and *sound effect driver* converge at *media coordination* where synchronisation is performed to match speech and music to vision, as shown in Figure 4.2.

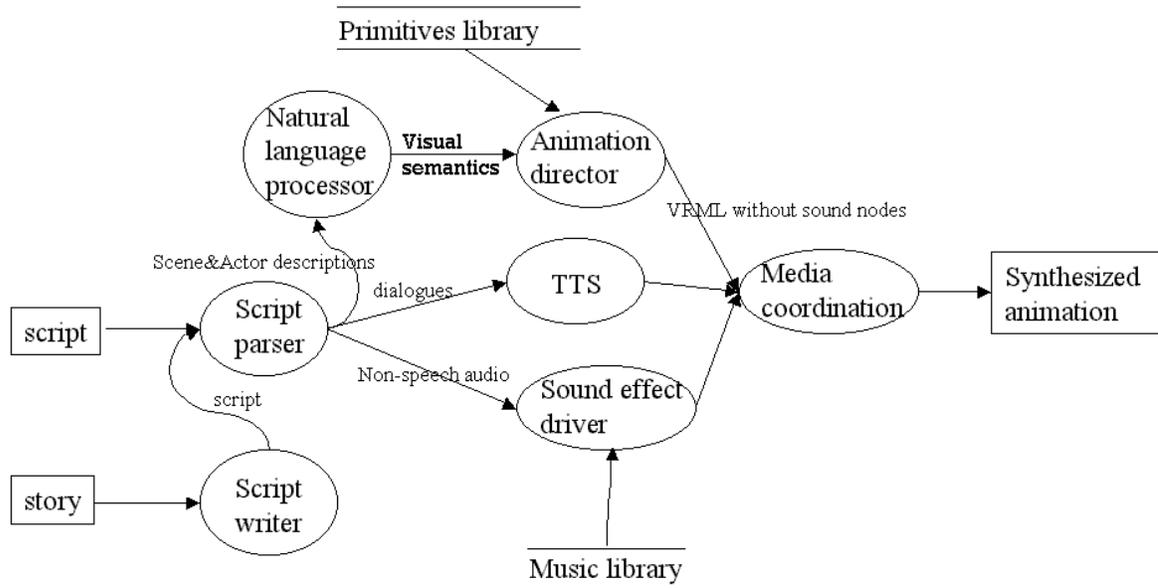


Figure 4.2: Level 1 Data Flow Diagram of CONFUCIUS

The *animation director* that accepts marked scripts and generates silent animations. The DFD of this component (level 2) is illustrated in Figure 4.3. The coordinator component is in charge of the communication among actor manager, scene generator and props provider, such as spatial relationship information.

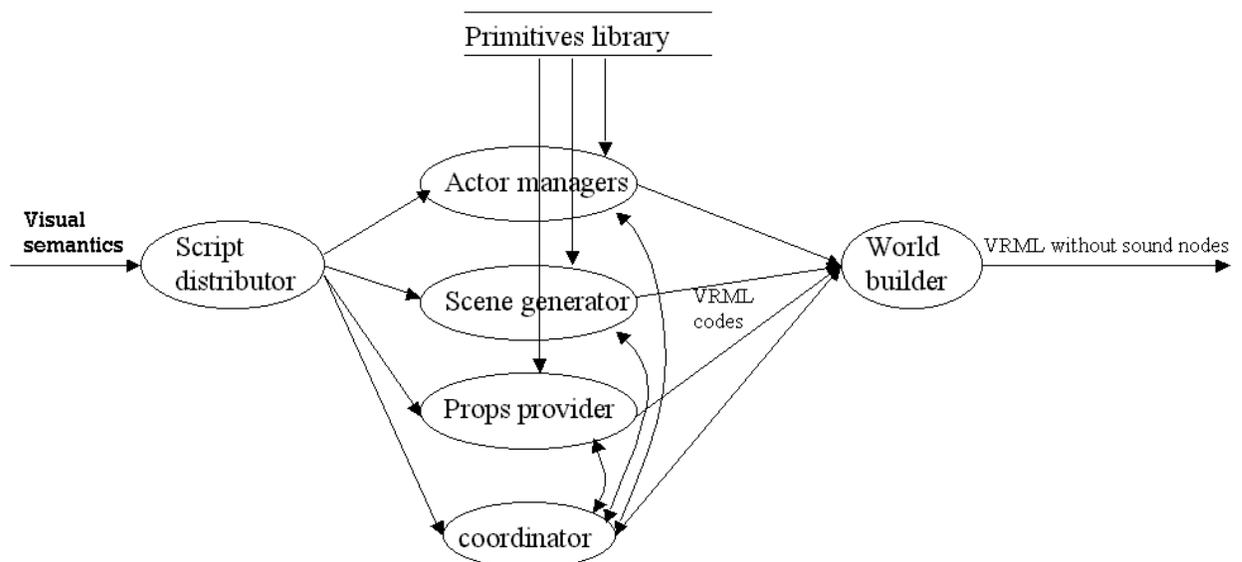


Figure 4.3: Level 2 Data flow diagram of *animation director*

4.4. Comparison with previous work

The following Table 4.1 presents a comparison showing features of various intelligent multimedia systems. Besides the systems listed in the table, many other practical applications of intelligent multimedia interfaces have been developed in domains such as intelligent tutoring, retrieving information from a large database, car-driver interfaces, real estate presentation and car exhibition etc.

Most of the current multimedia systems mix text, static graphics (including map, charts and figures) and speech (some with additional non-speech audio) modalities. Static graphical displays in these systems constrain presentation of dynamic information such as actions and events. To make references on a static graph, some use highlighting and temporally varying effects like moving, flashing, hopping, zooming, scaling (Zhou and Feiner 1998), which are frequently used in PowerPoint “animation” effects presentation. The graphics of most systems which present animation is two dimensional, such as KidsRoom, Gandalf and PPP persona. Only SI (SONAS), Larsen and Petersen’s interactive storytelling environment, Cassell’s agents, and Narayanan’s primitive-based language animation have three-dimensional image quality. However, SI/SONAS needs user’s intervention to navigate and modify the 3D world and hence its performance is rather like a 3D browser which responses to speech commands. The application domain of SI/SONAS is limited to scene description and spatial relationships. Cassell’s SAM and REA achieve great improvement in simulating humanoid behavior in conversation, but they are limited to human-computer interface applications. Although Larsen and Petersen’s interactive storytelling environment does present 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. Albeit Narayanan’s language animation may generate 3D animation automatically, the quality of its images (iconic feature) is inadequate.

Hence, a core contribution of CONFUCIUS is to generate dynamic 3D graphics automatically from natural language stories and script-based input. CONFUCIUS will provide a new methodology of multimodal semantic representation and multimodal fusion and coordination. Current state of the art techniques in natural language processing and speech synthesis, automatic graphic and animation design, 3D graphic design, media design and coordination will be utilized and incorporated in CONFUCIUS.

4.5. Animation generation

Automatic generation of animation incorporates design expertise and automated selection, creation and combination of graphical elements. The animation generation of CONFUCIUS 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 4.4, the animation producer generates VRML codes of *acts* (or parts of a story). It consists of *actor manager, world builder* and *graphic library*. The world builder simulates the world model, i.e. it sets up the stage: sets and props (including lights and sound) from scene description of the script input. 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 performers, including their costumes and makeup. It manages speech and motion of synthetic characters. The graphic library contains reusable graphic components of the knowledge base (e.g. actors, props, and tiles). It is possible to reuse sets, props and performers in other stories 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 (e.g. an actor walking

Categories	SYSTEMS	NLP Component		Multimodal interaction									
		Natural language generation	Natural language understanding	Input Media				Output Media					
				Typed-in text	Pointing ¹	Speech recognition	Vision recognition	Text	Audio		visual		
									Text to speech	Non-speech audio	Static graphics	Animation	
									2D	3D			
Intelligent multimedia authoring systems	WIP	v	v	v				v			v		
	COMET	v	v	v				v			v		
	TEXPLAN			v	v			v	v			v	
	CUBRICON	v	v	v	v	v			v		v		
Automatic Text-to-Graphics	WordsEye		v	v							v		
	Primitive-based language animation ²		v	v									v iconic
	SI and SONAS		v	v			v						v
Multimodal Storytelling	Schank's SAM and PAM	v	v	v				v					
	Interactive Storytelling ³						v	v	v				v
	AesopWorld	v	v	v				v	v	v		v	
	KidsRoom						v	v	v			v	
	OZ	v	v	v				v				v	
Intelligent multimedia agents	Cassell's SAM and REA (BEAT)	v	v	v			v	v					v
	Gandalf		v				v	v				v	
	PPP persona			v	v			v	v		v	v	
Intelligent multimedia interfaces	AIMI			v	v	v		v	v	v	v		
	AlFresco	v	v		v	v		v	v		v		
	XTRA		v	v	v			v			v		
This Project	CONFUCIUS		v	v	v ⁴				v	v			v

1 Pointing on menus, pictures, maps, charts, tables, or spreadsheets through mouse or touchscreen

2 Narayanan's language visualisation based on Schank's primitives

3 Larsen and Petersen's interactive storytelling in a multimodal environment

4 tailored menu for script input

Table 4.1: Comparison of intelligent multimedia systems

towards a house), the *world builder* exchanges information like spatial relationships with the *actor manager*.

The stories will be presented using 3D graphics. A 3D graphical approach is used in presenting the stories instead of 2D graphics because this is the state of the art of the technology in games and recent multimedia applications.

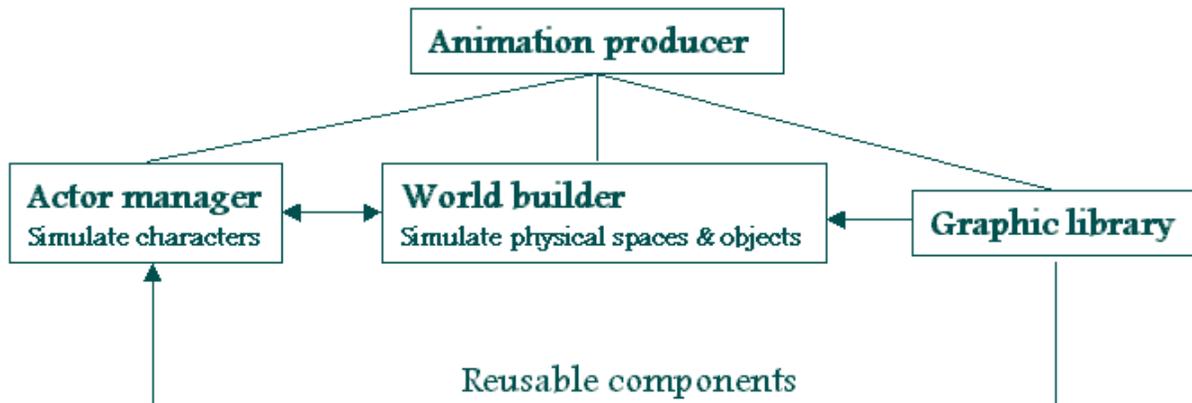


Figure 4.4: Structure Diagram of *animation producer*

4.5.1. Animated narrator

The animated narrator in CONFUCIUS is an intelligent autonomous agent who tells stories beside the animation. The requirements of a narrator are less than characters in multimodal storytelling systems, and therefore the number of body movement types that a narrator is capable of carrying out is much less than that of characters. In Larsen and Petersen's (1999) interactive stories 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 the actors in CONFUCIUS's 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 script or request for user inputs. For instance, in Larsen and Petersen's stories when an actor (not the user controlled avatar) "sees" (perceives) another actor or a prop, he might have some reaction such as surprising expression or a greeting by sound modality, which is decided by his behavior model. This mechanism makes a story more dynamic and characters can stand on their own. In CONFUCIUS, the events of encountering (the actor's perception) and his corresponding response are provided by the input story text.

In addition, the narrator's voiceover has another significant role besides introducing the storyline. It may cover the information that cannot be presented by other modalities successfully. 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 this task is rather a challenge. When CONFUCIUS' animation generation model detects that it is hard to express the information about *marry* via physical movements of characters in animation, the narrator's voiceover will do the job by just speaking this sentence with the accompanying animation showing the couple is together.

4.5.2. Synthetic actors

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

Motion animation

There are some basic requirements that are needed for synthetic 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 (circumventing objects and avoiding collision with other actors), *path planning* involves obstacle avoidance and collision detection.

Behavior models are a feasible way to create an autonomous actor in an interactive story generating process. The more complex behaviors which are possible to give the actors, the less scripting is needed in the original story script (story frame). Behavior 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. Since CONFUCIUS is a story presentation system, characters' behaviors rely on input story scripts rather than behavior models.

4.5.3. 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 incompleteness of natural language. *Common sense* (default attributes) is used to bridge the gap between them. 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 color. 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 4.5), the attributes of an object are decided by default values.

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 4.5: *Alice in Wonderland*: 'Down the Rabbit-hole'

Without the particular measurement in the story, CONFUCIUS would draw a door about 7 feet high and Alice as a 3 feet high little girl. So after interpretation the first paragraph in the example, a 3-foot-high (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.

Default values of an object's attributes are indispensable not only in object visualisation but in setting values to modifiers that modify the object (see section 3.5). Moreover, default values may also concern events such as speed, mode of movement etc. 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.

4.5.4. Layout

When we design the layout of a stage, i.e. the placement of props, and the movement of actors, besides considering their size and position (spatial relationship) one inevitable problem we may encounter is *collision detection*--a 'naïve' physical problem about how to detect collision of objects and actors. Although VRML provides a built-in collision detection mechanism for the avatar (user), the mechanism does not apply to other objects. However, a large proportion of collisions occur among objects and actors' (if the actor is not the first person) motions.

We found two ways to detect collision between objects. The first is to write up scripts detecting intersection between the bounding boxes of objects. It needs 3D translation calculation especially when the objects detected are moving (rotating). The other is to group a viewpoint with the geometry which we want to detect collisions and bind it temporarily, therefore we may utilize the automatic collision detection for the avatar (the active viewpoint) by using Collision node and setting `avatarSize` in `NavigationInfo` Node meanwhile.

4.6. Multimedia presentation planning

Enlightened by previous intelligent multimedia presentation systems such as WIP (Wahlster et al. 1992) and COMET (Feiner and McKeown 1991a, b) as we discussed in section 2.2.1, we formulate the principles for media allocation within CONFUCIUS as the following:

1. Realize spatial information, physical attributes, physical actions and events in animation.
2. Realize dialogues, abstract concepts (including abstract actions and abstract relations), and temporal information in language (dialogue, voiceover narrative, or caption). For example, if the media allocator detects an unobservable adjective that can not be presented visually in the narration part of a story, the whole sentence is sent to a text-to-speech engine and is output in the narrator's speech, while the other visually presentable parts of the sentence are still allocated to the animation generator to create animations which will be played when the narrator is talking.
3. Realize failed attempts and successful attempts with low confidence in principle 1 in language modality according to feedback from the animation generator.
4. Realize mood, frequency, velocity etc. in non-speech audio (if music switch is on) and/or other modalities. Since auditory information can be redundant with other modalities (e.g. visual and language), when a *non-speech audio medium realization* succeeds we usually make the audio information redundant instead of eliminating the visual (or speech) presentation, to secure the information is transmitted without loss.

Figure 4.6 shows the block diagram of CONFUCIUS from the perspective of multimedia presentation planning. Media allocation also receives feedback from media realization to influence the selection of media for a definite content. Thus we allow a decision made or failed realization at a later stage of processing can propagate back to undo an earlier decision. For example, realization in animation generator may fail because of visualisation difficulties, and this message should be fed back to *media allocation*, where the content could be re-allocated to other media.

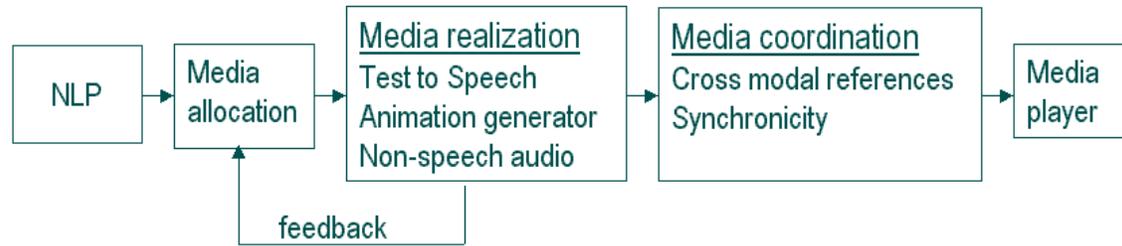


Figure 4.6: Block diagram of CONFUCIUS

4.7. Issues raised

Consideration of this intelligent storytelling system raises a number of issues which are addressed in the following sections.

4.7.1. Size of CONFUCIUS' knowledge base

The problem of knowledge base size answers the question how much *common sense* knowledge shall be incorporated in a general story animation generating system. To generate understandable story animation, some human common sense including social conventions and other aspects of the culture and world in which the story occurs, e.g., Schank's *scripts*, and default attributes of objects must be incorporated into CONFUCIUS. Without proper knowledge of a specific domain that a story/script¹⁴ pertains to, the system could not tell the story intelligently. Schank's SAM (c.f. section 2.6.1) 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.

We showed the design of CONFUCIUS' knowledge base in section 3.1. It consists of language models, visual models, world knowledge and spatial and qualitative reasoning modules. Language models are lexical semantics used in natural language processing component to extract concept semantics from text. Visual models consist of the information required to generate moving image sequences, comprising a graphic library. Schank's *scripts* are a kind of world knowledge.

To implement a domain-independent intelligent storytelling system, a gigantic knowledge base consisting of the above four parts is required. In order to make the project more practical, we are not going to create a complete knowledge base for all kinds of stories. We just show how it works on some example children's stories and example Beckett play scripts. However, CONFUCIUS should be reusable for other story categories, provided the corresponding knowledge is added. Therefore, to present a new story, it would be necessary to augment the lexicon, the world knowledge, and the graphics knowledge base (adding the new objects' detailed geometry and physical properties). These are currently substantial tasks, esp. expanding the graphics library, but so is the effort required to create conventionally authored storytelling presentations (e.g. film and cartoon) for a new story. Also, note that the graphics knowledge base should be available in VRML format¹⁵.

¹⁴ 'Script' in normal format means movie script, drama, or opera script as an input, when in italics, it means Schank's (1977) data structure of representing world-knowledge of stereotypical situations.

¹⁵ Other common formats such as 3ds could be converted to VRML through most 3D graphic authoring/conversion tools.

4.7.2. Modelling dynamic events

Modelling dynamic events is another major issue in language visualisation. If one is to describe action in the world, or moving sequences of images, it is necessary to represent dynamic events such as running, jumping, and giving. This requires the matching of low-level dynamic visual percepts with intermediate concepts in event frames (c.f. Figure 2.21).

According to our events categorization in section 3.4, VRML's linear interpolators (ColorInterpolator, PositionInterpolator, OrientationInterpolator, ScalarInterpolator etc.) for keyframed animation is adequate for *atomic events*. They can be used to change values of objects' attributes like color, position, and size. For human action verbs, the movement decomposition method proposed in section 3.4 can decompose an action into primitive movements in sequential order and hence present it in animation.

4.7.3. Ungrammatical sentences in natural language input

A Context-Free Grammar (CFG) defines a formal language (a set of strings). *Grammatical sentences* that can be derived by a grammar are in the formal language defined by that grammar. However, there are many *ungrammatical sentences* in real natural language that cannot be derived by a given formal grammar, and they are popular in scene descriptions of play/film scripts, e.g. the beginning of S. Beckett's *Come and Go* in Figure 4.7.

Sitting centre side by side stage right to left FLO, VI and RU. Very erect,
facing front, hands clasped in laps.
Silence.

Figure 4.7: The beginning of Samuel Beckett's short play *Come and Go*

However, ungrammatical sentences do not cause any difficulty in human understanding because understanding a given sentence often depends on the context. Winograd (1972)'s model made it clear that the problem of parsing was well-enough understood to begin to focus on the more difficult problems of semantics and discourse models. This is the downside of CFG and generative grammar. Since language does not normally consist of isolated, unrelated sentences, but instead of collocated, related groups of sentences — *discourse*, it is not adequate to interpret natural language only in isolated formal ways. Natural language stories are in fact a discourse of a particular sort — a *monologue*. Monologues are characterized by a *speaker* (include writers) and a *hearer* (includes readers). The communication flows in only one direction in a monologue, that is, from the speaker to the hearer. CONFUCIUS' story understanding is a monologue process: the story writer (or playwright) could be regarded as the *speaker*, and CONFUCIUS itself is regarded as the *hearer* which interpret the story (monologue) and present it to the ultimate hearer (the users of CONFUCIUS). As a monologue, the data flow of COUFUCIUS has only one direction, i.e. from the writers (hence the stories) to the system (c.f. Figure 1.1), and it does not receive users' interaction during its translation and presentation. However, there can be dialogue within the monologue between characters, which does not require CONFUCIUS' understanding and is passed to a text-to-speech engine.

Because of the important role of discourse models in reference resolution and disambiguating ungrammatical sentences it is necessary to combine it with CFG in the syntax parser of CONFUCIUS' natural language processing module.

4.7.4. Deriving visual semantics from text

One of the key problems arising in this project is deriving visual semantics from the 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. This is discussed further in the NLP module of CONFUCIUS in section 5.1.

One major problem that text-to-animation applications face is vagueness. Most text-to-graphics systems solve the vagueness in natural language by substituting an object type with a more specific object of the type. For example, to visualize the phrase *give her a toy* by substituting *a toy* with a specific toy such as a teddy bear. Although it can be solved by specific-general substitution which facility is provided by lexical semantic networks like WordNet, a vague representation of the meaning of the object/event may be appropriate in some situations. It will be advantageous for a semantic representation to maintain a certain level of vagueness. But it is almost impossible in vision, because the video modality itself always requires more specific and more information than the language modality.

4.8. Project schedule and current status

The work proposed previously required several steps to be carried out in order to achieve the desired objectives of CONFUCIUS. Appendix A Table 4.2 outlines the main tasks and schedule of this project. The tasks above the bolded line have been completed during the past one year. We surveyed previous research in related areas of language visualisation, interactive multimedia interfaces, multimodal storytelling, non-speech audio, autonomous agents, and cognitive science, and explored corresponding computer systems. We also reviewed the various methods of multimodal semantic representation, and multimedia fusion and coordination, and proposed ideas on improving them. Moreover, we completed the system design and some units' design (c.f. section 4.3 and 5.1). Having analysed a variety of available software of natural language processing, 3D graphic authoring tools and modelling languages, and programming languages, we decided the software/tools which will be used in CONFUCIUS. They are discussed in detail in the next chapter.

5. Software analysis

Rather than trying to build CONFUCIUS from scratch, we will make use of existing software tools for natural language processing, text-to-speech and 3D modelling and animation. An analysis of development tools for CONFUCIUS has begun early in the research, and the selection has already been decided in the current stage. Several potential tools have been identified and analysed. As CONFUCIUS is composed of several modules with different tasks to accomplish, the integration of the selected tools is important.

5.1. Natural Language Processing (NLP) tools

For language processing in CONFUCIUS, a part-of-speech tagger, syntactic parser and semantic analyser of English input are needed. Gate (Cunningham et al. 2002), CPK NLP Suite (Brøndsted 1999), PC-PATR (McConnel 1996), and WordNet (Fellbaum 1998) NLP tools may be useful here. We discuss Gate, PC-PATR, WordNet and text-to-speech engines for CONFUCIUS' syntax, semantics, and speech synthesis in this section.

5.1.1. Natural language processing in CONFUCIUS

The language processing module consists of a pre-processing module, part-of-speech tagger, a syntactical parser, a semantic interpreter, a stemmer, coreference resolution and temporal reasoning (see Figure 5.1). The task of the language processing component is to extract key ideas from the text and then represent them by frames (in semantic interpreter), which lists each action mentioned in the input sentence, the entity that performed the action, the object(s) of the action, and other information such as the time or location of the action. Temporal reasoning module analyses tenses of the input sentence and transduces events expressed in different tenses to sequential events, i.e., decides the temporal order of presentation events. For example, the temporal reasoning component will translate the sentence: "Having had a rich dinner, John walked along the river." into two sequential events: (1) *eat* (John, dinner) and (2) *walk* (John, river). The stemmer is used for both providing information for temporal reasoning and identifying objects (singular form) and events (verb root) which will be looked up in the primitives library in the animation generation process later (c.f. Figure 4.1).

5.1.2. Syntactic parser

Gate 2.0 (Cunningham et al. 2002) is an infrastructure for developing and deploying software components that process natural language. It is written in Java and available as open-source free software under the GNU library licence. It contains most of the language processing components that CONFUCIUS requires, such as sentence splitter and part-of-speech (POS) tagger. The POS tagger integrated in Gate is a Brill-style POS tagger (Hepple 2000), which produces a POS tag as an annotation on each word. The tagger uses a default lexicon and ruleset (the result of training on a large corpus taken from the Wall Street Journal). Both of these can be modified manually. Gate uses XML to represent the output of components, e.g. the output of POS tagger and sentence splitter. As we can see from Figure 5.1, we will use Gate in pre-processing (including tokenising and sentence splitting) and POS tagging.

PC-PATR (McConnel 1996) is a parser based on *context-free phrase structure grammar* and *unifications on the feature structures* associated with the constituents of the phrase structure rules. It uses bottom-up parsing with top-down filtering. A PC-PATR grammar consists of a set of rules and a lexicon. Each rule consists of a context-free phrase structure rule and a set of feature constraints, that is, unifications on the feature structures associated with the constituents of the phrase structure rules. The lexicon provides the items that

can replace the terminal symbols of the phrase structure rules, that is, the words of the language together with their relevant features. As shown in Figure 5.1, we will use PC-PATR in syntactic parsing and stemming.

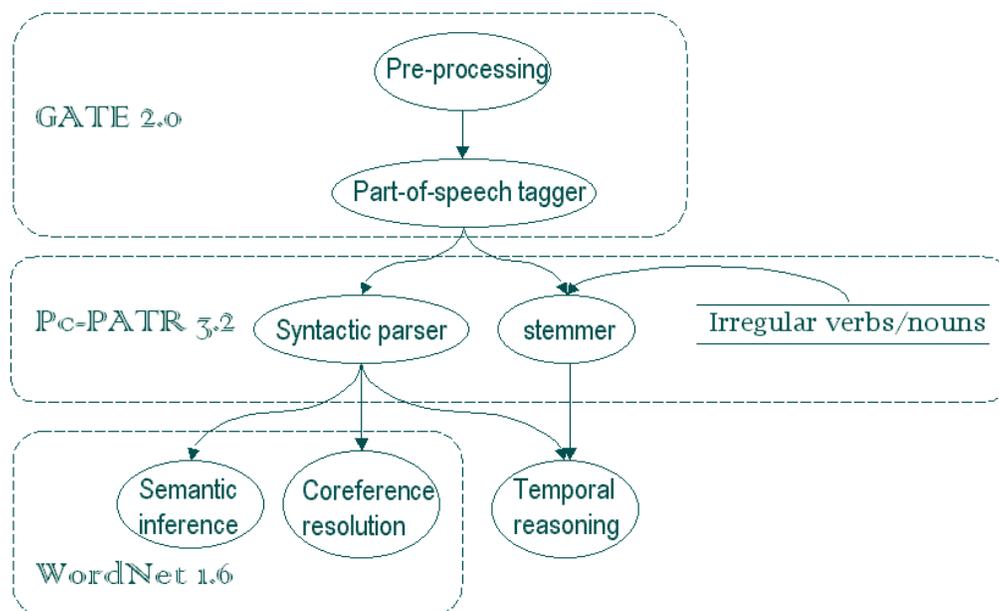


Figure 5.1: Using NLP toolkits in CONFUCIUS

5.1.3. Semantic inference

To fulfil the task of semantic inference as shown in Figure 5.1, we need a lexicon semantic database at least. Choice can be made from WordNet (Fellbaum 1998), and AI-ontologies such as CYC (Cyc 1997) or Sensus/Pangloss (Knight and Luk 1994). WordNet is a relational and taxonomic semantic network. It incorporates information on lexicalisation patterns, semantic components and conceptual inferences. Figure 5.2 lists the elementary relations distinguished between synsets in WordNet. They are of great advantage to semantic inference in language visualisation. *Hypernym* and *hyponym* are typically used for gaps when the language input is too general and for coreference resolution. For example, {toy} HYPERNYM {teddy bear} may bridge the gap in visualizing 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*.

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.
 {man, adult male} ==> {male, male person}

Hyponyms: synsets which are particular kinds of a synset, e.g.
 {weather, atmospheric condition} ==> {fair weather, 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}{perianth}
[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, hoof it} ==> {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 5.2: Basic relations between synsets in WordNet

WordNet also contains specifications of the argument structures of verbs, and includes a system for encoding conceptual dependencies between concrete nouns and verbs, in the form of semantic roles. As shown in Figure 5.3 the semantic roles of agent, patient, instrument, location and direction could be used in semantic inference to link verbs and their involved nouns (also see section 3.1). The syntagmatic relations in Figure 5.3 can be seen as specification of a potential semantic context for a verb, where the role-relations may coincide with grammatical contexts as well.

- Role_Agent/Involved_Agent
e.g. watch-dog ROLE_AGENT to guard
- Role_Patient/Involved_Patient
e.g. to teach INVOLVED_PATIENT learner
- Role_Instrument/Involved_Instrument
e.g. knife ROLE_INSTRUMENT to cut
- Role_Location/Involved_Location
e.g. school ROLE_LOCATION to teach
- Role_Direction/Involved_Direction
(Role_Source_Direction/Involved_Source_Direction)
e.g. to emigrate INVOLVED_SOURCE_DIRECTION one's country

Figure 5.3: Role/Involved syntagmatic relations in WordNet verbs

Furthermore, WordNet has argument-frames for verbs as listed in Figure 5.4. 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 represents the verb and the left and right strings the complementation patterns. This somehow manages certain tasks of syntactic parsing.

1. Sth --s
2. Sb --s
3. It is --ing
4. Sth is --ing PP
5. Sth --s sth Adjective/Noun
6. Sth --s Adjective/Noun

7. Sb --s Adjective
8. Sb --s sth
9. Sb --s sb
10. Sth --s sb
11. Sth --s sth
12. Sth --s to sb
13. Sb --s on sth
14. Sb --s sb sth
15. Sb --s sth to sb
16. Sb --s sth from sb
17. Sb --s sb with sth
18. Sb --s sb of sth
19. Sb --s sth on sb
20. Sb --s sb PP
21. Sb --s sth PP
22. Sb --s PP
23. Sb's (body part) --s
24. Sb --s sb to INFINITIVE
25. Sb --s sb INFINITIVE
26. Sb --s that CLAUSE
27. Sb --s to sb
28. Sb --s to INFINITIVE
29. Sb --s whether INFINITIVE
30. Sb --s sb into V-ing sth
31. Sb --s sth with sth
32. Sb --s INFINITIVE
33. Sb --s VERB-ing
34. It --s that CLAUSE
35. Sth --s INFINITIVE

Figure 5.4: Verb-frames in WordNet

Finally, semantic relations in WordNet can be augmented with specific features to differentiate the precise semantic implication expressed. Figure 5.5 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 a sine qua non for semantic inference in disambiguating process of visualisation, for example, when a door is mentioned in language input, the animation generator needs to know which door, a door of car, a door of room, or a door of airplane, should be created in the story world. The negation of implications expressed by relations provides a potential to extend the visual library of the system. In the case of the example in Figure 5.5, 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 children's when they are told that 'ape is like monkey but it has no tail'.

- Conjunction or disjunction of multiple relations of the same type

Example 1: *airplane*

HAS_MERO_PART *door* conjunctive

HAS_MERO_PART *engine* conjunctive

Example 2: *door*

HAS_HOLO_PART *car* disjunctive

HAS_HOLO_PART *room* disjunctive

HAS_HOLO_PART *airplane* disjunctive

- Factivity of causal relations

Example 1: *kill* CAUSES *die* factive
Example 2: *search* CAUSES *find* non-factive

- Negation of implications expressed by relations
Example 1: *monkey* HAS_MERO_PART *tail*
 ape HAS_MERO_PART *tail* not

Figure 5.5: Relations with features to differentiate semantic implication

Due to the above facilities that WordNet provides, we will use it for our semantic inference as shown in Figure 5.1.

5.1.4. Text-to-speech

For Text-To-Speech (TTS) software, choice 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 TTS engine should take operating system platforms into account since some of them might only work on specific platforms.

There are two ways to synchronize a character's lip animation with his speech (through a TTS engine). The first is to obtain estimates of word and phoneme timings and construct an animation schedule prior to execution (time-driven). The other 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 (event-driven). The first approach must be used for TTS engines such as Festival (Taylor et al. 1998) which is also used by the CSLU toolkit in Baldi's lip movements (c.f. Figure 3.14), while the second must be used with TTS engines such as Microsoft Whistler (Huang et al. 1996).

5.2. Three dimensional graphic authoring tools and modelling languages

5.2.1. Three-dimensional animation authoring tools

There are a variety of prospective 3D graphic authoring tools that can create static and dynamic objects in a virtual story world. SDKs (Software Development Kits) for graphics for Microsoft Windows platforms are Microsoft DirectX (DirectX 2002) and Silicon Graphics OpenGL (OpenGL 2002). Both of them provide low-level 3D-graphics programming, Immediate Mode (DirectX) and basic OpenGL, and a high-level programming interface, Retained Mode (DirectX) and Performer (OpenGL).

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 were introduced in section 3.4. The third is the function of importing animations made in a 3D-Studio like 3D Studio Max (Ethier and Ethier 2002), which is provided by Retained Mode. Hence one can create the animations in a tool which has been designed to produce animation of 3D objects. The drawback is that once the animations have been included into a Retained Mode object there is no way to manipulate the animations.

Decisions on graphic authoring tools have been made in the current stage as shown in Appendix A. We consider using 3D Studio Max and Poser 4.0 (Poser 2002) to create the actors, props and stage and exporting them to VRML 2.0/97 format that give more control on the objects in the 3D virtual world. Of course we will use 3D models available on the Internet as much as possible to save substantial efforts on graphic

design. Microsoft Agent (2002) is also used to create the narrator and minor characters in stories since it provides limited movements.

5.2.2. Three-dimensional graphic modelling language — VRML

A graphic modelling language is required to represent visual data symbolically and combine the prefabricated visual primitives created by authoring tools to make a virtual world. The representation of visual information of an object usually consists of its color, texture, geometric shape, size, position and its sub-components etc. The syntax of VRML (Virtual Reality Modelling Language) and its hierarchical structure suit for the representation of the visual information aforementioned. Figure 5.6 defines a world with only one child—a cube called Box01 using Tiles.jpg in the same directory as its texture. Its attributes like size, position, and orientation are specified by size, translation, and rotation respectively.

```
#VRML V2.0 utf8

DEF Box01 Transform {
  translation 0 10 0          # position
  rotation -0.25 1 0.15 -1.1 # orientation
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.54 0.2 0.2 # color
          ambientIntensity 0.1
          shininess 0.2875
          transparency 0
        }
        texture ImageTexture { # texture
          url "Tiles.jpg"
        }
      }
      geometry Box { # shape
        size 50 50 70 # size
      }
    }
  ]
}
```

Figure 5.6: An example of VRML hierarchical structure

Using VRML can also save effort in implementing naïve physics such as collision detection, i.e. any physical object may not pass through another one, which is a deficiency of existing 3D animation systems like SONAS (Kelleher et al. 2000). VRML provides a built-in collision detection mechanism (proximity sensors and collision detection) to solve this problem. However, proximity sensors and collision detection in VRML apply only to users (avatar(s)); no objects or creatures in a virtual world have any way of directly detecting the location of a wall, and nothing prevents an animated character from walking right through walls. Therefore, it is required for program animated characters to have some *knowledge* of where the walls and other solid objects in the world are, or design motion algorithms to stay away from certain areas, to avoid walking across ponds, for example.

Using Background node to build stage setting

VRML also provides stage-like facilities that suit the story presentation purpose of CONFUCIUS. Background binding (Figure 5.7) facilitates similar operations to scene-changing in plays. When the 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 node. 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`; 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 it was at the top of the stack, the node sends out an `isBound` event with value `FALSE`, and the new top of the stack (formerly the second item on the stack) becomes bound and sends out an `isBound TRUE` event. If there is only one background in the world, it never needs to be explicitly bound or unbound; the first Background node in the file is automatically bound.

Figure 5.7 lists the syntax for the Background node in VRML. The event `set_bind` is used to bind or unbind a background. The Background node is more powerful than conventional theatre design by providing background for all six sides of the stage, that is, besides setting the back background (the usual one) it also allows to set front, left, right background, top background (for sky), and bottom background (for the ground).

```
Background {
    eventIn SFBool set_bind
    exposedField MFFloat groundAngle [] # [0,π/2]
    exposedField MFColor groundColor [] # [0,1]
    exposedField MFString backUrl []
    exposedField MFString bottomUrl []
    exposedField MFString frontUrl []
    exposedField MFString leftUrl []
    exposedField MFString rightUrl []
    exposedField MFString topUrl []
    exposedField MFFloat skyAngle [] # [0,π]
    exposedField MFColor skyColor 0 0 0 # [0,1]
    eventOut SFBool isBound
}
```

Figure 5.7: Syntax for Background node

To present a multi-act play or a story with several backgrounds in CONFUCIUS, we may list all Background nodes in the begin of the `.wrl` file and bind one after each act in order. A more advanced feature in Background node of VRML than that in traditional drama is the background could be changed within an act using the ordinary events-and-routes method. This feature enlarges the usage of background and enables us to put some properties in background and hence reduce the 3D modelling of unimportant objects to 2D imaging.

Using interpolators and ROUTE to produce animation

VRML defines piecewise linear interpolators (`ColorInterpolator`, `PositionInterpolator`, `OrientationInterpolator`, `ScalarInterpolator` etc.) for keyframed animation. They can be used to change values of objects' properties like color, position, and size. The `ROUTE` statement in VRML is a construct for establishing event paths between objects (nodes), i.e., an object (node) may generate events (`eventOut`) and send it to any other objects (`eventIn`) connected via `ROUTE` statements. This message-passing facility allows the communication between characters and objects. Figure 5.8 lists two examples of `ROUTE` statement: `TIMER1` controls a box' movement and `TIMER2` starts a character's motion.

```
ROUTE TIMER1.fraction_changed TO boxPositionIntp.set_fraction
ROUTE TIMER2.isActive TO MAN_ACTION.set_animationStarted
```

Figure 5.8: 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 (c.f. section 3.4). Secondly, animating the position and orientation of a viewpoint achieves some special movie-editing effects like *cut* or to guide the viewer to explore around in the world. In the former case, setting `jump` field of a `Viewpoint` to `TRUE` will cut the viewpoint to the next location (see the first example in Figure 5.9). The latter case can bind 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 present a story in the first person *I* (see the second example in Figure 5.9). 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. In the code of Figure 5.9, the first example simply cuts a viewpoint to the next location (bus stop 1), whilst the second guides the viewer's sight gradually as if (s)he is in an elevator.

```
DEF CUT_TO_STOP1 Viewpoint {
    position 10 5 30
    fieldOfView 1.8
    description "initial viewpoint, will be cut to bus stop 1"
    jump TRUE
}

DEF IN_ELEVATOR Viewpoint {
    position 38 17 -40
    orientation -0.02 1 0.05 2.36
    fieldOfView 0.79
    description "Elevator system overview"
    jump FALSE
}
```

Figure 5.9: Examples of `Viewpoint` node

Most stories are told in the first person (a character's point-of-view) and the third person (a neutral non-character point-of-view). Setting a viewpoint properly can fulfil the narrative on any desired point-of-view in storytelling.

In addition, VRML may spare efforts on media coordination since its *Sound node* is responsible for describing how sound is positioned and spatially presented within a scene. It enables the animation

generator to locate a sound at a certain point and make the viewer aware of the sound's location. This function is useful in presenting non-speech sound effects in storytelling, e.g. the sound of a car when it passes by. The Sound node brings the power to imbue a scene with ambient background noise or music, as well. It can also describe a sound that will fade away at a specified distance from the Sound node by ProximitySensor.

In summary, as a graphic modelling language VRML can make an animation system meet the following sine qua non: (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 such as walking, and a library of versatile built-in motion methods should be available.

5.2.3. Java in VRML Script node

A programming language (or script language) is needed to coordination with the graphic modelling language to define animation, link events occurring on different objects/characters, and 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 have maximum portability across browsers.

There are two specified methods to use Java with VRML. One method is to use Script nodes. There is a normative Java script¹⁶ 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, currently using Live Connect¹⁷, 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 plug-in interface that allows embedded objects on web page to communicate with each other. Figure 5.10 illustrates the communication within the Netscape browser. We use internal Script nodes in CONFUCIUS because it does not focus on user interaction when it tells a story or plays a drama script and Java Script nodes are enough for the purpose of creating (modifying) the story world dynamically.

For the VRML browser, we have compared and analysed the commonest browsers, Blaxxun Contact, Cosmos player and Parallelgraphics' Cortona. 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, finally we decide to use Parallelgraphics' Cortona 4.0 (for Netscape) 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 their lip movements when they are speaking, we are limited only by the processing speed of Java

¹⁶ Here Java script is not Javascript, but script in Script node of VRML written in the Java™ language. Usually, Javascript (vrmlscript—a subset of Javascript) and Java are the most popular languages used in the VRML Script node.

¹⁷ Live Connect is a Netscape product which enables communication between JavaScript and Java applets in a page. It also enables communication between JavaScript and plug-in (e.g. a VRML browser). LiveConnect may be used as an alternative to AWT to manage applet GUI with VRML.

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.

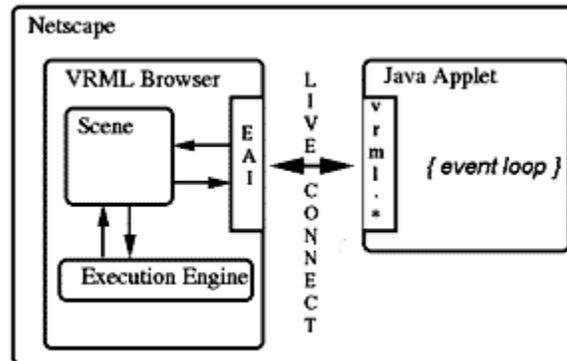


Figure 5.10: Overview of communication between VRML and Java applet

5.2.4. Basic narrative montage and their implementation in VRML

In this section we explore the elementary film-editing techniques listed in Smith and Bates' (1989) interactive fiction research and sketch possibilities for their implementation in VRML.

1. *Cut*, probably the most fundamental montage technique, joins separated shots together in the editing process. It is easy to be implemented in VRML, by just using guided Viewpoint with the field `jump` set to `TRUE` (see Figure 5.9) or the `replaceWorld` (`MFNode nodes`) function in the VRML browser API (Application Programming Interface).
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. However, unlike with 2D media, it is infeasible to implement this in a 3D VRML world. We have to use cut to substitute for lap dissolve in CONFUCIUS.
3. *Pan shots*. In a pan shot a stationary camera turns horizontally and smoothly scans the scene to reveal new areas. It can also be achieved in VRML by guided Viewpoint with the field `jump` set to `FALSE` (see Figure 5.9).
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. These camera angles can be defined by setting the fields `position` and `orientation` in Viewpoint node of VRML.
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 the chase scene in D.W. Griffith's (an inspiration for film-edit technique from Dickens' writing style) *A Girl and Her Trust* or to establish links between the scenes presented in parallel. This technique can be achieved by some VRML browser API functions such as `loadURL` (`MFString url`, `MFString parameter`) and `replaceWorld` (`MFNode nodes`) to switch between two `.wrl` files.

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). It can be implemented by `loadURL(MFString url, MFString parameter)` and using `Script` with `TimeSensor` to control the returning to the present.
7. *Subliminal flash 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*. The implementation of subliminal flash shot is same as flashback except the value of `cycleTime` of `TimeSensor` is shorter than that of flashback.
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. All these can be controlled through proper setting of `TimeSensors`.
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 which 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* 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 expect the exclusion of the non-emphasized details is more deliberate. It can be accomplished by simply adding a physical pipe-like object near the active `Viewpoint` position, pointing it to the emphasized object, to allow the viewer see through.
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 character's imagery, i.e. opening a second window and present 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.
12. *Voiceovers* 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. CONFUCIUS will support any of these narrations. The default is no narrator. It also supports omniscient external narrator through a speak-aside talking head, and character as narrator through `Sound` nodes without presenting the owner of the voice and his corresponding lip animation.

As enumerated above, we see the difficulty in an intelligent storytelling system like CONFUCIUS is not presenting contents by these montage techniques, but selecting contents, and more ambitious, choosing available techniques automatically to achieve its communication 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, CONFUCIUS 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, 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.

As we discussed in the *imagery* technique (11) previously, computer graphics supplies us with more channels to present information such as opening a new window which can be more effective than conventional cinematic cross-cutting, flashback, and subliminal shots.

5.3. Using autonomous agents to model the actors

The animation of actors includes their body poses, gestures, and facial expression (e.g. lip movement). 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. All these actors' movements have been researched in intelligent agents, such as Baldi in the CSLU toolkit (CSLU 2001) and Cassell et al.'s (2000, 2001) BEAT. The following agents tools might be used to model characters' animation in CONFUCIUS.

Microsoft Agent (2002) provides a set of programmable software services that supports the presentation of interactive animated characters within Microsoft Windows. It enables developers to incorporate conversational interfaces, that leverages 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 synthesized speech, recorded audio, and/or text in a cartoon word balloon. One advantage of agent characters designed by Microsoft Agent is they provide higher-levels of a character's movements often found in the performance arts, like blink, look up, look down, and walk.

Table 5.1 shows a sample character--VRGirl's details. There are not only fundamental locomotion but also emotions such as *confused*, *pleased*, and *sad* in the 59 defined animations of VRGirl. It puts forward another important requirement for our characters: emotion. How to draw on emotions from the story input? This itself can be an independent research in deep natural language understanding. Fortunately, most children's stories, explicitly describe characters' emotion. As shown in the story *The Tortoise and the Hare* in Figure 5.11, italic words 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 (happy, sad, angry etc.) 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 audience. Obtaining emotional information from natural language is a challenge in natural language understanding, which deserves to be one direction of further research.

BEAT (Cassell et al. 2001), another animator's tool which was incorporated in Cassell's REA (see section 2.4), allows animators to input typed text that they wish to be spoken by an animated human figure. It facilitates behavior generation task in dialogue-rich stories/scripts. However, BEAT has 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). This is also the deficiency of all presentation agents including

Microsoft Agent. Typically, the actions that most presentation agents can perform comprise the following types:

	Name:	VRGirl
	File Name:	Vrgirl.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 Process Read RestPose Sad Search Searching Suggest Surprised Think Wave Write ...
0 Looping Animations:		

Table 5.1: Character VRGirl specification

- ❑ High-level presentation acts. This group of actions includes pointing gestures, speaking and the expression of emotions, e.g. `VRGirl.Explain`, `VRGirl.GestureLeft`, `VRGirl.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. Baldi's (see section 3.5.2) periodic blinking, mouth and eye movements, and eye tracking when he's 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. The method we proposed in section 3.4 is to break up complex movements into simpler ones until all of them are available basic acts.

These actions are more than enough for the narrator and minor characters of CONFUCIUS, but they may be limited for the protagonists in stories. To create more complicated movements that agents do not provide, we have to resort to 3D character authoring tools like Poser and 3D Character Studio. These tools could be used to implement protagonists in CONFUCIUS.

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 5.11: 'The Tortoise and the Hare' from Aesop's Fables

6. Conclusion and future work

In this chapter we conclude by first summarizing the research proposal, the literature review and the main research areas and contribution of this project. We then discuss the software analysis focussing on tools and programming languages to be used, following by potential applications of the research. Finally, we give some thoughts on promising future directions.

The objective of the work described in this research report is the proposed development of CONFUCIUS, an intelligent multimedia storytelling interpretation and presentation system that automatically generates multimedia presentations from natural language stories or drama/movie scripts. The storytelling employs several temporal media such as animation, speech and sound (including music) for the presentation of stories. The primary objectives of CONFUCIUS are: (1) to interpret natural language story or movie (drama) script input and to extract concepts from the input, (2) to generate 3D animation and virtual worlds automatically, with speech and non-speech audio, and (3) to integrate the above components to form an intelligent multimedia storytelling system for presenting multimodal stories. These objectives of CONFUCIUS meet four challenging problems in language animation: (1) mapping language primitives with visual primitives to present objects, actions, and events, (2) visualisation of a story in natural language requires a gigantic knowledge base on ‘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. The architecture, data flow, and issues in core modules of CONFUCIUS such as animation generation and multimedia presentation planning are also introduced in this report.

Previous research in related areas of language visualisation, multimodal storytelling, intelligent multimedia agents and interfaces, non-speech audio, and cognitive science is surveyed, and corresponding computer systems are explored. Various representations of multimodal semantics and visual semantics (e.g. frames and XML for multimodal semantics, and conceptual dependency and x-schemas for visual semantics), and methods of multimedia fusion and coordination in these systems are also reviewed. Most of the current intelligent multimedia systems mix text, static graphics (including map, charts and figures) and speech (some with additional non-speech audio) modalities. Static graphical displays in these systems constrain presentation of dynamic information such as actions and events. To make references on a static graph, some use highlighting and temporally varying effects like moving, flashing, hopping, zooming, and scaling in their presentations. The graphics of most systems which present animation is two dimensional or ready-made, such as KidsRoom (Bobick et al. 1996) and Gandalf (Thórisson 1996). Only 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) have three-dimensional image quality. However, SI/SONAS needs user’s intervention to navigate and modify the 3D world and hence its performance is rather like a 3D browser which responses to speech commands. The application domain of SI/SONAS is limited to scene description and spatial relationships. Cassell’s SAM and REA (Cassell et al. 2000) focus on simulating humanoid behavior in conversation, but they are limited to human-computer interface applications. Although Larsen and Petersen’s interactive storytelling environment does present 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. Albeit Narayanan’s language animation may generate 3D animation automatically, the quality of its images (iconic feature) is inadequate. We also investigated techniques of non-speech audio and decided to use auditory icons and to select music clips for different story situations. Our research in the form of CONFUCIUS will overcome the limitations of previous intelligent multimedia systems and automatically present stories in realistic 3D live animations.

Accordingly, CONFUCIUS' language animation has three main areas of contribution to solve the above problems: (1) multimodal semantic representation, (2) multimodal fusion and coordination, and (3) automatic animation generation. Existing multimodal semantic representations within various intelligent multimedia systems may 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, which links language modalities to visual modalities, is proposed in this report. CONFUCIUS' multimodal semantics is composed of language, visual and non-speech audio modalities. Between the general multimodal semantics and each special modality there are two levels of representation: one is a high-level multimodal semantic representation which is *media-independent*, the other is an intermediate level *media-dependent* representation. CONFUCIUS will use an XML-base representation for high-level multimodal semantics and an extended predicate-argument representation for intermediate semantic representation which connects language with visual modalities. This research will also introduce a new method of multimedia fusion and coordination for multiple output modalities, all of which are in temporal media. For example, the media allocation of CONFUCIUS is based on the visualisation and observability of concepts and multimedia presentation planning allows backward propagation for unsuccessful realisation. In addition, the work will also advance research in automatic text-to-graphics generation. CONFUCIUS integrates and improves novel theories and techniques in the areas of natural language processing, intelligent multimedia presentation, and language visualisation.

CONFUCIUS will be developed using existing software tools such as Gate and WordNet for natural language processing, 3D Studio Max for object modelling, Microsoft Agent and Poser for authoring humanoid animation, and Java Speech API for speech synthesis. Virtual Reality Modelling Language (VRML) will be used to model 3D graphics. Java programs will generate/assemble VRML codes and integrate all these components into the system. To demonstrate and test CONFUCIUS some example children's stories (e.g. *Alice in wonderland*) and scripts (e.g. Beckett's short plays) will be given as input and automatically presented. Prospective practical applications of CONFUCIUS could be a wide variety of applications, such as children's education, multimedia presentation, movie/drama production and training, script writing, animated e-postcards, and game applications.

There are a number of problems that have to be solved before CONFUCIUS can be widely applied. To accomplish more *intelligent* storytelling, further research may 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 CONFUCIUS, as discussed in research on emotion understanding and generation within Mueller's (1998) *understanding agency*. Moreover, there are various directions in which CONFUCIUS has potential expansions. The first is 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 the storytelling system 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 stories/scripts (that 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.

References

- André, E., J. Müller and T. Rist (1996) The PPP Persona: A Multipurpose Animated Presentation Agent. In *Advanced Visual Interfaces*, T. Catarci, M.F. Costabile, S. Levialdi and G. Santucci (Eds.), 245-247, New York, USA: ACM Press.
- 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.
- Badler, N., Webber, B., Kalita, J. and Esakov, J. (1991) Animation from Instructions. In *Making them Move*, Badler, Barsky and Zeltzer (Eds.), 51-93, Cambridge, MA: MIT Press.
- 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.
- Beckett, S. (1984) *Collected shorter plays*. London, U.K.: Faber and Faber.
- 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.
- Berners-Lee, T., J. Hendler and O. Lassila (2001) The Semantic Web. *Scientific American*, May 2001. <http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2> Site visited 12/09/2002.
- 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.
- 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.
- 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, Morgan Kaufman.

- Bolt, R.A. (1987) *Conversing with Computers*. In *Readings in Human-Computer Interaction: A Multidisciplinary Approach*, R. Baecker and W. Buxton (Eds.), California: 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., Bly, S., Frysinger, S., Lunney, D., Mansur, D., Mezrich, J. and Morrison, R. (1985) Communicating with Sound. In *Proceedings of The Human Factors in Computing Systems (CHI-85)*, New York, 115-119.
- Cassell, J., C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, 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.
- 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.
- CLIPS (2002) CLIPS: A Tool for Building Expert Systems. <http://www.ghg.net/clips/CLIPS.html> Site visited 28/09/2002.
- Collins, M. (1999) *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.
- Coyne, B and R. Sproat (2001) WordsEye: An Automatic Text-to-Scene Conversion System. Computer Graphics Annual Conference, *SIGGRAPH 2001 Conference Proceedings*, Los Angeles, Aug 12-17, 487-496.
- CSLU <http://cslu.cse.ogi.edu/toolkit/index.html> Site visited 18/08/2002.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C. and Dimitrov, M. (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.

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.

Ethier, S.J., Ethier, C.A. (2002) *3D Studio MAX in Motion: Basics Using 3D Studio MAX 4.2*. Prentice Hall.

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 (1991a) COMET: Generating coordinated multimedia explanations. In S. P. Robertson, G. M. Olson, and J. S. Olson (Eds.), 449-450, *CHI'91 Conference Proceedings*, Reading, MA: Addison-Wesley.

Feiner, S.K. and K.R. McKeown (1991b) 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.

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.

Hartman, J. and J. Wernecke (1996) *The VRML 2.0 Handbook – Building Moving Worlds on the Web*. Silicon Graphics, Inc. Harlow, England: Addison-Wesley Publishing Company.

Hayes-Roth B. and R. van Gent (1997) Story-making with improvisational puppets. In *Proceedings of the First International Conference on Autonomous Agents*, 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.

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.

Jackendoff, R. (1987) On Beyond Zebra: The Relation of Linguistic and Visual Information. *Cognition*, 26(2), 89-114.

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. Ms., IBM T. Watson Research Center, Yorktown Heights, NY.

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.

Knight K. and S. Luk. (1994) Building a large knowledge base for machine translation. In *Proceedings of the American Association of Artificial Intelligence Conference AAAI-94*, Seattle, WA.

Kosslyn, S.M. (1994) *Image and Brain: The Resolution of the Imagery Debate*. Cambridge, MA: MIT Press.

Laird J.E. (2001) Using a Computer Game to Develop Advanced AI. *Computer*, 34(7), 70-75.

Larsen, C.B. and Petersen, B.C. (1999) Interactive Storytelling in a Multimodal Environment. Technical Report, M.Sc. Thesis, Aalborg University, Institute of Electronic Systems.

Lee, Mark. G. (1994) A Model of Story Generation. M.Sc. Thesis, Dept. of Computer Science, University of Manchester.

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: W.H. Freeman.
- 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.) (1995a) *Integration of Natural Language and Vision Processing (Volume I): Computational Models and Systems*. London, U.K.: Kluwer Academic Publishers.
- Mc Kevitt, P. (Ed.) (1995b) *Integration of Natural Language and Vision Processing (Volume II): Intelligent Multimedia*. London, U.K.: Kluwer Academic Publishers.
- Mc Kevitt, P. (Ed.) (1996a) *Integration of Natural Language and Vision Processing (Volume III): Theory and grounding representations*. London, U.K.: Kluwer Academic Publishers.
- Mc Kevitt, P. (Ed.) (1996b) *Integration of Natural Language and Vision Processing (Volume IV): Recent Advances*. 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.
- Microsoft Agent (2002) <http://www.microsoft.com/products/msagent/>, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msagent/agentstartpage_7gdh.asp Sites visited 14/08/2002.
- 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.

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 (2002) <http://www.opengl.org/> Sites visited 25/09/2002.

OWL (2002) Feature Synopsis for OWL Lite and OWL, W3C Working Draft. <http://www.w3.org/TR/2002/WD-owl-features-20020729/> Sites visited 21/08/2002.

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.

Poser (2002) <http://www.creativepro.com/software/home/211.html> Sites visited 25/09/2002.

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.

SALT (2002) <http://xml.coverpages.org/salt.html> Sites visited 20/08/2002.

SAPI (2002) <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcesapi/hm/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, Ill.: 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.
- 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.
- 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.
- Stock, O. and the AIFresco Project Team. (1993) AIFresco: 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.
- Taylor, P., Black, A. and Caley, R. (1998) The architecture of the Festival Speech Synthesis system. *Proceedings 3rd ESCA Workshop on Speech Synthesis*, 147-151, Jenolan Caves, Australia.
- Thórisson, K. (1996) Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills. Ph.D. thesis, Media Arts and Sciences, Massachusetts Institute of Technology.
- 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.
- 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.ilc.uva.nl/EuroWordNet/corebcs/topont.html> Site visited 15/08/2002.
- W3C (2002) <http://www.w3.org/XML/> Site visited 20/08/2002.
- 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, S. Bandyopadhyay, W. Graf and T. Rist (1992) WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation. In *Communication from Artificial Intelligence Perspective: Theoretical and Applied Issues*, J. Slack, A. Ortony and O. Stock (Eds.), 121-143, Berlin, Heidelberg: Springer Verlag.

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.), DLR, Berlin, Germany, October 2001, 23-34.

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.

Worton, M. (1994) Waiting for Godot and Endgame: theatre as text. In *The Cambridge Companion to Beckett*, J. Pilling (Ed.), Cambridge, U.K.: Cambridge University Press.

Zhou, M.X. and S.K. Feiner (1998) IMPROVISE: automated generation of animated graphics for coordinated multimedia presentations. In *Cooperative Multimodal Communication Second International Conference (CMC'98)*, H. Bunt and R. Beun (Eds.), 43-63, Tilburg, The Netherlands.

Appendix A: Project schedule

	2001	2002				2003				2004		
Research Activities	Oct-Dec	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec	Jan-Mar	Apr-Jun	Jul-Sep
Literature survey												
Writing Chapter 2 ‘Literature Review’												
Analysis and selection of tools												
Learning how to make animation manually using some graphic tools												
Selection of 3D authoring tools												
Selection of NLP tools												
Selection of other reusable components (e.g. Agents) and familiarisation with their usage												
System design												
Unit implementation												
Text-to-animation (visualize sentences/short paragraph for scene description)												
vision-speech coordination (dialogues between actors)												
parse natural language stories to scripts												
other units												
Integration and testing												
Write up PhD thesis												
Improving system												
Modifying thesis												

Table 4.2: Project schedule