A full-scale semantic content-based model for interactive multimedia information systems

by

HARRY WAYNE AGIUS

A thesis submitted for the degree of Doctor of Philosophy

September 1997

The London School of Economics and Political Science

Dedicated to my mother, Rose,

and the memory of my father, Harry

For forty-odd years in this noble profession

I've harboured a guilt and my conscience is smitten,

So here is my slightly embarrassed confession –

I don't like to write, but I love to have written.

— Michael Kanin, 'My Sin', quoted in *The Author* (Autumn 1979)

ACKNOWLEDGEMENTS

I would first like to thank God for allowing me to pursue my ambitions and for providing me with the abilities to achieve them. I am indebted to my supervisor, Dr Marios C. Angelides, for his invaluable advice, support, and encouragement. I am eternally grateful to my mother, Rose, for her patience and for always encouraging me to carry on. Finally, many thanks to my family and all of my friends for understanding that, most of the time at least, research often takes over your life completely!

ABSTRACT

Issues of *syntax* have dominated research in multimedia information systems (MMISs), with video developing as a technology of images and audio as one of signals. But when we *use* video and audio, we do so for their *content*. This is a *semantic* issue. Current research in multimedia on semantic content-based models has adopted a *structure-oriented* approach, where video and audio content is described on a frame-by-frame or segment-by-segment basis (where a segment is an arbitrary set of contiguous frames). This approach has failed to cater for semantic aspects, and thus has not been fully effective when used within an MMIS. The research undertaken for this thesis reveals seven semantic aspects of video and audio: (1) explicit media structure; (2) objects; (3) spatial relationships between objects; (4) events and actions involving objects; (5) temporal relationships between events and actions; (6) integration of syntactic and semantic information; and (7) direct user-media interaction.

This thesis develops a full-scale semantic content-based model that caters for the above seven semantic aspects of video and audio. To achieve this, it uses an *entities of interest* approach, instead of a structure-oriented one, where the MMIS integrates relevant semantic content-based information about video and audio with information about the entities of interest to the system, e.g. mountains, vehicles, employees. A method for developing an interactive MMIS that encompasses the model is also described. Both the method and the model are used in the development of ARISTOTLE, an interactive instructional MMIS for teaching young children about zoology, in order to demonstrate their operation.

TABLE OF CONTENTS

CHAPTER TWO: A FULL-SCALE SEMANTIC CONTENT-BASED MODEL.69

LIST OF FIGURES

LIST OF TABLES

Chapter One

DIGITAL VIDEO AND AUDIO: SYNTAX VERSUS SEMANTICS

"High thoughts must have high language."

— Aristophanes

The main characteristic of a multimedia information system (MMIS) is its ability to deal not only with alphanumeric data, as in traditional databases, but also with still and full-motion video, audio, graphics, and animation (Fox, 1991; Narasimhalu and Christodoulakis, 1991; O'Docherty and Daskalakis, 1991; Price, 1991; Berra et al., 1993; Furht, 1994; Grosky, 1994; Jain, 1994a; Jain, 1994b; Triebwasser, 1994; Furht and Milenkovic, 1995; Rodriguez and Rowe, 1995; Steinmetz and Nahrstedt, 1995; Angelides and Dustdar, 1997; Dustdar and Angelides, 1997). MMISs are thus faced with the challenge of handling new data types and their relationships together with the traditional ones, i.e. retrieval and processing mechanisms for static media, such as text and graphics, as well as for dynamic, time-variant media, such as video and audio (Burrill et al., 1994).

MMISs achieve this by representing all information uniformly, as a bit stream. Unfortunately, much work has been dominated by this bit stream: video has developed thus far as a technology of images, and audio as a technology of signals. These issues of *syntax* do not address how to *use* video or audio effectively within an MMIS. Video and audio will only become effective parts of everyday computing environments when they can be used with the same ease as text. We do not use video just because the images are steady or focused, or audio because it sounds crisp or stereophonic. We use these media for their *content*. This is a *semantic* issue. Without knowledge of its content a bit stream remains a bit stream that cannot be interpreted. To use and interact with it, the bit stream must be converted into a form that can be understood.

The current situation is analogous to that of information processing before artificial intelligence. Artificial intelligence introduced advanced symbolic reasoning, enabling software to concentrate on *what* the problem is, rather than *how* the problem is manifest (Feigenbaum, 1996). Similarly, a more semantics-aware indexing of multimedia information emphasises *what* is taking place within the media, i.e. the meaning of the content, as opposed to the format of *how* this content is stored, which is an issue of syntax. Unlocking the potential of multimedia through semantics enables us to go someway toward aligning the processing of temporal (i.e. video and audio), textual and other data types.

Thus far, then, the field of multimedia has concentrated on the 'what' at the expense of the 'how'. The result has been that while current MMISs can handle *multimedia syntax*, they cannot handle well *multimedia semantics* and often lack architectural support for semantic multimedia computation in terms of comparing, combining, and processing semantic multimedia elements. Consequently, any integration of media today consists of displaying objects, each embedded in a separate medium, in a window on a screen, or, for audio, recorded independently and connected only by synchronisation (Blattner, 1994; Agius and Angelides, 1997a).

Unfortunately, the complex nature of video and audio has made focusing on content-based semantics a much more difficult problem than has been the case with text (Aigrain et al., 1996; Lee et al., 1997). Current research has adopted a *structure-oriented* approach, where the models describe the content of the video and audio stream on a frame-by-frame or segment-by-segment basis (a segment is an arbitrary set of contiguous frames). This approach has suffered from a number of weaknesses:

- The explicit media structure, i.e. the explicit way in which sequences of video and audio are split and grouped together, in all models is often very basic and predominantly video-oriented. Audio is frequently underspecified compared to video, or is disregarded altogether. Uniform processing on both video and audio is rarely employed. Instead, video and audio are either treated in unison as one inseparable unit or audio is left unspecified.
- Although most content-based models represent content objects within the medium, very few of the models are concerned with the *location* of these objects, e.g. through the use of on-screen co-ordinates. Frequently, content-based multimedia models have been satisfied with merely representing the *presence* of a content object in a particular frame or set of contiguous frames.
- At best, only limited spatial relationships between content objects that are simultaneously on-screen (or are simultaneously heard) may be determined implicitly from the other semantic information represented.
- The representation of events and actions has received limited attention, with the semantic information taking on a very unstructured format. Semi-structured information makes processing, e.g. in terms of identifying and comparing terms, more

difficult than if the information were fully structured. Tighter integration between these and other referenced elements becomes restricted as a result.

- Temporal relationships between events and actions (e.g. A occurs during B, A occurs before B) has not been adequately addressed in content-based models, with many models providing no capability for this aspect.
- The models do not seek to integrate the media (i.e. the video and audio) streams with their associated semantics, which puts an unnecessary burden on the processing requirements of the system utilising model.
- Only a handful of models provide the ability for the user to directly interact with the media (i.e. the video and audio). That is, for the user to interact with the content objects.

Moreover, existing models concentrate on specific semantic aspects, such as the representation of content objects. Consequently, while the structure-oriented approach is certainly useful for 'virtual browsing' paradigms and sequential playing of multimedia in CD-ROM movies, this approach has clearly been less effective when adopted for use within an interactive MMIS. This thesis addresses the above problems by developing a full-scale semantic content-based model that encompasses the semantic aspects of video and audio.

The following section distinguishes between the syntax and semantics of video and audio. Relevant research on semantic content-based multimedia models is then reviewed. Next, these models are interrogated to see how well they cater for the various semantic aspects of video and audio. The chapter then presents the research objective and research method. Finally, the chapter concludes with an overview of the remaining thesis chapters.

1.1 SYNTAX AND SEMANTICS IN MULTIMEDIA INFORMATION SYSTEMS

This thesis uses the term *syntax* to refer to the organisation and representation of information in MMISs, whether this be the bit stream (e.g. text represented through ASCII codes, video represented through formats such as MPEG, and audio represented through Wave and other formats) or objects presented on-screen. In contrast, the term *semantics* is used to refer to the meaning depicted within videos and audios. Multimedia semantics has proved more problematic for MMISs.

Because multimedia semantics are manifested through multimedia syntax, there is some overlap between the application of the two terms. For example, while the arrangement of objects on-screen is considered to be syntactic, this arrangement may also have a particular meaning which is semantic. To illustrate this, consider a video that shows the motherboard of a personal computer. While the arrangement of the components on the motherboard is syntactic, this arrangement also has meaning. For example, the location of memory chips within memory banks has meaning, especially to somebody who is about to add additional RAM to their computer.

The follow sections use this distinction between multimedia syntax and semantics to distinguish between pixel and semantic representations in video, and signal and semantic representations in audio.

1.1.1 Representing video: pixels and semantics

Figure 1.1 presents the distinction between pixel and semantic representations of still and full-motion video.

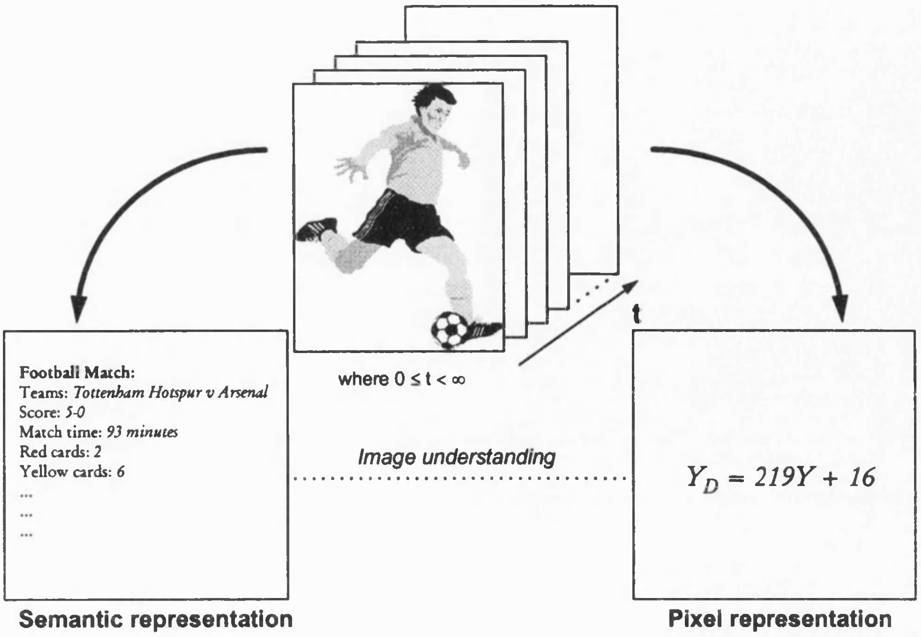


Figure 1.1 Semantic and pixel representations of still and full-motion video.

Pixel representations are concerned with the storage of arrays of values, in which each value represents the data associated with a pixel in the image. For a bitmap this value is a binary digit; for a colour image, the value may be a collection of numbers or an index indicating the intensities of various key colours, e.g. red, green and blue (Steinmetz and Nahrstedt, 1995). Pixel representations for video applications increasingly take advantage of motion-compensated transform coding methods, as in MPEG (Le Gall, 1991; Meyer-Boudnik and Effelsberg, 1995) and H.261 (Liou, 1991). These image representations are described in terms of video frames divided into arbitrary square blocks and, as such, are mathematically intensive.

Intelligent image understanding techniques have sought to move toward more semantic representations of images by attempting to recognise objects within images. However, they have only partly attempted to bridge the gap (represented by a dotted line in the figure). Image understanding is necessarily process-oriented, focusing on three broad stages (Chang and Hsu, 1992): (1) image analysis and pattern recognition;

(2) image structuring and understanding; and (3) spatial reasoning and image information retrieval. This perspective yields a hierarchical structuring of information such as that illustrated in Table 1.1.

Table 1.1 Five-level structuring of information in image understanding systems. Source: Chang and Hsu (1992).

Level	Tasks	Example
User view	Spatial reasoning	Find all motor vehicles with wheels
Semantic feature view	Image knowledge structuring	Find icons, such as (image_object, wheel)
Image feature view	Image understanding	Find icons, such as (image_object, circle)
Feature representation	Image data structuring	Find icons, such as (image_object, contour_of_circle)
Feature organisation	Image data storage/retrieval	Store/access icons (image_object, contour_data_structure)

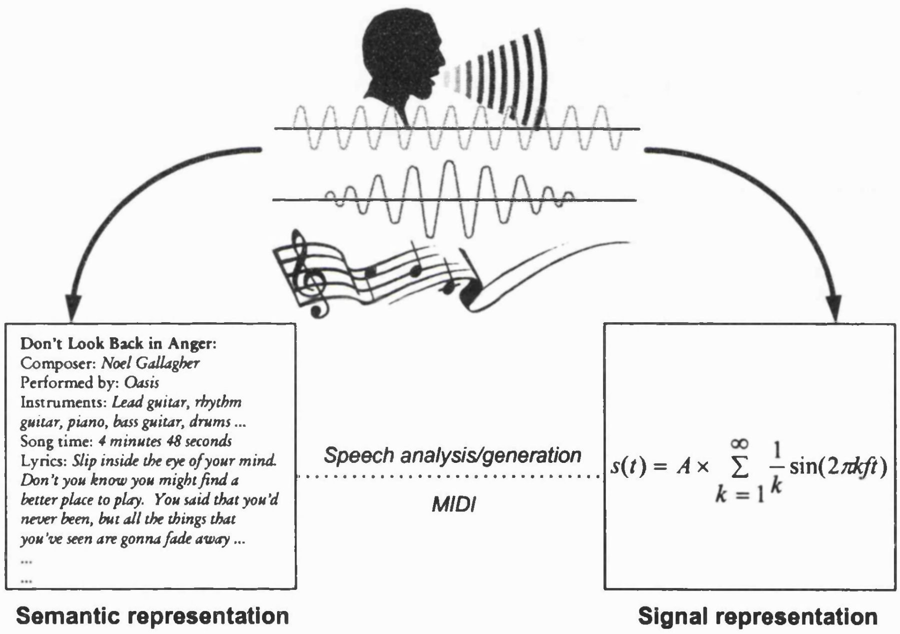
At present, however, image understanding researchers do not completely agree on a common representation for important tasks, e.g. the appropriate decomposition of an object into parts that enable efficient recognition is still a subject of basic research (Mundy, 1995). However, there is an extensive body of well-accepted algorithms and data structures that define the current state of achievement (Chang and Hsu, 1992; Bach et al., 1993; Zhang et al., 1993; Bove et al., 1994; Golshani and Dimitrova, 1994; Pentland et al., 1994; Sakauchi, 1994; Smoliar and Zhang, 1994; Tonomura et al., 1994; Wu and Narasimhalu, 1994; Yoshitaka et al., 1994; Barber et al., 1995; Flickner et al., 1995; Gudivada and Raghavan, 1995; Mundy, 1995; Gong et al., 1996; Kanade, 1996; Wactlar et al., 1996), but these models are not rich enough to capture the information necessary for comprehensive processing and are therefore inadequate for domain- and task-independent image understanding (Gudivada and Raghavan, 1995). Table 1.1 also emphasises the predominance within image understanding of merely identifying objects and not necessarily any further information about the objects. For example, we may be concerned with what a particular motor vehicle is doing within an image: Is it parked? Is it racing? Has it crashed? Which way is it facing?

Full-motion video further complicates the matter. Understanding that relies on the contents of the video frames is a very difficult problem. Current successful efforts at visual querying of image databases fail to capture and exploit the massive information contained in video. Video is temporal, spatial, and often unstructured; the combined video and audio signals convey an abundance of information (Kanade, 1996). While Swanberg et al. (1992) argue that, in many cases, video information is structured in the sense that there exists both a strong spatial order within individual frames and a strong temporal order among different frames pertaining to the same scene, a broader perspective would reveal this to be only true to a limited extent, e.g. in scenes from a news programme. Furthermore, it is the temporal nature of video that brings to the fore issues concerning what particular objects are doing within the video. For example, suppose we want to determine all those frames in which a specified object performs a particular act, such as video frames in which a white horse is galloping. Whereas recognising the white horse is relatively easy, selecting frames in which the horse is galloping (and not jumping or cantering) is extremely difficult.

Incorporation and further development of image processing techniques based on the motion and similarity between pictures are steps towards a possible solution (Golshani and Dimitrova, 1994). However, in dealing with images and video, equality and matching are special cases of similarity. Yet, image understanding systems replace the notion of 'equality' with 'similarity', whereas in mathematics and in traditional databases, equality and matching are dominant notions used at every stage. Thus, techniques to define similarity appropriately and to organise data based on the notion of similarity do not yet exist (Jain, 1996).

1.1.2 Representing audio: signals and semantics

Figure 1.2 presents the distinction between signal and semantic representations of audio. Audio is often put to a variety of uses, including speech, music and sound effects, and the figure illustrates these many facets.



Don't Look Back in Anger:
Composer: *Noel Gallagher*
Performed by: *Oasis*
Instruments: *Lead guitar, rhythm guitar, piano, bass guitar, drums ...*
Song time: *4 minutes 48 seconds*
Lyrics: *Slip inside the eye of your mind. Don't you know you might find a better place to play. You said that you'd never been, but all the things that you've seen are gonna fade away ...*
...
...

$$s(t) = A \times \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k f t)$$

Figure 1.2 Semantic and signal representations of audio.

Whatever audio is used for, however, the signal representations are always concerned with the storage of digital samples. These are discrete numbers representing the amplitude of the analogue sound waveform at regular time intervals. The greater the number of bits used to approximate the height of the waveform, the closer the resultant waveform - reconstructed from the stream of discrete numbers - will be to the original analogue waveform. For example, if eight bits are used in sampling the amplitude, then the amplitude may take on one of 256 possible values at each interval. With fewer bits, however, less possible values are available, and so the shape of the digitally reconstructed waveform will become less discernible, resulting in lower quality sound (Steinmetz and Nahrstedt, 1995).

Intelligent speech analysis and speech generation techniques have both sought to move toward more semantic representations of speech. The former by attempting to recognise who is speaking, what is being said (i.e. what words), or how something is being said within digital audio (e.g. angrily), thus moving from signals to semantics; the latter by attempting to transform text into speech, thus moving from semantics to signals. However, like intelligent image understanding techniques, they have only partially bridged the gap (represented by a dotted line in the figure). Similarly, the use of digitised music has led to more symbolic forms of music representation, the most popular being the MIDI (Music Instrument Digital Interface) data format included in the standard. More bespoke approaches to music include encoding the sheet music into a digital representation (Rader, 1996).

Speech analysis, like image analysis, is necessarily process-oriented, focusing on three broad stages (Steinmetz and Nahrstedt, 1995): (1) acoustic and phonetic analysis; (2) syntactical analysis (speech recognition); and (3) semantic analysis (speech understanding). In contrast, speech generation uses one or more of the following techniques (Steinmetz and Nahrstedt, 1995): pre-recorded speech samples; time-dependent speech concatenation; or frequency-dependent sound concatenation. With the latter two, the process focuses on translating text into a sound script which is then translated into a speech signal.

As with the field of image understanding, there is a body of well-accepted algorithms and data structures for speech analysis and generation that define the current (rather modest) state of achievement (Brand, 1988; Waibel, 1988; Riley, 1989; Bonarini, 1993; Koons et al., 1993; Rich et al., 1994; Strawn, 1994; Alonso et al., 1995; Edwards and Blore, 1995; Vicsi, 1995; Hemphill et al., 1996; Manaris and Slator, 1996; Martin et al., 1996; Moore and Mittal, 1996; Paris and Linden, 1996; Sheremetyeva and Nirenburg, 1996; Waibel, 1996; Wermter and Weber, 1996), but these are not extensive and, again, tend to be domain-specific and task-dependent. They also are predominantly speaker-dependent.

Kanade (1996) explains that audio is also intrinsically linked to video. The audio signal includes language information in the form of narration and dialogue that, when transcribed, provide direct indices to the video content. Natural language analysis of the transcript, together with production notes and other text information about the video, can determine the narrative's subject area and theme. This understanding can be used to generate summaries of each video segment for icon labelling, browsing, and indexing. The audio signal conveys other information, including pauses, silence, music, and laughter. These bits of information can supplement the other structured descriptors, e.g. pauses might be useful in identifying natural start and stop positions for video segmentation.

1.2 REVIEW OF RESEARCH ON SEMANTIC CONTENT-BASED MODELLING

Efforts to introduce semantics into video and audio have centred around the development of models and architectures that seek to capture content-based information to complement the video and audio stream. These models may be seen to fit within one of four groups, according to the technique they employ: (1) those modelling 'physical' (i.e. syntactic) content information, such as colour, texture, and camera motion; (2) those concerned with representing the spatial and temporal location of content objects; (3) stratification-based techniques; and (4) formal techniques.

This section reviews relevant research work within these areas. Because the distinctions may not always be 'clear cut', work is categorised according to how the majority of the model fits into the categories. Moreover, because this thesis is specifically concerned with the modelling of video and audio information - and not other forms of multimedia information, such as graphics and animation - the discussion will centre around those models that specifically take into account video and audio information.

1.2.1 Physical models

These models are primarily concerned with 'physical' content information, which is typically syntactic in nature, e.g. colour, texture, and camera motion.

The 'NTT' model

At the NTT Human Interface Laboratories, Japan, Tonomura et al. (1994) developed methods for video parsing where each shot (a logical video segment) is then further analysed to obtain features of the video content, called *video indexes*. The indexes are organised into two kinds of structures: the *link structure* describes the link relations between shots, and the *content structure* stores information about the scene and objects as obtained by shot analysis. Camera work information suggests the scene's spatial situation, while representative colour information provides some information about the objects. Techniques are discussed to automatically extract this information.

Query by Image Content

The data model used in IBM's Query by Image Content (QBIC) system (Barber et al., 1995; Flickner et al., 1995) stores still images or video scenes that contain objects (subsets of an image), and video shots that consist of sets of contiguous frames and contain motion objects. This data model is used for both database population (where images and videos are processed to extract and store features describing their content) and querying (where the user composes a query graphically). The content used in both cases includes the colour, texture, shape, sketch, and location of image objects and regions. For video, content includes object and camera motion.

1.2.2 Techniques for locating content objects

Models within this category are focused on identifying the spatial and temporal location of content objects, often for enabling user interaction with the video and audio.

Visual Repair

Visual Repair (Goodman, 1993) is a prototype explanation generation component for an intelligent multimedia training system in the domain of Apple Macintosh IIcx repair. Video is used in the student's repair plans, to illustrate to the student what he has advised should be done to fix the fault, and when giving help at the student's request. Relevant parts of the video are graphically highlighted as it is played. The beginning frame of each video has information associated with it about the content of that video frame (e.g. as shown in Figure 1.3, the name of important elements and the size and location of each element). That information can be used to automatically generate graphics that are superimposed on the video in order to point out the recipient objects of important actions during the execution of the presentation plan.

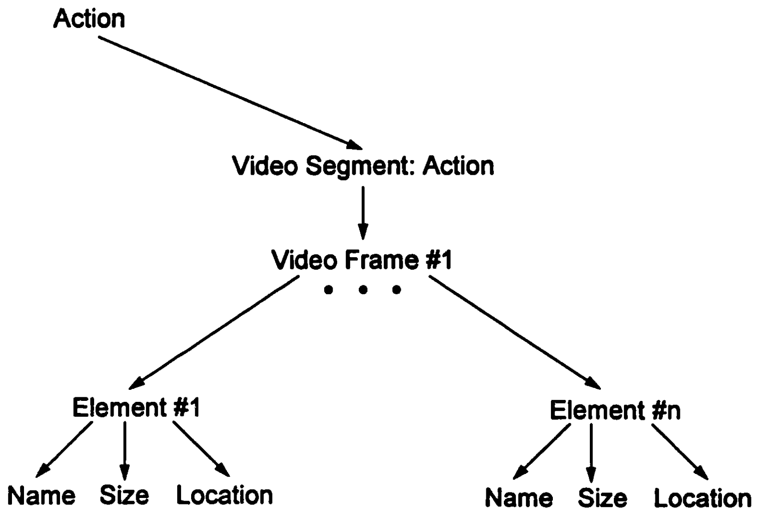


Figure 1.3 How video frames are described in Visual Repair. Source: Goodman (1993).

Sensitive Regions

Burrill et al. (1994) propose the use of Sensitive Regions (or 'hot-spots'), which use pre-editing to define regions of interest within video frames. The regions are identified through the use of polyhedral 3D volumes, on the representational axes 'width', 'height', and 'time'. In specific implementations, the authors suggest that the model can be extended to attach application-dependent semantics to the objects delineated within these regions, but they do not discuss this any further.

In its simplest form, the approach can be used as a trigger mechanism which enables the user to click within the hot-spot, e.g. actors, stage 'props' and scenery, to identify the object or invoke some hyperlink to another part of the underlying hyperbase.

The concept of Sensitive Regions could also be used for non-visual objects such as background music and film mood.

The timeline-tree model

Hirzalla et al. (1995) use an enhanced timeline (a timeline is a graph representing the flow of media over time) as a theoretical model for interactive multimedia. The enhanced timeline has six basic units (Figure 1.4). The edges of these units, which represent start or end events, are either straight (representing synchronous events) or bent (representing asynchronous events). Maximum start and end times are used to ensure that a multimedia presentation is kept moving.

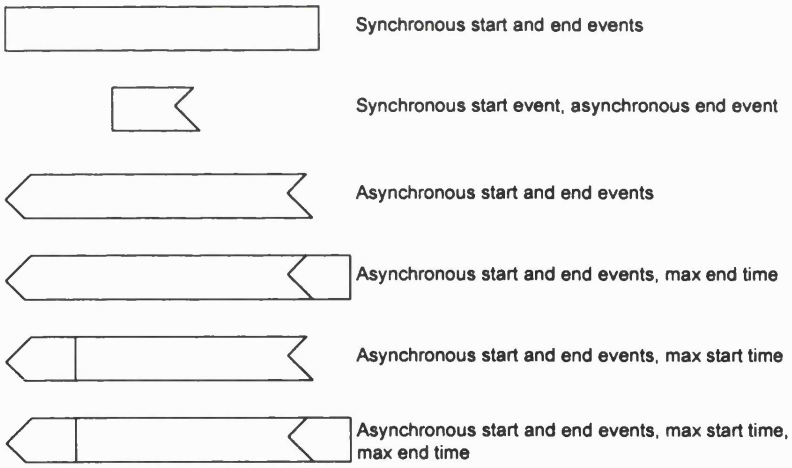

Synchronous start and end events
Synchronous start event, asynchronous end event
Asynchronous start and end events
Asynchronous start and end events, max end time
Asynchronous start and end events, max start time
Asynchronous start and end events, max start time, max end time

Figure 1.4 The six types of units in the timeline-tree model. Source: Hirzalla et al. (1995).

To permit interactive multimedia, the authors also introduce a symbol, Choice$_i$ (C$_i$). Because each user choice results in a different timeline, *i* refers to timeline$_i$, where $i \geq 0$. Thus, there are many different timelines, so timeline$_i$ is a timeline that branches from timeline$_j$, where $j < i$. C$_i$ also helps to distinguish between temporal equalities and inequalities with other asynchronous events. Events that share temporal equalities (that is, that do not admit terms such as 'at least' or 'at most') carry the same symbol; otherwise the symbols differ.

A data structure that determines the user action that initiates the object is associated with C_i. It contains several fields:

- **user_action**, which describes what input should be expected from the user, such as 'keypress-y' or 'left-mouse'.
- **region**, which establishes which region of the screen (if applicable) is a part of the action, e.g. 'rectangle(100, 100, 150, 180)'.
- **destination_scenario_pointer**, which names a pointer to some other part of the scenario, or even a different scenario.

Figure 1.5a shows an example car demonstration scenario, where a user is presented with a graphic of a car. Embedded onto the presentation screen (i.e. modelled in the scenario as a combination of a user action (mouse click) and a region on the screen) are three hot-spots: the hood, the door, and the background. The user can either choose one of three options by clicking on one of the hot-spots or opt not to make a choice at all. Each choice triggers either text explaining the features of the car's engine, a video-audio clip showing and explaining the interior of the car, or the disappearance of the car, respectively. If the user does not respond within a certain time frame, the car image disappears. If the user chooses the hood and gets the text object, he might then choose to listen to the engine by making another interactive selection from the playback area. The presentation ends after the audio plays. Since the end events of 'Text' and 'Audio$_2$' objects have temporal equality, both are labelled with C_4.

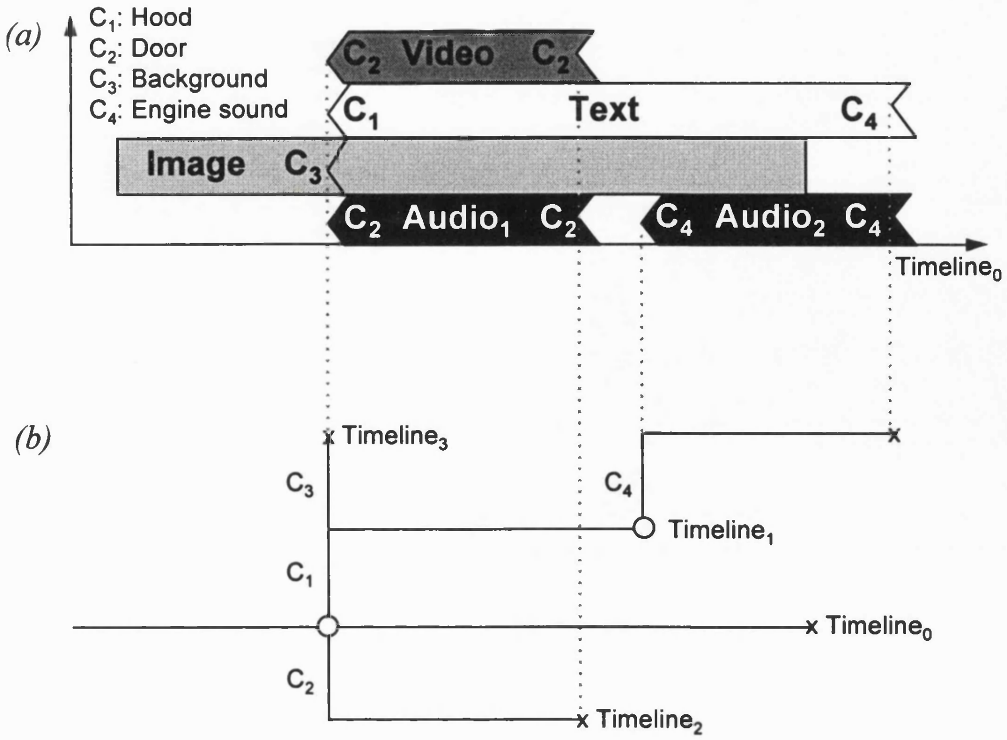


Figure 1.5 The timeline-tree model represents interactive scenarios like this car demonstration using: **(a)** an expanded timeline; and **(b)** a tree-like structure that traces all possible timelines. Source: Hirzalla et al. (1995).

Figure 1.5b shows the tree corresponding to the interactive scenario in Figure 1.5a. The small circles represent branches where user actions may change the course of the scenario. If the user makes no choices, the current timeline simply plays itself out (timeline$_0$); otherwise, users traverse the timeline tree, viewing custom presentations (timeline$_1$ through timeline$_4$) determined by their choices. In the figure, each 'x' represents possible ending points of the scenario.

At most one choice, C_i, can be selected at a time. Consequently, the presentation flow will branch to timeline$_i$. In the tree model, the circles represent the times that asynchronous events corresponding to the symbols at the circle become activated. Those events are deactivated only when the presentation flow branches to another timeline.

IntelligentPad

The IntelligentPad architecture (Tanaka, 1996) is based on pads, each of which consists of a display object, which defines both its view on the display screen and its reaction to user events, and a model object, which defines its internal state and behaviour (Figure 1.6).

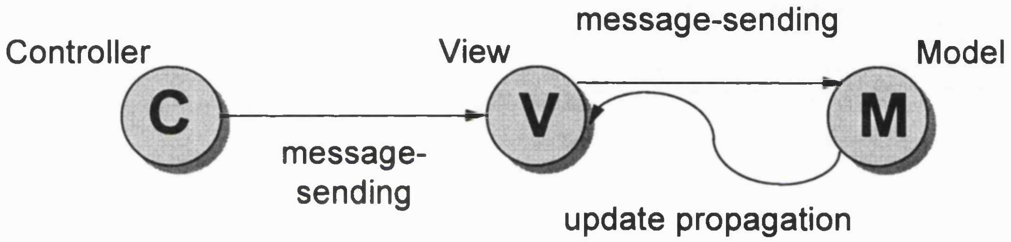


Figure 1.6 The internal structure of each pad in the IntelligentPad architecture. Source: Tanaka (1996).

Pads may be used to represent container objects (container media that carry content information), media objects (container objects with their content objects), and reference frames (which indirectly specify the corresponding sub-portion of content, with time segments working as temporal reference frames and rectangular areas working as spatial reference frames). For the access of non-articulated (that is, non-machine recognisable) content objects, i.e. those in images, movies and sounds, in a media object, the media object can be provided with a special slot named 'reference_frame' that receives the location and size of a reference frame and returns the corresponding portion of its content information. Spatial reference frames can be represented as transparent pads that minimally cover the target content objects.

1.2.3 Stratification-based techniques

These models assign strata to contiguous segments of video and audio which provide descriptions of the content of the segment. The detail and makeup of such descriptions varies considerably between the models.

CLORIS

Parkes (1988) proposes a model for handling descriptive data for video information that is used in the CLORIS intelligent multimedia tutoring system. The model has two basic concepts: *events* and *settings*. An event is a hierarchical description of a video scene based on PART-OF relationships. For instance, suppose a video scene A shows how to use a micro-meter. The event 'USING THE MICRO METER' is assigned to A. This is the root of the description. The event 'USING THE MICRO METER' consists of four sub-events, that is, 'REMOVE MICRO FROM CASE', 'CLEAN MICRO', 'MEASURE METAL', and 'RECORD MEASURE'. Each event corresponds to some portion of video A. The event 'CLEAN MICRO' itself consists of four further sub-events, 'HOLD MICRO', 'LIFT CLOTH', 'WIPE ROD', and 'REPLACE CLOTH'. These may each consist of further sub-events. A setting corresponds to different representations of the same object in the real world. For instance, the binary relations 'zoom in' and 'zoom out' are defined between these settings.

Experimental Video Annotator (EVA)

EVA (Mackay, 1989; Mackay and Davenport, 1989) is a video annotator system, written in Athena Muse and developed at MIT. It provides software researchers with the facilities to create labels and annotation symbols prior to a session and then permits live annotation of video during an experiment. Although EVA is a useful tool for analysing (particularly live) video data, the capability to share descriptive information among annotated video scenes is relatively weak. It is not fully addressed what operations are needed to compose/decompose the annotated video scenes.

The Stratification System

The Stratification System (Aguierre Smith and Davenport, 1992) is a video annotation system that uses the concept of stratification to assign descriptions to video footage, where each stratum refers to a sequence of video frames. The strata may overlap or totally encompass each other. Figure 1.7 shows an example of video footage annotated by strata. Strata are stored in files accessible by a simple keyword search. A user can find a sequence of interest, but cannot easily determine the context in which it appears because of the absence of relationships between the strata.

The video object data model

Oomoto and Tanaka (1993) propose the video object data model as a new modelling construct for video database management. They consider that any portion of a video frame sequence is an independent entity, and so make it possible to define a *video object*, which corresponds to a certain set of video frame sequences. It has its own attribute-value pairs to represent the content (meanings) of the corresponding video scene.

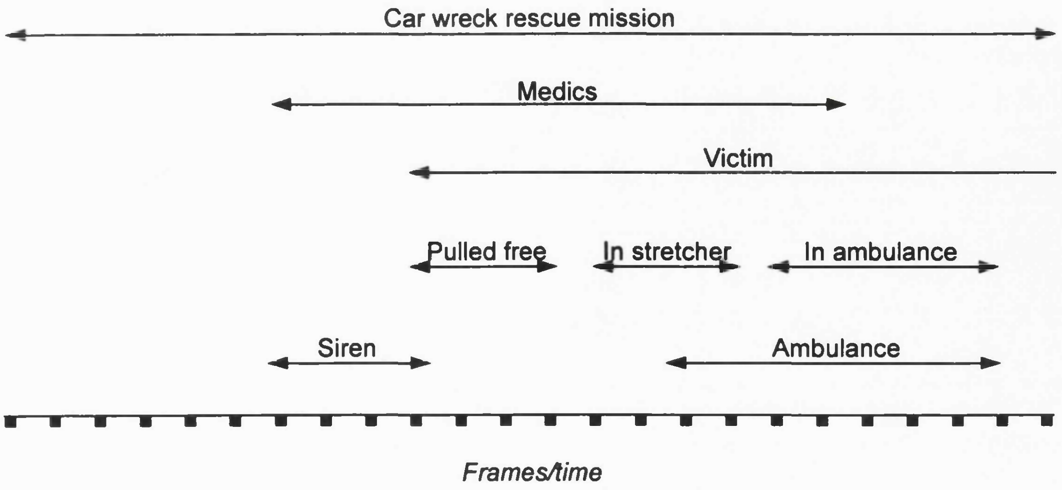


Figure 1.7 Example of strata in the Stratification System.

Figure 1.8 shows an example video object database. The main features of the video object data model are:

- **Schemaless description of database,** i.e. there is no assumption of a specific database schema such as classes and a class hierarchy, so users can define any attribute's structure for each video object.
- **Interval inclusion inheritance,** whereby some descriptive data of video objects can be inherited by other video objects. For example, in Figure 1.8, object 3 (O_3) has attribute 'Location' and its value 'America'; thus O_4 to O_7 also have this attribute-value by the interval inclusion relationship.
- **Composition of video objects based on an IS-A hierarchy.** The authors define several operations, *interval projection*, *merge* and *overlap*, for video objects that compose new video objects. These operations also derive, based on the IS-A hierarchy, the attribute-values of the synthesised video object from the original video objects.

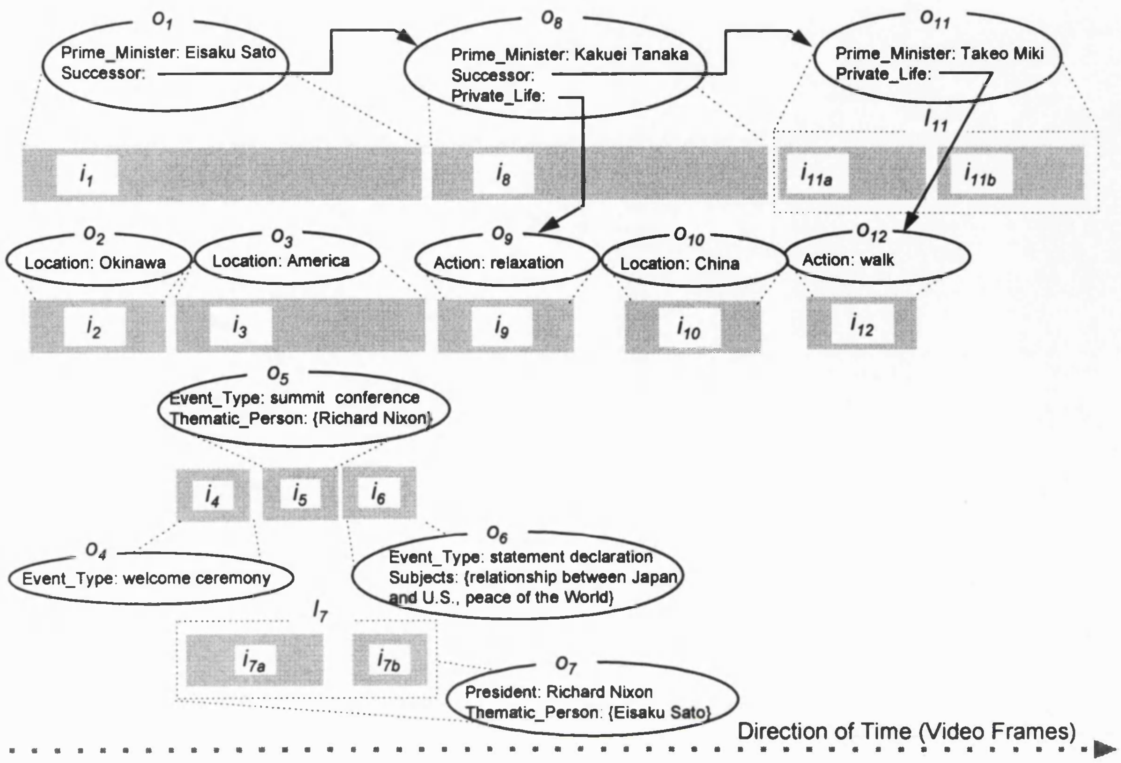

o_1
Prime_Minister: Eisaku Sato
Successor:
o_8
Prime_Minister: Kakuei Tanaka
Successor:
Private_Life:
o_{11}
Prime_Minister: Takeo Miki
Private_Life:
I_{11}
i_1
i_8
i_{11a}
i_{11b}
o_2
Location: Okinawa
o_3
Location: America
o_9
Action: relaxation
o_{10}
Location: China
o_{12}
Action: walk
i_2
i_3
i_9
i_{10}
i_{12}
o_5
Event_Type: summit conference
Thematic_Person: {Richard Nixon}
i_4
i_5
i_6
o_4
Event_Type: welcome ceremony
o_6
Event_Type: statement declaration
Subjects: {relationship between Japan and U.S., peace of the World}
I_7
i_{7a}
i_{7b}
o_7
President: Richard Nixon
Thematic_Person: {Eisaku Sato}
Direction of Time (Video Frames)

Figure 1.8 Example video object database using the video object data model. Source: Adapted from Oomoto and Tanaka (1993).

Media Streams

Media Streams (Davis, 1993) is an iconic visual language that enables users to create multi-layered, iconic annotations of video content. Icons denoting objects and actions are organised into cascading hierarchies of increasing levels of specificity. Additionally, icons are organised across multiple axes of descriptions such as objects, characters, relative positions, time, or transitions. The icons are used to annotate video streams represented in a timeline. Currently, around 2,200 iconic primitives can be browsed. However, this user-friendly visual approach to annotation is limited by a fixed vocabulary. Also, it does not exploit textual data such as closed-captioned text.

The Virtual Video Browser

Little et al. (1993; 1995) propose a system that supports content-based retrieval of video footage. They define a specific data scheme composed of Movie, Scene, and Actor relations with a fixed set of attributes. The system requires manual feature extraction, then fits these features into the data scheme. It permits queries on the attributes of movie, scene, and actor. Having selected a movie or a scene, a user can scan from scene to scene. To achieve this, the model uses an object composition Petri net (OCPN) to represent the interconnections of the various scenes, based on the earlier work of Little and Ghafoor (Little and Ghafoor, 1990; Little and Ghafoor, 1991; Little and Ghafoor, 1993; Little, 1994). An OCPN uses the structure of a Petri net to maintain synchronisation between the various elements (in this case, scenes) in a multimedia presentation (in this case, a movie). Unfortunately, the data model and the virtual video browser are limited because descriptions cannot be assigned to overlapping or nested video sequences as in the Stratification System. Moreover, the system is focused on retrieving previously stored information and is not suitable for users who need to create, edit, and annotate a customised view of the video footage.

The algebraic video data model

The algebraic video data model (Weiss et al., 1995) consists of hierarchical compositions of video expressions with high-level semantic descriptions, constructed using video algebra operations. Video algebra is used as a means of combining and expressing temporal relations, defining the output characteristics of video expressions, and associating descriptive information with these expressions. Interaction with algebraic video is accomplished through four activities: Edit and Compose, Play and Browse, Navigate, and Query. The operations that support playback, navigation, and content-based queries are grouped together as interface operations.

The fundamental entity of the model is a *presentation*, a multi-window spatial, temporal, and content combination of video segments. Presentations are described by video expressions. The most primitive video expression creates a single-window presentation from a raw video segment. These segments are specified using the name of the raw video and a range within it (Figure 1.9).

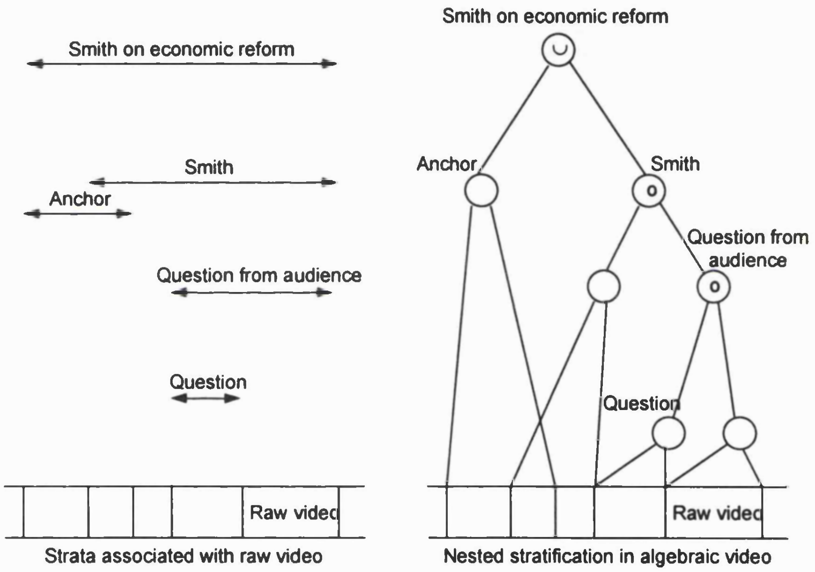


Figure 1.9 Nested stratification in the video algebra data model. Source: Weiss et al. (1995).

Compound video expressions are constructed from simpler ones using video algebra operations. Video expressions can be named by variables, can be composed to reflect the complex logical structure of the presentations, and can share the same video data. A video expression may contain composition information, descriptive information about the contents, and output characteristics that describe the playback behaviour of the presentation.

An algebraic video node provides a means of abstraction by which video expressions can be named, stored, and manipulated as units. It contains a single video expression that may refer to children nodes or raw video segments.

Video algebra operations fall into four categories. First, *creation* defines the construction of video expressions from raw video. Second, *composition* defines temporal relationships between component video expressions. The composition operations can be combined to produce complex scheduling definitions and constraints. As Table 1.2 shows, these operations make up, by far, the largest proportion of the model's operations. Third, *output* defines spatial layout and audio output for component video expressions.

Finally, *description* associates content attributes with a video expression. Of all the four, the description operations of the algebraic video data model are the most pertinent to this thesis. The model permits the association of arbitrary descriptions with a given video algebra expression. It allows textual descriptions, non-textual descriptions like key frames, icons, and salient still images, and image features like colour, texture, and shape. The Description operation associates content information with a video expression. The Content description of an expression is not fixed by the model, but a Content may be considered to be a Boolean combination of attributes that consists of a field name and a value, e.g. *title* = "*Smith on economic reform*". Some field names have pre-defined semantics - for example, *title* - while other fields are user-definable. Values can assume a variety of types, including strings and video node names. Field names or values do not have to be unique within a description. Therefore, a description can have multiple titles, text summaries, and actor names associated with a video expression. The components of a video expression inherit descriptions by context, such that all the content attributes associated with some parent video node are also associated with all its descendant nodes. The Hide-content operation defines a video expression *E* that does

Table 1.2 Video algebra operations. Source: Weiss et al. (1995).

	Usage	Function
Creation		
Create	`create` *name begin end*	Creates a presentation from the range within the identified raw video segment
Delay	`create` *time*	Creates a presentation with empty footage for duration *time*
Composition		
Concatenation	$E_1 \circ E_2$	Defines the presentation where E_2 follows E_1
Union	$E_1 \cup E_2$	Defines the presentation where E_2 follows E_1 and common footage is not repeated
Intersection	$E_1 \cap E_2$	Defines the presentation where only common footage of E_1 and E_2 is played
Difference	$E_1 - E_2$	Defines the presentation where only footage of E_1 that is not in E_2 is played
Parallel	$E_1 \parallel E_2$	Defines the presentation where E_1 and E_2 are played concurrently and start simultaneously
Parallel-end	$E_1 \wr\wr E_2$	Defines the presentation where E_1 and E_2 are played concurrently and terminate simultaneously
Conditional	`(test)` $? E_1 : E_2 : ... : E_k$	Defines the presentation where E_i is played if `test` evaluates to *i*
Loop	`loop` E_1 *time*	Defines a repetition of video expression E_1 for a duration of *time* (can be *forever*)
Stretch	`stretch` E_1 *factor*	Sets the duration of the presentation equal to *factor* times duration of E_1 by changing the playback speed of the video expression
Limit	`limit` E_1 *time*	Sets the duration of the presentation equal to the minimum of *time* and the duration of E_1, but the playback speed is not changed
Transition	`transition` E_1 E_2 *type time*	Defines *type* transition effect between expressions E_1 and E_2; *time* defines the duration of the transition effect
Contains	`contains` E_1 *query*	Defines the presentation that contains component expressions of E_1 that match *query*
Output		
Window	`window` E_1 (x_1, y_1) - (x_1, y_2) *priority*	Specifies that E_1 will be displayed with *priority* in the window defined by the bottom-left corner (x_1, y_1) and the right-top corner (x_2, y_2) such that $x_i \in [0, 1]$ and $y_i \in [0, 1]$
Audio	`audio` E_1 *channel force priority*	Specifies that the audio of E_1 will be output to *channel* with *priority*; if *force* is true, command overrides `audio` specifications of the component expressions
Description		
Description	`description` E_1 *content*	Specifies that E_1 is described by *content*
Hide-content	`hide-content` E_1	Defines a presentation that hides the content of E_1

not contain any descriptions, and provides a method for creating abstraction barriers for content-based access. Figure 1.9 showed an example of description added to the raw video.

Table 1.2 presents the video algebra operations. The arguments denoted by E_1, E_2, ..., E_k are video expressions. The result of each is a presentation. The operations are inherently not commutative because they include a temporal component.

The Matilda information representation model

The Matilda information representation model (Lowe, 1995) separates the information domain (which contains application-independent information) from the application domain (which contains application-dependent information). Figure 1.10 shows the model. Using the model, databases may be populated with multimedia and standard information and then used for various applications. For example, a multimedia application might add structuring information onto the media information contained in the database, while a video archive might layer reference data on the media information.

The Advanced Video Information System (AVIS) model

Adah et al. (1996) present a content-based model for video data that has been implemented within a prototype system, AVIS. The model represents three main types of entities within the video:

- **Video objects** are present in video frames and include characters and objects that are present in a movie. 'Invisible' objects may also be modelled. It is therefore possible to represent the fact that some object X is present inside a cupboard (which is visible) even though X cannot be physically seen.

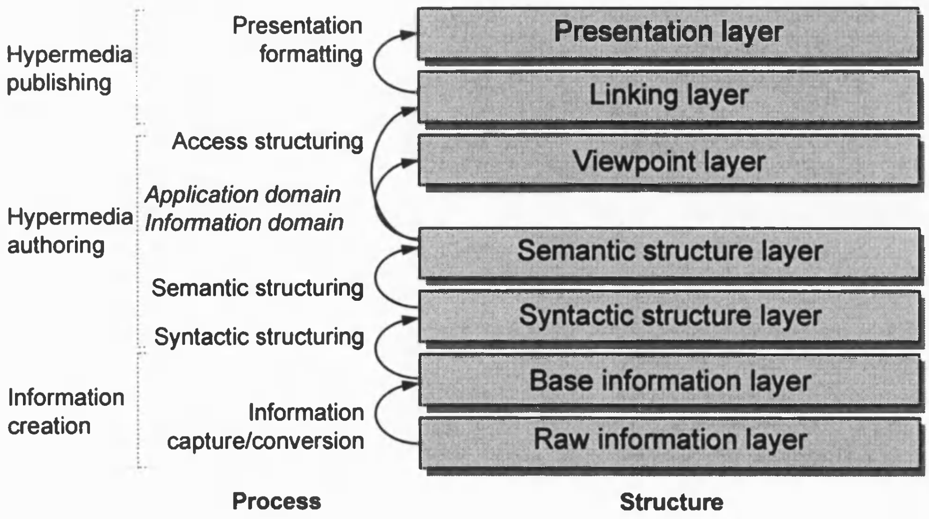


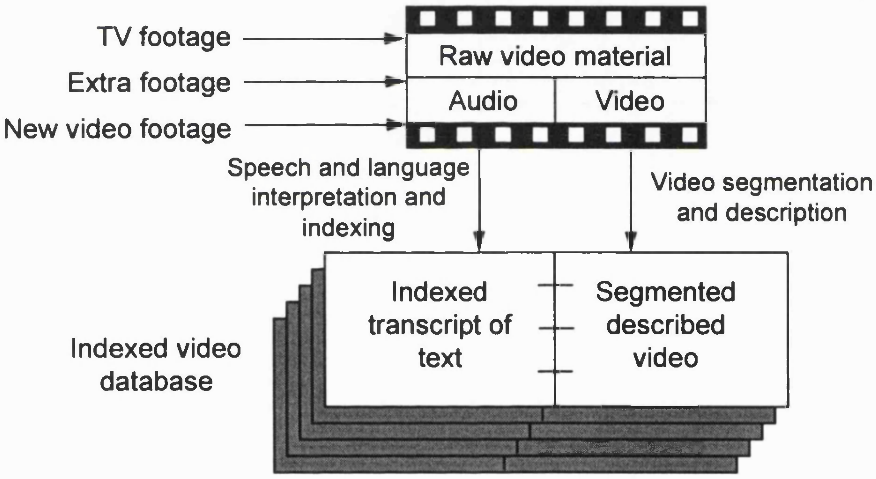
Figure 1.10 The Matilda information representation model. Source: Lowe (1995).

- **Activity types** describe the (generic) subject of a given video frame sequence, such as 'murder' or 'giving a party'. Multiple activities may occur simultaneously.
- **Events** are instantiations of an activity type which make the activity more specific. Activity types are therefore general groups containing many events. Two further sub-entities that are used to construct events help distinguish events from activity types:
  (a) **Roles** are descriptions of certain aspects of an activity. They may involve objects (e.g. 'victim' and 'murderer' are roles in the activity 'murder') and descriptions (e.g. 'murder motive' and 'murder weapon').
  (b) **Teams** are sets of roles (objects/descriptions) that jointly describe an event; that is, they are instantiations of the roles in an activity type. For example, for the event 'murder', the team involved might consist of Tom in the role 'victim', and Dick and Harry both in the role 'murderer'. A gun may play the role of the 'murder weapon', while 'mugging' is the role 'murder motive'. Members of a team are called players.

These entities are represented using association maps and a specially adapted form of segment trees, which the authors refer to as frame segment trees.

The Informedia Digital Video Library

Informedia (Christel et al., 1995; Kanade, 1996; Wactlar et al., 1996) is a digital video library system that uses integrated image, speech, and language understanding for the creation and exploration of the library. Figure 1.11 provides an overview of Informedia's off-line creation facilities.



TV footage
Extra footage
New video footage
Raw video material
Audio
Video
Speech and language interpretation and indexing
Video segmentation and description
Indexed video database
Indexed transcript of text
Segmented described video

Figure 1.11 The creation aspects of the Informedia Digital Video Library system. Source: Adapted from Kanade (1996).

Using speech recognition techniques, Informedia converts each videotape's sound track to a textual transcript. A language understanding system analyses and organises the transcript, then stores it in a full-text information retrieval system. Image understanding techniques segment video sequences, detect and identify objects (human faces and text), obtain a visual characterisation of the scene, identify the representative images for the skim video (comprising the significant words and images of the original video), and match images by incorporating language and speech information. Thus, for a particular video clip, Informedia stores information about the following: when scenes change, the different forms of camera motion within the clip (e.g. pan, static, zoom), the location of identified faces, the location of identified text, the word relevance, and the audio level. These are used later for interactive retrieval by a user of the indexed video library.

Jabber

Jabber (Kazman et al., 1996) uses content-based indexing of an audio stream to access the parallel streams produced by video conferences. It performs speech recognition on the audio stream, then groups the recognised words into semantically-linked trees. Jabber uses four forms of indexing (which may be combined):

- **Indexing by intended content**, where meetings are indexed according to an explicit agenda that accompanies the meeting. This agenda is used by users, in real time, to annotate the data streams to indicate the current topic or other aspects of the meeting's structure.
- **Indexing by actual content**, where meetings are indexed by what was said or done, rather than what was planned. A speech recognition system is applied to the stored audio track to create text-based records of the meeting. Clusters of related words (which in turn relate to topics) are identified and used as indexes back into the original audio/video streams.
- **Indexing by temporal structure**, where meetings are indexed by their structure in terms of human interactions over time.

- **Indexing by application record.** A log of a computer application's activity can be kept and used as an index back into the audio/video streams, e.g. object creations, deletions, modifications, changing focus, grouping, and undoing.

1.2.4 Formal techniques

Models within this category use formal techniques, usually based on mathematics, in order to specify the content information.

The Video Classification project

The Video Classification project at the Institute of Systems Science, National University of Singapore (Smoliar and Zhang, 1994), has developed an architecture that characterises the tasks of managing video content (Figure 1.12). It assumes that video and audio information (compressed wherever possible) will be maintained in a database. The database management system (DBMS) defines attributes and relations among the audio and video entities in terms of a frame-based knowledge representation. This representation approach, in turn, drives the indexing of entities as they are added to the database. Those entities are initially extracted by the tools that support the parsing task.

The parsing and indexing aspects of the architecture are the most relevant to this thesis. Three tool sets address the *parsing* task: the first segments the video source material into individual camera shots, which then serve as basic units for indexing; the second set identifies different camera techniques in these clips, e.g. panning and tilting, zooming; and the third set applies content models to the identification of context-dependent semantic primitives, e.g. news broadcasts usually provide simple examples of such models because all shots of the anchorperson conform to a spatial layout, and the temporal structure simply alternates between the anchorperson and more detailed footage (possibly including breaks for commercials).

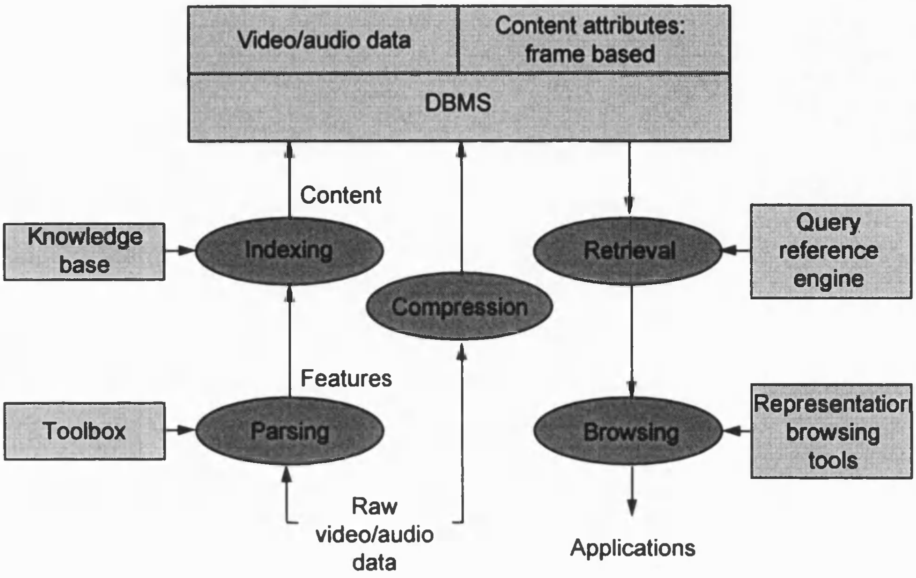


Figure 1.12 The video management architecture of the Video Classification project. Source: Smoliar and Zhang (1994).

Indexing tags video clips when the system inserts them into the database. The tag includes information based on a frame-based knowledge representation model that guides the classification according to the semantic primitives of the images (as opposed to lower level features). Indexing is thus driven by the image itself and any semantic descriptors provided by the model. The various subject matter categories of the material being indexed are represented in a hierarchy as a tree, where each node is a knowledge representation frame. This permits specialisation and generalisation among the categories. For instance, for a documentary video about information systems at the London School of Economics we may have a tree with an 'Information_Systems' frame at its root (to symbolise the entire video). From this root, we may have three categories

'Activity' (which may be further split into 'Academic' and 'Non-academic', to represent all the different information systems activities within the video), 'Person' (to represent the different people in the video), and 'Video_Types' (which may be further split into 'Talking_Heads', 'Animation', 'Demonstration', 'Scenery', and 'Headings', to represent the different kinds of shots that exist in the video). Then a frame for an instance of a laboratory would be something like:

```
Name: Intelligent_Multimedia
Class: Multimedia_Lab
Description: "Applying AI principles to multimedia."
Video: AIMultimedia_CoverFrame
Course: Multimedia_Information_Systems
Equipment: #table[Computer VCR Video_Camera]
```

The Video Classification project is also working on audio and preliminary algorithms have begun to be developed that detect content changes in an audio signal. Plans are to develop models of audio events, similar to the models used in image-based content parsing, e.g. in a sports video, very loud shouting followed by a long whistle might indicate that someone has scored a goal, in which case the system should recognise an 'event'.

The 'Hiroshima' model

At Hiroshima University, Japan, Yoshitaka et al. (1994) developed an object-oriented technique for the composition of domain knowledge in a multimedia database system. In their approach, domain knowledge is held in the database system, which describes how the system views the target multimedia data for content-based retrieval. Domain knowledge, Dk, is a way for a class to present knowledge representing a certain concept held by objects in the class. It is defined as a triple:

$$Dk(C) = \langle Fi\ [fi, extp], Op\ [op\ [fi, mf]\ ], Cm\ [cd, fi, v] \rangle$$

C denotes a concept representing a pseudo object. This pseudo object derives from the objects in a class by providing the domain knowledge. A pair of brackets represents a set.

Fi represents the features constituting a concept C, such as 'colour' and 'length' for the concept 'hair'. A feature item Fi consists of a feature item name fi and a procedure $extp$ to extract the information from objects in the associated class. A feature item is the instance variable (the query attribute) of a pseudo object representing the concept C. For example, if two feature items named colour and length are defined for a piece of domain knowledge whose concept is hair, a pseudo object derived through the domain knowledge has two attributes (pseudo component objects) called colour and length.

Op defines the semantics of operators appearing in a query and how the operator is evaluated during the retrieval. The semantic behaviour of an operator may change depending on the class of objects to be evaluated. For example, the behaviour of an operator '=' for objects in an integer class differs from that in a colour class. A member of Op thus consists of an operator op and a set of descriptions of semantic behaviours corresponding to the operator. That is, the description of a semantic behaviour is given by the combination of a specific item fi and a function mf for evaluating the fitness between the extracted value of feature item fi and a data value v (which is a part of the description of Condition Mapper). Op itself possesses a formalisation function that takes the result of one or more functions and returns an evaluation value that is normalised to take from 0.0 to 1.0. The higher the value, the more the object satisfies the query condition.

Cm converts a condition value *cd* specified in a query into a certain data value (or a certain range of values) *v* whose data type is the same as that of the data values of *fi*. Therefore, both *fi* and *v* are the same type and are processed through *mf*. *v* can be a certain function *f*(*cd*) that returns a certain data value (or a certain range of values).

This approach permits new concepts to be defined by combining pieces of pre-defined domain knowledge in the component classes. Figure 1.13 shows one example. In the figure, a Scene object is composed of a Video_with_annotation object and a Sound object. The Video_with_annotation object is composed of a String object and a Video object. The Video object contains such items as scenery of mountains, a train station, and a main street, and the Sound object is associated with the corresponding Video_with_annotation object. Assuming that there is domain knowledge describing the concepts of mountains, sea, and buildings in the Video class and domain knowledge about waves, birds singing, cars, and trains in the Sound class, then the query "Get the scene objects which include a mountain train", is feasible because the system understands the existence of mountains and trains: the existence of a mountain is derived through the feature item 'existence' in the domain knowledge 'mountains', and the existence of a train is achieved through the feature item 'existence' in the domain knowledge 'trains'.

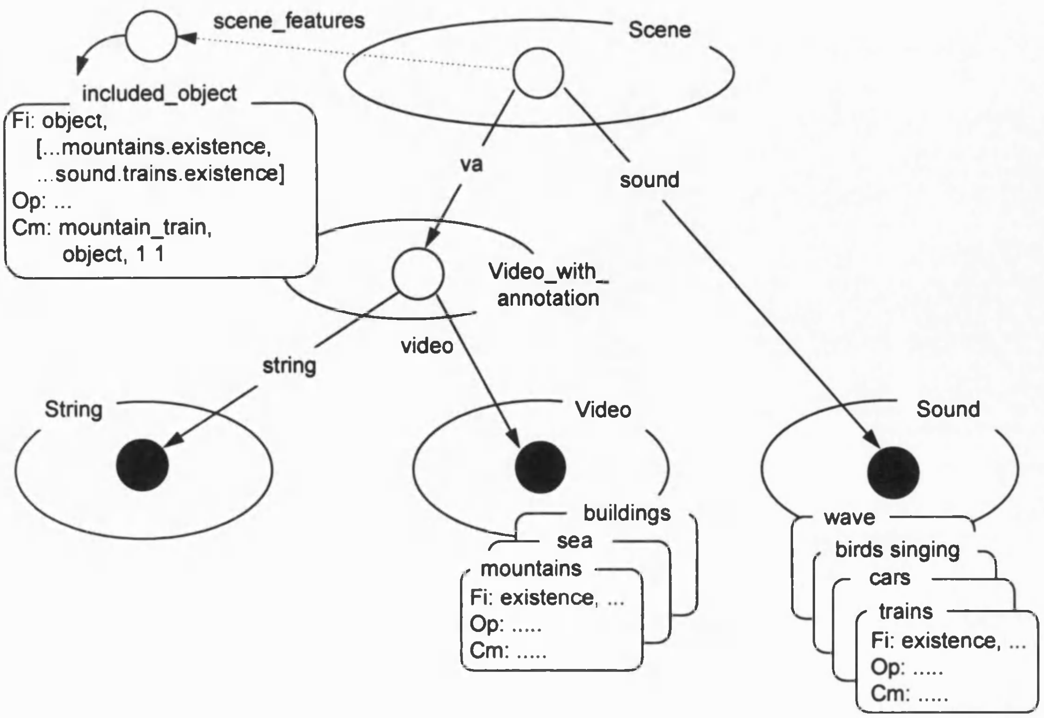


Figure 1.13 The composition of domain knowledge in the 'Hiroshima' model. Source: Yoshitaka et al. (1994).

The media abstraction

Brink et al. (1995) propose the *media abstraction*, expressed as a formal mathematical structure. It captures, as special cases, content-based information. Degrees of certainty, indicating the confidence in object identification, can also be incorporated.

In mathematical terms, a media abstraction is a 7-tuple:

$$\mathrm{M} = (ST, fe, \lambda, R, F, Var_1, Var_2)$$

where ST is a set of objects called states; fe is a set of object features; λ is a map from ST to functions from fe to $[0,1]$; Var_1 is a set of objects called state variables, ranging over states; Var_2 is a set of objects called feature variables, ranging over features; R is a set of fuzzy interstate relations (of possibly different arities – number of arguments) on the set ST; and F is a set of fuzzy feature-state relations. Each relation in F is a map from either fe^i to [0,1] (when relationships between features are independent of state) or $fe^i \times ST$ to [0,1], where $i \leq 1$ (when relationships between features are state dependent).

Thus, a media abstraction called photo would consist of:

- **States.** All files containing a photograph will be separate states in the media abstraction.
- **Features.** These may include persons of interest (e.g. Tony Blair, Gordon Brown) and inanimate features (e.g. Houses of Parliament, 10 Downing Street). Only features of interest are captured in this way; for example, a perfectly recognisable chair in a picture of Tony Blair speaking outside 10 Downing Street – if not of interest – would not be designated as a feature with respect to that picture.
- **Feature map.** This map λ specifies the confidence of a particular feature occurring in a given image. For instance, $(\lambda(s_2))$ (Tony Blair) = 0.7 indicates that the certainty of Tony Blair occurring in state s_2, which may be a picture, is 70 percent.
- **Relations.** There are two types of relations: those that depend on a given state and those that are state independent. For instance, consider a relation, called is_wearing, that has three arguments: a person's name, an item of clothing, and a colour. Since the relation is_wearing changes from state to state – the same person may be dressed differently in two different pictures – this is a state-dependent relation. Hence, an extra, fourth argument, must be added to it: the state name. A sample tuple for this relation, ('Tony Blair', 'tie', 'red', file5) : 0.99, says there is a 99 percent certainty that in the picture contained in file 5, Tony Blair is wearing a red tie. For state-independent relations, there is no need to add an extra state-name argument.

Domains involving audio input can be modelled as follows. *ST* would be the set of all sample acoustic signals. Features are extracted by signal processing and pattern recognition (of phonemes) and can include signal properties such as spectral properties, frequency, and amplitude. These properties, in turn, determine who or what is the originator of the signals, e.g. John Major giving a speech. State variables range over sets of audio signals. Relations in *F* may include feature-based relations such as owns(major, socks, S), which specifies that Socks is owned by Major in all states in the system.

The authors explain that media abstractions are rich enough to capture many other types of media data, including document data and video data, but do not provide similar details of how this may be done.

1.3 INTERROGATING EXISTING RESEARCH

Current research has been preoccupied with a *structure-oriented* approach to semantic content-based multimedia modelling. In other words, the modelling has been organised around the explicit media structure. Such models describe the content of the video and audio stream on a frame-by-frame or segment-by-segment basis, e.g. the Stratification System, where a 'segment' corresponds to an arbitrary sequence of two or more contiguous frames.

The discussion so far has highlighted seven important aspects in semantic content-based modelling (illustrated graphically in Figure 1.14):

1. **Explicit media structure:** The explicit way in which sequences of video and audio are discretely split and grouped together to create flat or hierarchical structures. For example, splitting the video into a number of scenes which each consist of a number

of frames. This structure has implications for the scope and generality of the overlaid semantics since it determines the size of the video and audio units that semantics may be attributed to, e.g. one second, one minute, or one hour. Any semantic content-based model that lacks an explicit media structure is limited to overlaying the semantics onto an entire, often lengthy, media sequence. This may result in semantic information that is too general, too vague, or incomplete.

2. **Objects:** Information about objects active within a video or audio segment, including their location. In the case of video this would typically be those objects that currently appear on the screen, whose location could be determined by pixel co-ordinates. In the case of audio it would typically be those objects that are currently emitting sound, whose location could be determined from 'min:sec' co-ordinates. Without this aspect, semantic content-based models are unable to determine which objects are present and where they are located within given media sequences.

3. **Spatial relationships between objects** that appear on-screen together or are heard together. A content-based model that does not cater for this semantic aspect is unable to support detailed user-led or system-led interrogation, since this information is not always determinable from object co-ordinates. While two-dimensional spatial relationships (e.g. 'is X to the left of Y?') are easily derived using the co-ordinates, deducing three-dimensional spatial relationships (e.g. 'is X diagonally in front of Y?') in this way is a difficult problem. Furthermore, it is all but impossible to determine the spatial relationship between X and Y when X completely obscures Y, as would be the case if X was inside of Y.

4. **Events and actions involving objects:** Information about events and actions taking place within the media and typically involving objects. *Events* are distinguished from *actions* by the fact that they are more general and tend to only implicitly refer to the

objects, whereas actions make it clear that one or more objects are involved. Events frequently consist of many actions and thus often determine the context for their constituent actions. For example, consider a full-motion video sequence depicting a wedding: the corresponding event would be 'wedding', whereas individual actions would be those such as 'gives ring to' and 'kisses'. Without this semantic aspect, a semantic content-based model is unable to determine the intentions and purposes of the objects represented within the media stream, and therefore does not have the 'full picture' of what is taking place.

5. **Temporal relationships between events and actions:** The way in which the sequence and timing of events and actions taking place within the media is determined, e.g. 'A occurs before B', 'C happens while D is happening'. If a semantic content-based model is unable to determine temporal relationships, then the representation of events and actions becomes extremely unstructured, leading to ambiguity within the model. For example, with temporal relationships, if a media stream depicts a fight taking place at a party, we would be able to have two events, 'party' and 'fight', which the model would know took place simultaneously. Without temporal relationships, we would need to have just one event, 'fight during party'. Determining the exact point at which the fight took place during the party (e.g. 5 minutes after the party started?) and how long it lasted for then becomes all but impossible.

6. **Integration of syntactic and semantic information:** Since the content of video and audio is manifested physically through multimedia syntax, multimedia semantics must be tightly integrated with the multimedia syntax. In this way, the video and audio streams together with their associated syntactic and semantic information are able to be used conjointly by the system utilising the model. Omission of this aspect by a

semantic content-based model causes an unnecessary processing burden on the utilising system which must search through all the information to find the required video or audio stream.

7. **Direct user-media interaction:** In other words, a user should be able to directly interact with the media, e.g. by clicking on objects of interest as they are presented to him or her (i.e. as they are delivered through the media stream). This should be in real-time so that the user receives a response from the system within a 'reasonable' amount of time. If the waiting time is too long, the user may be denied the feeling that they are actually interacting with the media. Semantic content-based models that do not support direct user-media interaction must rely on the user entering extraneous information about the objects they wish to interact with, e.g. object names and co-ordinates.

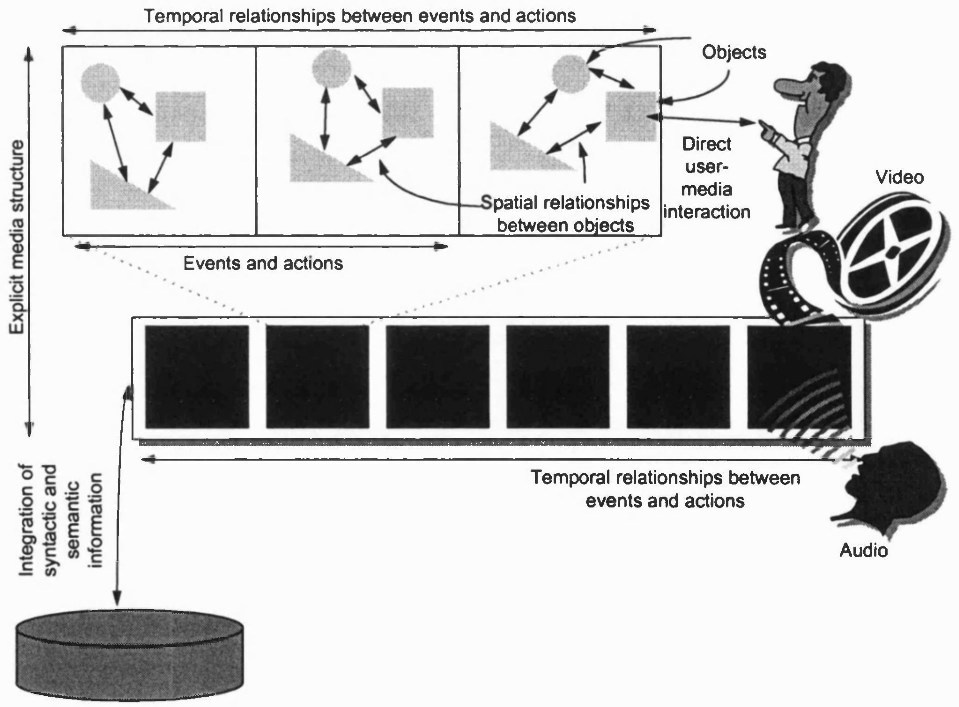

Temporal relationships between events and actions
Objects
Direct user-media interaction
Video
Spatial relationships between objects
Events and actions
Explicit media structure
Integration of syntactic and semantic information
Temporal relationships between events and actions
Audio

Figure 1.14 The seven semantic aspects of video and audio.

Table 1.3 How related work addresses the seven semantic aspects of video and audio.

	Explicit media structure	Objects	Spatial relationships between objects	Events and actions involving objects	Temporal relationships between events and actions	Integration of syntactic and semantic information	Direct user-media interaction
Physical models:							
The 'NTT' model	Frame:scene:video (Video and audio treated in unison)	Via image understanding	N/A	Only camera techniques	Via the link structure	N/A	N/A
QBIC	motion_object: frame:video_shot (Audio unspecified)	Via image understanding	Via image understanding	Only object and camera motion	N/A	N/A	N/A
Techniques for locating content objects:							
Visual Repair	Frames:segments (Audio unspecified)	Expressed as elements	Implicit (via Location in Element)	Limited to overall event/action in the segment	N/A	The Action is part of a plan in the system	N/A
Sensitive Regions	Only width-height-time for the (video) Sensitive Regions	Marked as Sensitive Regions by user	Implicitly via co-ordinates of Sensitive Regions	N/A	N/A	Application-dependent, not specified as part of model	Via clicking on Sensitive Regions, which are specified as 'anchors'

	Explicit media structure	Objects	Spatial relationships between objects	Events and actions involving objects	Temporal relationships between events and actions	Integration of syntactic and semantic information	Direct user-media interaction
Timeline-tree	Arbitrary, user-specified	Via 'region' in C_i's data structure	Implicit	N/A	N/A	N/A	Via C_i
IntelligentPad	frame:movie (video) and seconds:sound (audio)	Via transparent pads	Implicit, via location of pads	N/A	N/A	A standardised interface is provided	When pads specify hot-spots
Stratification-based techniques:							
CLORIS	Arbitrary, video is split into events (Audio unspecified)	Via events	N/A	Via events	Via sequencing of (sub-)events	N/A	N/A
EVA	Arbitrary (Video and audio treated in unison)	Via user-defined annotators	N/A	Via user-defined annotations	N/A	N/A	N/A
The Stratification System	Frame:stratum: video (Standalone audio unsupported)	Embedded in strata's descriptions	N/A	Embedded in strata's descriptions	N/A (no relationship between strata)	N/A	N/A

	Explicit media structure	Objects	Spatial relationships between objects	Events and actions involving objects	Temporal relationships between events and actions	Integration of syntactic and semantic information	Direct user-media interaction
Video object data model	frame:video_scene:video (Standalone audio unspecified – video and audio treated as one medium)	Specified as attribute-values of the video object	N/A	Specified as attribute-values of the video object	Via IS-A video object hierarchy	Model based on set theory, thus easily integrated into a DBMS	N/A
Media Streams	Standard video timeline (Only video supported)	Specified with annotations (using iconic primitives)	Specified by relative placement on axes	Specified with annotations (using iconic primitives)	Specified by relative placement on axes	N/A	N/A
The Virtual Video Browser	Shot:scene:movie (Only video supported)	Via Actor relations	Implicit, via manual feature extraction	Via Actor, Scene, or Movie relations	Via object composition Petri nets	N/A	N/A
The algebraic video data model	Frame: video_segment: presentation (for video)	In description element of video algebra expressions	N/A	In description element of video algebra expressions	Via video node hierarchy and composition algebra operations	Application-dependent	When a node is used as an anchor
Matilda	Application-dependent	Application-dependent	Application-dependent	Application-dependent	Application-dependent	Application-dependent	Application-dependent

	Explicit media structure	Objects	Spatial relationships between objects	Events and actions involving objects	Temporal relationships between events and actions	Integration of syntactic and semantic information	Direct user-media interaction
The AVIS model	User-defined, frame:frame sequence:movie Audio may be specified according to user-defined structures	Modelled as video objects	N/A	Modelled as general activity types and more specific events	Via association maps and segment trees	Indirectly through links to HERMES	N/A
Informedia	Frame:scene Audio is split based on video	Human faces and text	Limited, implicit, via the location of identified faces and text	Only camera motion	Limited, when scenes change	N/A	N/A
Jabber	Media (only audio) is used in its entirety	Based on words spoken	N/A	Via semantic analysis of words spoken	N/A	N/A	N/A
Formal techniques:							
Video Classification project	frame:video clip: video (Audio still at theoretical stage)	Through a frame-based knowledge representation	Limited, via a priori image-understanding content models	Through a frame-based knowledge representation	N/A	Via DBMS interface	N/A

	Explicit media structure	Objects	Spatial relationships between objects	Events and actions involving objects	Temporal relationships between events and actions	Integration of syntactic and semantic information	Direct user-media interaction
The 'Hiroshima' model	Arbitrary	E.g., via Video_with_ annotation objects and associating predefined domain knowledge	N/A	Limited, e.g., Video_with_ annotation objects and associating predefined domain knowledge	Via composition of associated objects (e.g. 'scene')	Model explicitly uses 'independent' domain knowledge	N/A
Media abstraction	Media is used in its entirety	Expressed as Features	Could be done via Relations	Expressed as Relations	N/A	Via well-formed interface	N/A

Table 1.3 views the models in the previous review in terms of how they accommodate the seven semantic aspects of video and audio. There are a number of problems with how they do this:

Explicit media structure: In all models this is predominantly video-oriented, yet the structure is typically basic, never going beyond two or three levels, e.g. the 'NTT' model uses three levels (frame:scene:video) as does the Virtual Video Browser (shot:scene:movie). Others, such as IntelligentPad, use only two levels (frame:movie). Audio is often unspecified and unconsidered, with few of the models providing facilities for the handling of standalone audio, that is, audio which is separated from full-motion video sequences. For instance, QBIC, Visual Repair, CLORIS, the Stratification System, the video object data model, and the model underlying the Video Classification project all fail to specify a media structure for audio. In contrast, Jabber is completely audio-oriented, with no facilities for the handling of video. Even in models where facilities for both video and audio are provided, the functionality for audio is vastly inferior to that provided for video. This has obvious implications for the other semantic aspects of video and audio. For example, failing to provide audio functionality means that the representation of content objects within the medium will only be those present in the video stream. Those in the audio stream will be excluded. Therefore, audio content objects that make noises but do not ever appear on-screen are never represented.

Objects: The representation of content objects within the medium is the best addressed area of semantic multimedia information, with all models providing some way for content objects to be represented, whether this be for video or audio alone, or for both. Nevertheless, very few of the models are concerned with the *location* of these content objects, such as through the use of on-screen co-ordinates. Frequently, content-based multimedia models have been satisfied with just representing the presence of a content object in a particular frame or set of contiguous frames. Exceptions include those approaches utilising 'hot-spots', such as Sensitive Regions, the timeline-tree model, and the IntelligentPad architecture.

Spatial relationships between objects: The modelling of spatial relationships between content objects has been less adequately addressed than that of objects within the medium. At worst, the models do not provide any facilities for representing spatial relationships; at best, limited spatial relationships may be determined implicitly from other areas of the model, as in the case of Visual Repair (where the Location values of Element fields could be used), or Sensitive Regions (where the comparative co-ordinates of the Sensitive Regions might be used).

Events and actions involving objects: These have almost been ignored in physical models and techniques for locating content objects, but have received some attention in the stratification-based approaches and the formal techniques. On the whole, however, the semantic information has been of a very unstructured form. For example, the algebraic video data model relies on attached strings of text, as does the Stratification System. Other models which take a more structured approach, such as the video object data model, still essentially put text strings into arbitrary attribute-value pairs. Semi-structured information makes processing on this information, e.g. in terms of identifying and comparing terms, more difficult than if the information were fully structured. Tighter integration between these referenced elements (i.e. the events and actions), and also between the referenced elements and the content objects, becomes restricted as a result.

Temporal relationships between events and actions: This aspect has not been adequately addressed in content-based models, with many models providing no capabilities at all, e.g. the models underlying QBIC and Visual Repair, the Sensitive Regions model, the timeline-tree model, IntelligentPad, EVA, the Stratification System, Jabber, the model underlying the Video Classification project, and the media abstraction. Of those models that do provide for temporal relationships, the best are the approaches used in models such as the Media Streams model, the video object data model, and the Virtual Video Browser.

Integration of syntactic and semantic information: Very few of the models integrate the video and audio stream with the semantic information. Exceptions include the IntelligentPad architecture, the AVIS model and all three of the formal techniques. Similarly, the video object data model could be easily integrated into a relational multimedia database management system because the model is based on mathematical set theory, however this is application-dependent. The fact that all the models follow the structured-oriented approach to varying degrees emphasises the weak incorporation of this semantic aspect, since the structure-oriented approach places prime emphasis on *attaching* semantics to the media stream, not *integrating* the semantics with the media stream.

Direct user-media interaction: The problems with the previous six aspects has had repercussions for the provision of interactive video and audio within the model. Only a handful of models provide such facilities. Even where this has been provided (e.g. Sensitive Regions and IntelligentPad), the specifications for the other semantic aspects of audio and video have been left wanting. For example, the Sensitive Regions model does not provide any facilities for representing events and actions or temporal relationships, and spatial relationships may only be determined implicitly from the co-ordinates of the Sensitive Regions.

In addition to these problems, none of the above models provide functionality for all seven of the semantic aspects of video and audio. While functionality for all seven could be provided by the Matilda model, Matilda is merely a framework for examining where existing models, including semantic content-based ones, could be used.

All of these limitations have impinged upon the effectiveness with which multimedia is used within an MMIS. While the structure-oriented approach is certainly useful for 'virtual browsing' paradigms and sequential playing of multimedia, e.g. movies on CD-ROM, such models are clearly less effective when used within an MMIS. The domain of an MMIS typically provides knowledge about various entities within the domain. However, in the structure-oriented approach, there is no direct correspondence between the content represented and the related entities within the domain knowledge of the utilising application. The information is not 'ready to hand' and the MMIS must therefore search through the media stream sequentially to find segments of interest related to the pertinent entities. The MMIS is further impaired during searches by the fact that much extraneous information will typically be provided by the model that is inappropriate for the current task.

1.4 RESEARCH OBJECTIVE

A multimedia model that is concerned with content-based semantics needs to provide functionality for all seven of the semantic aspects of video and audio. Omission of one or more of the aspects would devalue the effectiveness of the model within a system since the aspects that have been included cannot make up for those that have not. The objective of this research is thus to develop a full-scale semantic content-based model that encompasses all seven of the semantic aspects of video and audio.

1.5 RESEARCH METHOD

To achieve this objective, this thesis adopts an alternative to the structure-oriented approach, namely that of the *entities of interest* approach. In this approach, as well as the traditional knowledge, the utilising MMIS's database also integrates semantic content-based information about raw video and audio data that is relevant to each entity, and their various properties, within the MMIS.

For example, consider an MMIS whose domain of application is geography. Entities here would include countries, mountains, and cultures, and more specifically England, Mount Sinai, and Indian respectively. Relevant video and audio footage would then be integrated with the information concerning these entities (e.g. general footage of Mount Sinai for the entity of the same name) as well as the properties of the entities (e.g. specific footage of the 1973 Battle of Sinai for a property such as 'events of interest' of the entity 'Mount Sinai').

Thus to contrast, in the structure-oriented perspective, the utilising application must search through the media stream to find segments of interest appropriate to a particular entity, whereas with the entities of interest perspective, all media segments that are relevant to a particular entity are collated together.

A method will be developed to guide the construction of the model for use within an interactive MMIS. Both the method and the model will be used in the development of an interactive MMIS for teaching zoology.

1.6 OVERVIEW OF THE REMAINING CHAPTERS OF THIS THESIS

The remainder of this thesis is structured as follows. *Chapter Two* proposes and discusses an entities-of-interest-based semantic content-based model for interactive MMISs. To achieve this, the model uses *multimedia frames (m-frames)* as the representation framework to store all the syntactic and semantic content-based information about the video and audio.

Chapter Three presents a method for developing an interactive MMIS that encompasses the full-scale semantic content-based model. The method consists of seven stages, which prescribe the manner by which the model and then the system are developed.

Chapter Four discusses the use of the method in the development of ARISTOTLE, an interactive MMIS for teaching zoology. Both the architecture and the functionality of the system are discussed.

Chapter Five discusses how ARISTOTLE implements and uses the seven semantic aspects, both individually and collectively.

Chapter Six summarises the thesis, discusses the contributions made, and details further research and development.

Chapter Two

A FULL-SCALE SEMANTIC CONTENT-BASED MODEL

> *"To be an inventor you look at things in an unconventional way."*
>
> — Trevor Bayliss, inventor of the clockwork radio

In the previous chapter, a distinction was made between the syntax and semantics of video and audio. This highlighted the fact that automated analysis and generation techniques have been predominantly process-oriented and have not focused on the actual representation of content within the media. Relevant research on semantic content-based modelling was then reviewed. It was argued that all current semantic content-based multimedia models have adopted a structured-oriented approach in which video and audio are modelled on a segment-by-segment basis, without relevance to the entities of interest to the MMIS. Seven semantic aspects of video and audio were identified based on the discussion of the existing semantic content-based models. Existing models were then interrogated with respect to these seven semantic aspects and a number of weaknesses in the structure-oriented approach were identified. Moreover, none of the models encompassed all seven aspects.

This chapter proposes a full-scale semantic content-based model for use in interactive MMISs. The model accommodates the seven semantic aspects of video and audio (Figure 2.1), and thus addresses weaknesses in existing content-based models. To achieve this, the model proposed in this thesis adopts an *entities of interest* approach for use in an MMIS as opposed to a structure-oriented approach. The model is *not computational*, but *representational*. That is, it provides a representation of the information required for the seven semantic aspects of video and audio to function within an interactive MMIS, but does not provide the procedures by which the seven semantic aspects may be computed.

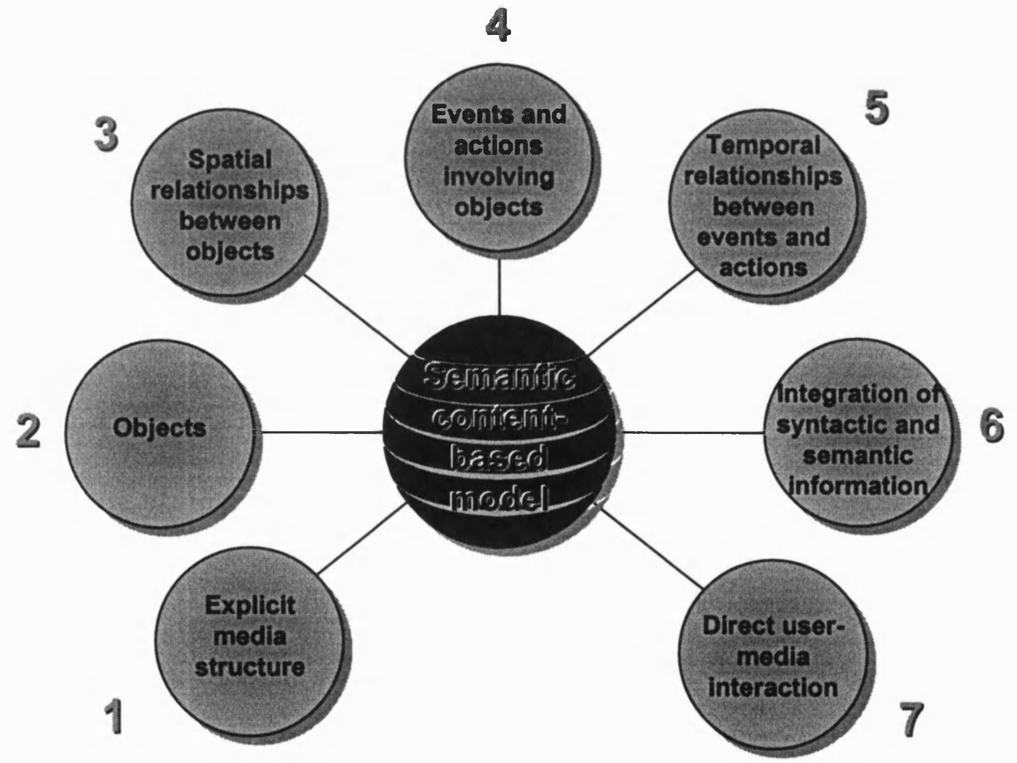

4
Events and actions involving objects
3
Spatial relationships between objects
5
Temporal relationships between events and actions
2
Objects
Semantic content-based model
Integration of syntactic and semantic information
6
Explicit media structure
1
Direct user-media interaction
7

Figure 2.1 The aspects that a full-scale semantic content-based model must accommodate.

The model uses the *multimedia frame*, or *m-frame*, as the representation framework to store all the syntactic and semantic content-based information about the media. Two types of m-frames are used: *syntactic m-frames* to model the syntactic content of the video and audio, and *semantic m-frames* to model the semantic content of the video and audio.

The following section discusses the underlying assumptions of the proposed model, after which the syntactic and semantic multimedia frames are presented. The full-scale semantic content-based model that uses the m-frames is then discussed. Following this, the chapter discusses how the proposed model caters for the seven semantic aspects of video and audio.

2.1 UNDERLYING ASSUMPTIONS

Text documents (e.g. books) are generally viewed as a number of chapters, which each consist of various sections. These sections contain many paragraphs, which are composed of many sentences, which are made up of various words, which are composed of many characters. This model of text is well understood. Unfortunately, no similar model exists for video or audio data. The most common approach is to view a video sequence as a collection of frames that present a specific scene. These frames are made up of a number of blocks (e.g. 16×16 pixel blocks), which each consist of pixels that are made up of luminance and chrominance values. Audio is typically broken down into merely a sequence of samples.

These notions of granularity are too syntactic and coarse to be useful within an MMIS. Often the granularity of the media affects the degree of interactivity within the end system because it determines how frequently the media can be interrupted. Moreover, these breakdowns differ for video and audio.

The model described in this chapter uses an explicit media structure that attempts to employ a more refined notion of granularity that may be applied to both video and audio in the same way – that is, it has a uniform structure.

Formally, a body of video data *VD* may be considered to consist of a number of *video frames* that are each of equal length t_{VD}, e.g. 40 ms ($\frac{1}{25}$ s) for PAL and 33 ms ($\frac{1}{30}$ s) for NTSC. It is thus possible to assume that the set of video frames of *VD* is the set $\{0, 1, 2, \ldots, n_{VD}\}$ for some fixed integer n_{VD}. Likewise, a body of audio data *AD* may be considered to consist of a number of *audio frames* that are each of equal length t_{AD}, e.g. 13 ms ($\frac{1}{75}$ s) for CD samples (i.e. the CD-Redbook standard), again assuming that the set of audio frames of *AD* is the set $\{0, 1, 2, \ldots, n_{AD}\}$ for some fixed integer n_{AD}. Therefore, for video and audio data that are intended to be played in synchrony, $t_{VD} = t_{AD} = t$ and $n_{VD} = n_{AD} = n$. This last assumption allows for a uniform definition of a *frame* as a segment of *video or audio* that is of short duration. Thus, video and audio may assume the same basic unit of division, and t can be stored as a constant within the system.

These initial formal assumptions enable video and audio frames to be aggregated. This is necessary because most segments of information are difficult to extract from single frames because they have meaning over time and are also often meaningless when taken out of context (Csinger et al., 1995). For example, it usually does not make sense to view a non-consecutive subset of frames nor does it make sense to view only disconnected frame segments. Moreover, it is not always possible to attribute events or actions based on a single frame. (Because the model is uniform, terms employed below will refer to both the audio and video data unless explicitly stated otherwise, e.g. the term *frame* will be used to mean a video frame *or* an audio frame, depending on the context of discussion.)

Consequently, a *shot* may be defined as an arbitrary sequence of contiguous frames that are related in that together they constitute some form of continuity in meaning within the sequence. For example, a shot may be a relatively short sequence of frames that depict a goal being scored in a football match, or it may be a much longer sequence that shows an entire football match. Expressed formally, a *shot* is a pair $[i, j]$, where $0 \leq i \leq j \leq n$, $n = n_{VD} = n_{AD}$, and $[i, j]$ represents the set of all frames between i and j, inclusive. In other words, $[i, j] = \{k \mid i \leq k \leq j\}$, where i is the *start* of the shot, j is the *end*, and k is a constituent frame of the shot. This definition enables a shot to consist of only one frame, e.g. $[4, 4]$ is a shot consisting only of frame 4. A partial ordering, $\prec$, can also be defined on the set of all shots as follows: $[i_1, j_1] \prec [i_2, j_2]$ if $i_1 \leq j_1 \leq i_2 \leq j_2$. This means that the shot denoted by $[i_1, j_1]$ precedes the shot denoted by $[i_2, j_2]$.

The above assumptions provide the foundation for representing the semantic content-based properties of multimedia. The frame is the smallest logical data unit (LDU) of the mathematical model. It is therefore impossible for the video or audio to be interrupted *within* a frame's time interval (i.e. t_{VD} or t_{AD}), and it must therefore be interrupted when one LDU (frame) has ended and another is about to begin. Even if the sequence was interrupted during the presentation of an LDU (frame) the sequence would treat it as having occurred either just at the start of the current LDU (frame) or just after (i.e. just at the start of the next LDU). Consequently, the smallest amount of time for which we need to represent information is t. All other information components, i.e. the shots, then become multiples of t.

Information is therefore associated with each frame, which defines the *syntax* of each frame's content. That is, the information is concerned with the presence and spatial arrangement of content objects on screen. Because content information also often only holds meaning within greater intervals of time than t (for the reasons discussed above), we also need to associate information with frame aggregates, i.e. shots. This information describes the *semantics* of a shot's content. In other words, it models the aggregate meaning of the sequence of related frames.

2.2 MULTIMEDIA FRAMES (M-FRAMES)

Multimedia frames, or *m-frames*, are the medium for representation within the full-scale semantic content-based model. They provide an 'object-oriented' manner by which the syntactic and semantic information may be grouped together and used conjointly. Since both syntactic and semantic information are required, the model uses two types of m-frames: *syntactic m-frames* which model the syntax of each frame's content, and *semantic m-frames* which model the semantics of several shots' contents. The structure differs for the two types of m-frame and they are therefore best described individually.

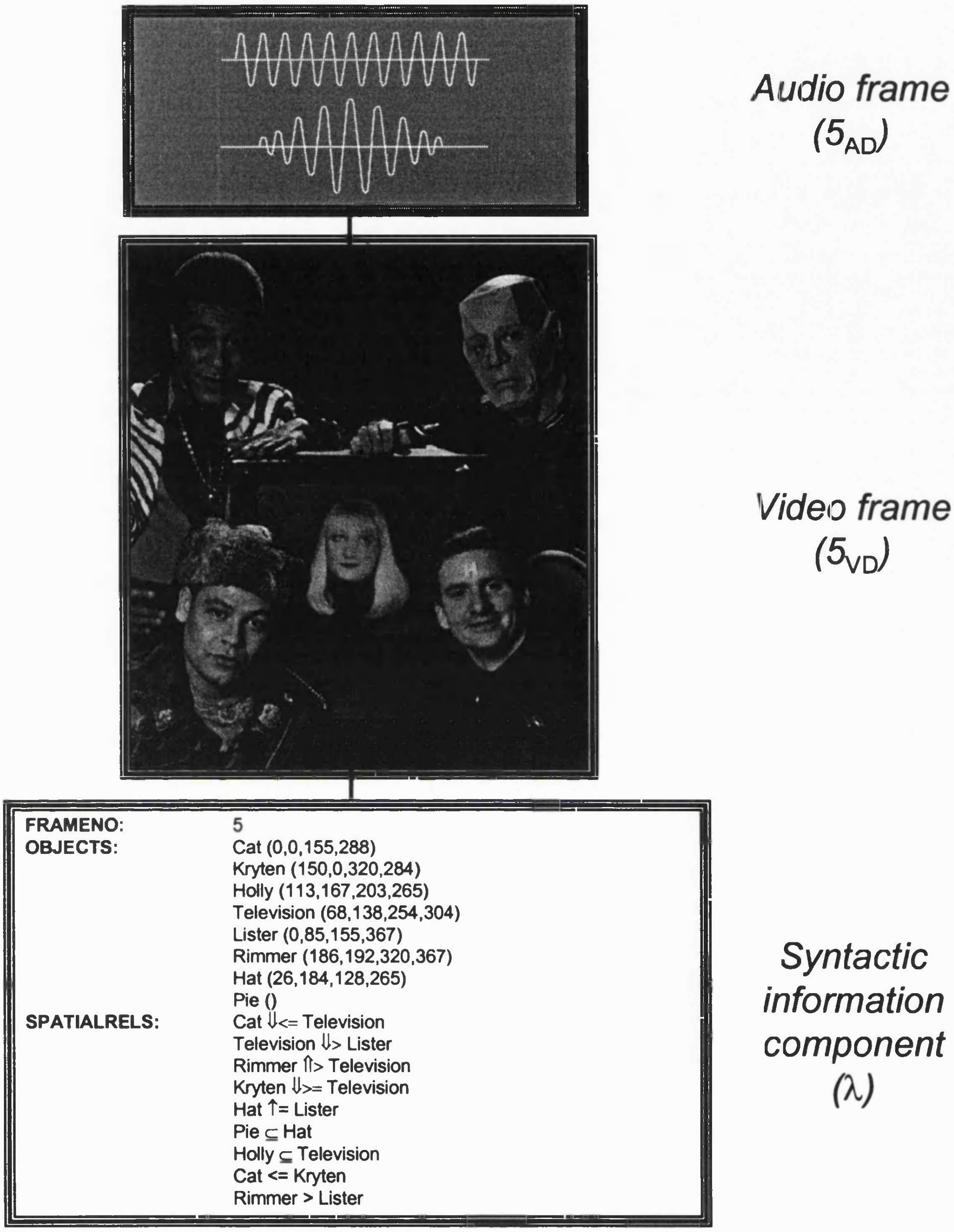
2.2.1 Syntactic m-frames (SYMs)

A syntactic m-frame models the content of a video/audio frame. The symbol SYM_k is therefore used to mean a syntactic m-frame for the video/audio frame *k*. Each video and audio within the MMIS therefore has a group of syntactic m-frames associated with it. The content that a syntactic m-frame models is the objects present, together with their on-screen co-ordinates, and the spatial relationships between the objects. Since audio does not have meaning on an individual frame-by-frame basis but has meaning over time, i.e. it is nonsensical to listen to one frame of audio, the information that is concerned with audio meaning in time is modelled by the semantic m-frames. Additionally, the unit of time that syntactic m-frames deal with, i.e. *t*, is too short to represent events and actions. Since these all occur in time, i.e. with a group of frames, they are catered for by the semantic m-frames.

Conceptually, each frame, k, has syntactic information associated with it that describes the syntactic content occurring within k. Formally, then, a syntactic m-frame is a triple $(k_{VD}, k_{AD}, \lambda)$, where k_{VD} is the video frame, k_{AD} is the audio frame, and λ is the syntactic information component. In cases where a video is used that does not have an associated audio stream, k_{AD} will be null. Where an audio is used that does not have an associated video stream, both k_{VD} and λ will be null. This is because, for the reasons stated above, audio only has meaning over a number of frames, e.g. a shot. While audio does have meaning in terms of which objects are present (i.e. making noise) within a frame, this information is unnecessary within a syntactic m-frame since the semantic m-frames will accommodate this information as a perspective on the audio (discussed in Section 2.2.2 below). Additional representation in syntactic m-frames is therefore redundant.

Figure 2.2 shows a conceptual representation of an example syntactic m-frame (SYM_5). The example video frame is taken from the BBC2 television series *Red Dwarf* and shows the stars of the show. For those unfamiliar with *Red Dwarf*, the characters are the Cat (top-left), Kryten (top-right), Arnold Rimmer (bottom-right), Dave Lister (bottom-left), and Holly (centre). The syntactic m-frame does not model everything in the video/audio frame but only those objects that are required by the MMIS.

The λ of syntactic m-frames contains three slots: FRAMENO, OBJECTS, and SPATIALRELS. The FRAMENO represents the frame number of the related audio *and* video, because the mathematical formalisms on which this model is based reference both uniformly. In the example, the frame number is 5.



FRAMENO:	5
OBJECTS:	Cat (0,0,155,288) Kryten (150,0,320,284) Holly (113,167,203,265) Television (68,138,254,304) Lister (0,85,155,367) Rimmer (186,192,320,367) Hat (26,184,128,265) Pie ()
SPATIALRELS:	Cat ⇓<= Television Television ⇓> Lister Rimmer ⇑> Television Kryten ⇓>= Television Hat ↑= Lister Pie ⊆ Hat Holly ⊆ Television Cat <= Kryten Rimmer > Lister

Figure 2.2 Conceptual representation of an example syntactic m-frame.

The OBJECTS slot stores the names and on-screen co-ordinates of the pertinent objects within frame *k*. These co-ordinates relate to a virtual rectangle around the object. In the example, the co-ordinates are based on a 320×357 video window and are listed in brackets next to the name of each object. Sometimes an object may be present, but will not be visible on-screen, e.g. it is inside or behind another object that masks its presence. In this case, no co-ordinates will be given for the object. For example, the object Pie is present inside Lister's hat but cannot be seen.

If an object is hidden and so does not have associated co-ordinates, but co-ordinates are required by the MMIS, the co-ordinates for the object can be determined by the MMIS based on the spatial relationships. Briefly, the system would determine why the object is hidden, e.g. because it is inside another object, and then use the co-ordinates of the obscuring object.

The SPATIALRELS slot stores the spatial relationships between the objects given in the OBJECTS slot. A number of primitives are used to model this information. These are discussed below. While the spatial relationships could be determined from the co-ordinates themselves, this is a very difficult problem for those spatial relationships that are three-dimensional. Moreover, it would be impossible to calculate certain relationships, such as the *inside* or *behind* spatial relations, since they are not always visible on-screen and can only be inferred from previous footage and the user's background knowledge. Finally, there is also the additional processing time of automatically calculating the spatial relations.

The spatial relationships are represented through the use of nine primitives, summarised in Table 2.1. Each spatial relationship has an inverse relationship, with the exception of the *touches* relation, =, whose inverse is equivalent to the original relation. The inverse relationships are provided to allow flexibility in the way the user chooses to represent the spatial relationships, and also to enable certain relationships to be inferred from those given. For example, if it is known that X is above Y, then it is also true that Y is beneath X. The primitives may also be combined to reduce the size of this information within the SYM. Table 2.2 shows the permitted combination of primitives.

Table 2.1 Primitives for spatial relationships within syntactic m-frames.

Spatial relation	Notation	Inverse spatial relation	Inverse notation
X touches Y	X = Y	Y touches X	Y = X
X above Y	X ↑ Y	Y beneath X	Y ↓ X
X inside Y	X ⊆ Y	Y encapsulates X	Y ⊇ X
X left Y	X < Y	Y right X	Y > X
X before Y	X ⇑ Y	Y behind X	Y ⇓ X

Table 2.2 How the spatial relationship primitives may be combined.

	=	↑	↓	⊆	⊇	<	>	⇑	⇓
=	▨	✓	✓	✓	✓	✓	✓	✓	✓
↑	✓	▨	▨	×	✓	✓	✓	✓	✓
↓	✓	▨	▨	×	✓	✓	✓	✓	✓
⊆	✓	×	×	▨	▨	×	×	×	×
⊇	✓	×	×	▨	▨	×	×	×	×
<	✓	✓	✓	×	✓	▨	▨	✓	✓
>	✓	✓	✓	×	✓	▨	▨	✓	✓
⇑	✓	✓	✓	×	✓	✓	✓	▨	▨
⇓	✓	✓	✓	×	✓	✓	✓	▨	▨

The *touches* spatial relation is denoted by =. For example, Chair = Table means that the chair is situated so close to the table that it is actually touching it. The inverse for this spatial relation is also *touches*, e.g. Table = Chair, since if the chair is touching the table it must also be true that the table is touching the chair.

The *above* primitive, ↑, is used to represent the fact that one object is above another in the video frame. For example, Helicopter ↑ Landing-Pad, is used to model the fact that a helicopter is positioned above a landing pad. The inverse of this primitive is the *beneath* primitive, ↓. For example, Landing-Pad ↓ Helicopter symbolises that the landing pad is beneath the helicopter. In the example given in Figure 2.2, the hat is above Lister (Hat $\uparrow$ Lister), however the hat is also touching Lister's head, and so the two spatial primitives are combined (Hat $\uparrow=$ Lister).

The *inside* primitive, $\subseteq$, is used to denote that one object is inside another in the video frame. For example, in Figure 2.2, Pie $\subseteq$ Hat is used to mean that a pie is inside the hat. Also, Holly $\subseteq$ Television means that Holly is inside the television. This primitive is powerful because it allows for the modelling of spatial relationships that are not visible on-screen. For instance, from the first example spatial relationship we know that a pie is inside the hat, but the pie is not visible in the video frame because the hat prevents this (hence the lack of co-ordinates beside Pie in the OBJECTS slot). The *encapsulates* primitive, $\supseteq$, is the inverse of *inside*. For example, Hat $\supseteq$ Pie would denote that the hat is encapsulating the pie. The spatial relationships could be represented in either way. The subset, $\subseteq$, and superset, $\supseteq$, symbols are used for *inside* and *encapsulates* respectively, because an object that is inside another object is analogous to it being a subset of that object, e.g. one chocolate is essentially a subset of the set 'a box of chocolates' since it contains a number of chocolates. The same is true for an *encapsulates* relationship.

The *left* primitive, $<$, indicates that one object is situated left of another object. For example, Lister $<$ Rimmer, indicates that Lister is situated on Rimmer's right-hand side (i.e. to the left). The inverse is the *right* primitive, $>$, which is used in the example in Figure 2.2: Rimmer $>$ Lister. The $<$ symbol was chosen for left because an object that is to the left of that object is 'less than' that object on the horizontal plane. Similarly for the $>$ symbol. The example also combines the *left* primitive with the *touches* relation (Cat $<=$ Kryten) to indicate that the Cat is to the left of Kryten, but is also touching Kryten.

The combination of primitives strengthens the case for using the < and > symbols when they are combined with the touches operator '='. In this case, to say that X <= Y means that, formally, X is less than or equal to Y on the horizontal plane. The *symbols* are also commutative, e.g. ⇑< is equivalent to <⇑. However, for obvious reasons, the same is not true of the entire *expression*.

The *before*, ⇑, and *behind*, ⇓, primitives model three-dimensional spatial relations. The *before* primitive denotes that one object is situated in front of another. For example, Rimmer ⇑ Television means that Rimmer is positioned in front of the television in the video frame. In the case of Figure 2.2, Rimmer is in front of the television and to the right of it, hence Rimmer ⇑> Television. The *behind* primitive acts as the inverse, e.g. Television ⇓ Rimmer. The complex spatial relation Cat ⇓<= Television in Figure 2.2 indicates that the Cat is behind the television and to its left, but is also touching the television. As with *inside* and *outside* expressions, *before* and *behind* may refer to objects that are not visible on-screen, e.g. if one object completely masks the object behind it.

The spatial relationships can be better understood if they are represented as an annotated spatial network diagram, where the objects are nodes, and the relationships are arcs between the nodes appended with the appropriate notation(s). The annotated spatial network diagram for the spatial relationships of Figure 2.2 is illustrated in Figure 2.3.

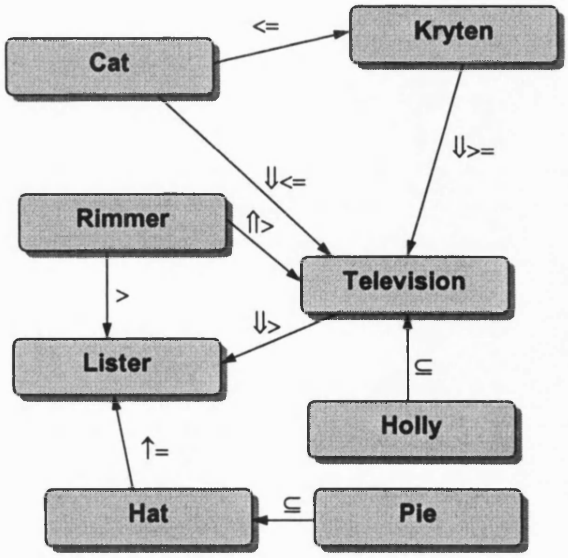


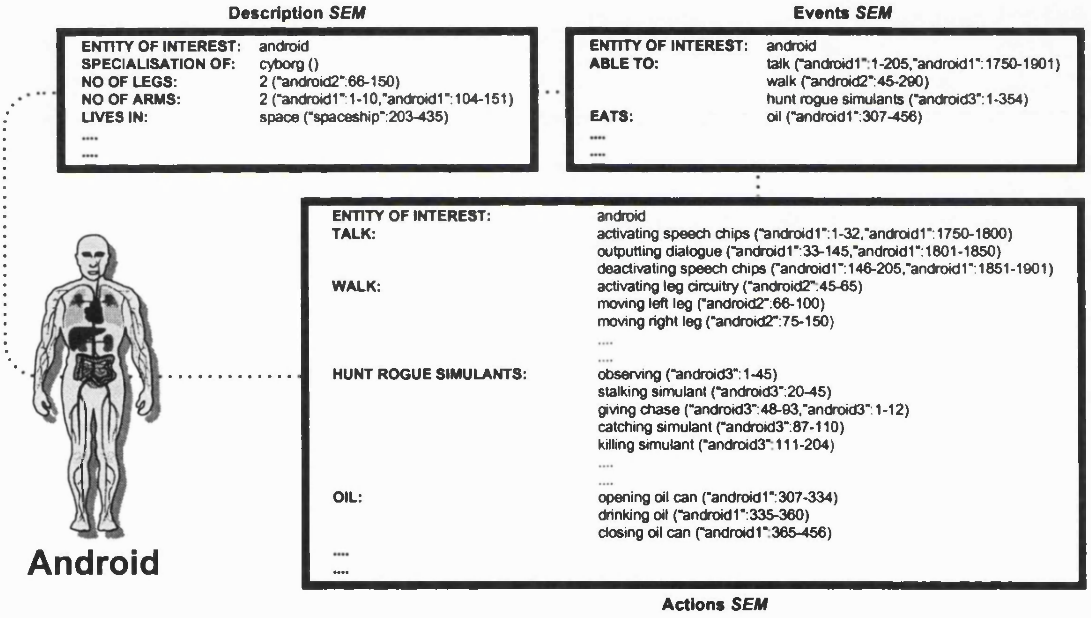
Figure 2.3 Conceptual representation of the spatial relationships detailed in Figure 2.2.

2.2.2 Semantic m-frames (*SEMs*)

A semantic m-frame (*SEM*) provides information about the semantic content of various segments of video and audio frames that are related to a particular concept that is pertinent to the MMIS. It is thus within the *SEMs* that shots are defined. A *SEM* also provides semantic information that is not related to a media segment within the MMIS. In this way, *all* information related to an entity of interest to the MMIS, i.e. both content-based and non-content-based information, is kept together.

Each entity of interest to the MMIS is represented by a collection of three *SEMs*: (1) the **Description *SEM*** describes the entity of interest, (2) the **Events *SEM*** models the events that are associated with the entity of interest, and (3) the **Actions *SEM*** models the constituent actions of the events modelled in the Events *SEM*. The *SEMs* therefore group together media segments that are related to an entity of interest to the system.

Figure 2.4 shows a conceptual representation of Description, Events, and Actions *SEMs* for an Android entity of interest, which is a specialisation of a Cyborg entity of interest. Each slot within the *SEM* represents a particular *perspective* on the multimedia content, with the slot values representing more specific *instances*. The slots and values of the *SEMs* are defined by the domain of discourse of the MMIS and the entities of interest to the system, unlike *SYMs* which have a predefined format. Shots are defined for instances within the *SEMs*, with multiple shots separated by commas within the figure.



ENTITY OF INTEREST:	android
SPECIALISATION OF:	cyborg ()
NO OF LEGS:	2 ("android2":66-150)
NO OF ARMS:	2 ("android1":1-10,"android1":104-151)
LIVES IN:	space ("spaceship":203-435)
....	
....	

ENTITY OF INTEREST:	android
ABLE TO:	talk ("android1":1-205,"android1":1750-1901)
	walk ("android2":45-290)
	hunt rogue simulants ("android3":1-354)
EATS:	oil ("android1":307-456)
....	
....	

ENTITY OF INTEREST:	android
TALK:	activating speech chips ("android1":1-32,"android1":1750-1800)
	outputting dialogue ("android1":33-145,"android1":1801-1850)
	deactivating speech chips ("android1":146-205,"android1":1851-1901)
WALK:	activating leg circuitry ("android2":45-65)
	moving left leg ("android2":66-100)
	moving right leg ("android2":75-150)
	....
	....
HUNT ROGUE SIMULANTS:	observing ("android3":1-45)
	stalking simulant ("android3":20-45)
	giving chase ("android3":48-93,"android3":1-12)
	catching simulant ("android3":87-110)
	killing simulant ("android3":111-204)
	....
	....
OIL:	opening oil can ("android1":307-334)
	drinking oil ("android1":335-360)
	closing oil can ("android1":365-456)
....	
....	

Figure 2.4 Conceptual representation of Description, Events, and Actions semantic m-frames for an Android entity of interest.

As an example, consider the NO OF ARMS perspective for the Description *SEM* in Figure 2.4. This perspective enables those shots to be used within the videos and audios that are concerned with how many arms an android has. In this case, there is one instance: 2. The first shot reference, “android1”:1-10’, for this instance, means that the shot denoted by [1, 10] (i.e. frames 1 to 10 inclusive) within the video and audio called "android1" has content that highlights an android's two arms.

The *SEMs* rely on associated *SYMs* for the on-screen representation of the objects involved in the semantic information that the *SEMs* model. Additionally, because $t = t_{VD} = t_{AD}$ and $n = n_{VD} = n_{AD}$, an audio frame and a video frame are of equal duration and therefore the shot references within the *SEMs* are valid for the video *and* audio components of the syntactic m-frames (provided that the relevant *SYMs* have both video and audio frames). For example, the shot reference denoted by "android1":1-10 utilises the *SYM* defined in Figure 2.2, which is contained within the "android1" video and audio.

The perspectives also enable the semantic m-frame to accommodate overlaps of content, since each slot represents different perspectives. For example, the Description *SEM* in the figure has a reference to a 'two arms' segment at "android1":104-151 and an 'outputting dialogue' segment at "android1":33-145. Thus $[104, 151] \cap [33, 145] = [104, 145]$. That is, 2:35-40 has content of an android's two arms *and* an android outputting dialogue. In this way, semantic m-frames are not restricted to representing only one particular, concrete view of the semantic content of specific media streams.

Each of the *SEMs* associated with the Android entity of interest model a different type of semantic content. The Description *SEM* within the figure models semantic content information that relates to a description of what an android is. The perspectives are therefore description-oriented. For example, there is footage depicting the fact that an android has two legs and two arms, and also footage that shows that an android lives in space.

The perspectives within the Events *SEM* are event-oriented. Each perspective therefore groups together one or more related events. Thus, *talking*, *walking*, and *hunting* *rogue simulants* are all events that represent what an android is able to do. In Section 2.2.1, it was noted that the modelling of audio was catered for in the semantic m-frames through the *SEM*'s perspectives. The Events *SEM* in Figure 2.4 provides an example of this. The ABLE TO perspective has a *talk* instance that has two associated shots: "android1:1-205" and "android1:1750-1901". This means that these shots have content of an android talking, with 'talking' very much an audio-oriented perspective on the media.

The perspectives of the Actions *SEMs* are determined by the instances of the Events *SEMs*. For each instance within an Event *SEM* there is a corresponding perspective within the associated Actions *SEM*. For example, there is an ABLE TO *talk* event within the Events *SEM* in Figure 2.4, thus there is a TALK perspective within the Actions *SEM*. The instances of each perspective within the Actions *SEM* represent the constituent actions of the event that the perspective represents. For example, the constituent actions of talking are *activating speech chips*, *outputting dialogue*, and *deactivating speech chips*. Thus, the shots for the actions serve to segment each of the shots that were defined for the event, in the Events *SEM*, into specific actions.

Splitting up an audio-oriented event into a number of actions allows the modelling of very specific audio content. Actions can be used that are very specific to audio. For example, the talking event modelled in the Events *SEM* could have modelled actions that were more oriented towards *what* the android is able to say. For example, one constituent action in this case could be *saying the word 'hello'*. The related shot for this action would therefore be that segment of the talking shot which had content of an android saying 'Hello'. Using audio-oriented events and actions enables the semantic m-frame to model both general audio perspectives as well as more specific ones, leaving the MMIS free to utilise the level of detail required.

2.3 USING THE SYNTACTIC AND SEMANTIC M-FRAMES IN A FULL-SCALE SEMANTIC CONTENT-BASED MODEL

Figure 2.5 conceptually presents the full-scale semantic content-based model that uses the syntactic and semantic m-frames. It shows one video and audio stream, which is divided into $n+1$ frames each of duration t. However, an MMIS would typically have many video and audio streams. Because the start of each video and audio frame is a multiple of t, the frames are labelled as such, i.e. t, $2t$, $3t$, ... $(n+1)t$, according to their temporal position as a multiple of t.

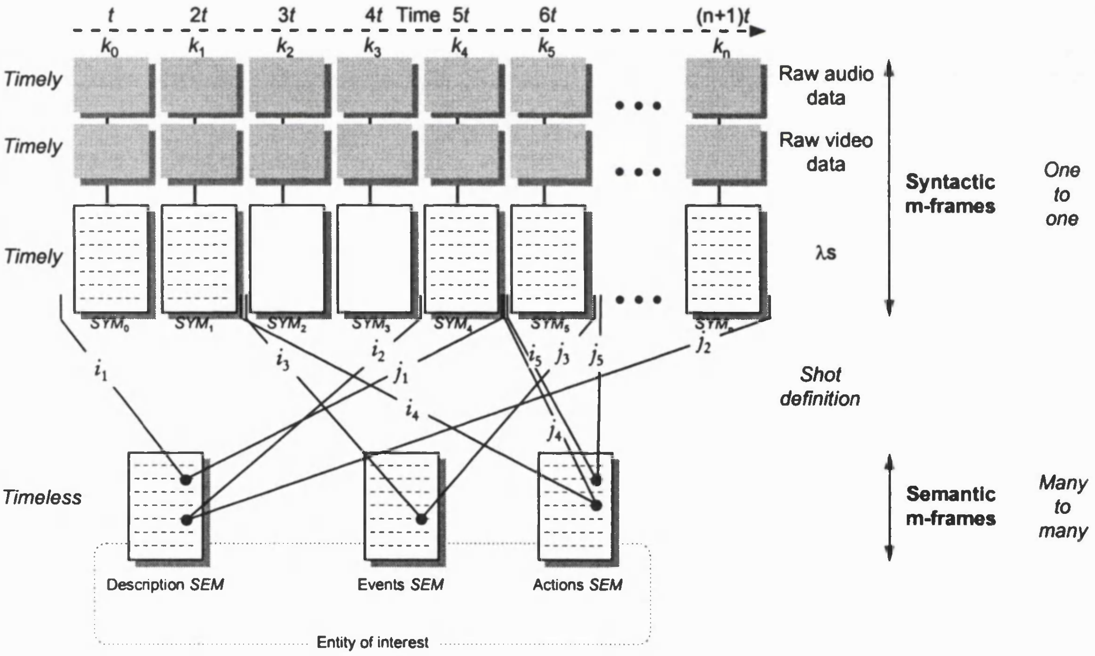


Figure 2.5 A full-scale semantic content-based model for interactive MMISs.

Each corresponding video and audio frame is grouped together to form a *SYM*. The three components of the syntactic m-frames – i.e. the audio frame, the video frame, and the syntactic information component – are *timely*, since the ordering of the frames (each of duration *t*) and the syntactic information components are important. This is because the syntactic m-frames maintain the original continuity of the media stream. The audio frames, video frames, and syntactic information components therefore have a one-to-one correspondence with each other.

In cases where λ remains completely unchanged for a number of frames within a stream of raw audio and video data, the model uses the λ of the first *SYM* in the sequence of identical *SYM*s. In other words, if the syntactic contents of frames k_x to k_y are equivalent to that of k_z, then the *SYM*s of k_x to k_y will simply refer to the λ of k_z. However, if there is a slight change, e.g. the co-ordinates of one object change by ±1, a new λ will be used. In this way, unnecessary duplication of information is eliminated, reducing the size of the stored information within the system. In Figure 2.5, SYM_2 and SYM_3 have blank syntactic information components. This represents the fact that their syntactic information has not changed from SYM_1. SYM_2 and SYM_3 are thus equivalent to SYM_1 and therefore reference its syntactic information component.

The semantic m-frames are shown at the bottom of Figure 2.5. The three that are shown form the Description, Events, and Actions *SEM*s of an entity of interest. Each *SEM* use various shots from the given audio and video stream. The semantic m-frames do not access the video and audio frames directly. Instead, this data is provided via the syntactic m-frames, which encapsulate the raw audio and video data. To simplify the figure, the Events *SEM* uses only one shot, while the Description and Actions *SEM*s uses two shots each. The two shots defined by the Actions *SEM* segment the shot that is defined by the Events *SEM*.

The shots are labelled by the start and end *SYM*s using the terminology of the underlying assumptions: the start of a shot is labelled with an '*i*' and the end of a shot with a '*j*'. The subscript indicates the number of the shot, i.e. i_1 and j_1 indicate a sequence of frames constituting one shot which is composed of SYM_0 to SYM_4.

The shots are permitted to overlap. For example, $[i_1, j_1]$ and $[i_2, j_2]$ both include SYM_4 (and therefore k_4). The semantic m-frames are thus *timeless* because the ordering of *SEMs* is irrelevant. *SEMs* model shots in a way that is unrelated to the original placement of those shots within the media stream, despite the fact that shots, by their definition, must be composed of a number of contiguous (timely) frames. Thus, one *SEM* may use many *SYMs*, and one *SYM* may be used by many *SEMs*. The *SEMs* and *SYMs* therefore have a many-to-many correspondence with each other.

In its use of media through the use of SYMs, the semantic m-frame is indifferent to whether the shot reference is for a video shot, an audio shot, or a video *and* audio shot. It merely references syntactic m-frames which will provide their associated video/audio frames. This eases the ability with which synchronised audio and video can be used separately. It is the MMIS that will determine whether it is currently using video, audio or both (via the syntactic and semantic m-frames). Thus, while a *SEM* and its associated *SYMs* are conceptually joined together in a virtual structure, they remain physically distinct. Video and audio frames may therefore be integrated into multiple *SEMs* within the same MMIS. This is important for the purposes of independent use of the video and audio streams.

The full-scale semantic content-based model presented here enables the interactive MMIS that is using it to ask questions of, and receive answers from, the modelled audio and video. These questions and answers centre around understanding the content and context of the media that is currently being used, as well as other questions that the model can provide answers to which allow the MMIS to use certain media according to its goals and objectives:

- **What is happening within the media at time *xt*?** This is answered by reference to the relevant syntactic and semantic m-frames.
- **What is being interacted with at time *xt*?** For example, what object has been clicked on? This is provided through the OBJECTS slot and the co-ordinates associated with each object.
- **What is the relative context of interaction?** In other words, what else has, is, and will be going on within the media? This current context is provided by the semantic m-frame currently being utilised, whether this be a Description, Events, or Actions SEM. The MMIS can use the syntactic m-frames to know what will take place within the scope of current shot, in terms of object movements, and the semantic m-frames to know what will take place beyond the scope of the current shot, in terms of events and actions.
- **Which media have footage of the object, events or actions currently being used within the system?** This is quickly provided by the Description, Events, and Actions semantic m-frames, which integrate together all the relevant shots related to objects, events, or actions, respectively.

The following section examines the developed model more specifically in terms of its provision for each of the seven semantic aspects of video and audio.

2.4 HOW THE DEVELOPED MODEL ACCOMMODATES THE SEVEN SEMANTIC ASPECTS OF VIDEO AND AUDIO

The developed full-scale semantic content-based model provides functionality for all of the seven semantic aspects of video and audio.

2.4.1 Explicit media structure

The developed model makes the media structure explicit by splitting up both audio and video into *frames* and *shots*. Each frame is of an equal length, t. Each shot is a multiple of t, and is an arbitrary sequence of contiguous frames that have related meaning and thus are grouped together. A shot is formally expressed by a pair $[i, j]$, where $[i, j] = \{k \mid i \leq k \leq j\}$. Thus i is the *start* of the shot, j is the *end*, and k is a constituent frame of the shot.

Organising the raw video and audio stream in this way enables semantic information about frames or shots to be incorporated into the m-frames. Moreover, the arbitrary nature of a shot means that what is meant by a shot may be adapted to the particular purposes of the semantic m-frame, depending on the current context. Because the start and end markers of a shot (i and j) are variable, a single shot can be used to denote a short sequence about a single object or a lengthy clip of an entire event (e.g. a football match).

Since the model uses the same explicit media structure for both audio and video, the processing that takes place on the two media within the model is also uniform. This is illustrated particularly by the semantic m-frames. SEMs model a particular perspective for a particular segment of a media stream and are detached from whether it is an audio stream, a video stream, or a synchronised audio *and* video stream.

2.4.2 Objects

This semantic aspect is catered for by the OBJECTS slot of the syntactic m-frames. The OBJECTS slot stores the names and on-screen co-ordinates of the pertinent objects within an audio/video frame, where the co-ordinates relate to a virtual rectangle around the object. In cases where an object is present, but is not visible on-screen, no co-ordinates will be given for that object in the relevant SYMs. This enables its presence to still be known.

2.4.3 Spatial relationships between objects

The SPATIALRELS slot of the syntactic m-frames stores the spatial relationships between the objects given in the OBJECTS slot, enabling the accommodation of spatial relationships by the model. Nine primitives are used to model this information: *touches* (=), *above* (↑) and *beneath* (↓), *inside* (⊆) and *outside* (⊇), *left* (<) and *right* (>), *before* (⇑) and *behind* (⇓). The inverse relationships allow flexibility in the way the user chooses to represent the spatial relationships, and also enable the system to infer certain relationships from those given, e.g. if X ↑ Y, then Y ↓ X is also true. The use of a SPATIALRELS slot to directly model spatial relationships between objects avoids the difficult problem of using co-ordinates to determine three-dimensional spatial relationships between objects, some of which are completely hidden on-screen.

2.4.4 Events and actions involving objects

Events and actions are represented by slots and associated values within the Events and Actions semantic m-frames, respectively. Each event that is represented within the Events SEM is split into its constituent actions in the Actions SEM that is associated with a particular entity of interest. For example, a CHILD BIRTH slot with an associated value of *natural birth*, in an Events *SEM*, would specify one or more shots that depict the event of natural child birth. The corresponding Actions *SEM* would split the natural birth event into its constituent actions, such as lying on the floor, opening legs, and pushing. The media streams that represent the event within the Events *SEM* are therefore split into more specific actions within the Actions *SEM*. The shot references within the *SEMs* utilise the *SYMs* that have been defined for the shot's constituent frames. The perspectives and instances used within the *SEMs* are entirely user-definable and are only constrained by the domain of discourse of the MMIS.

2.4.5 Temporal relationships between events and actions

Semantic m-frames are able to accommodate overlaps of content among shots, which are numbered in terms of their constituent frames. This enables temporal relationships between events and actions to be determined, such that it is possible to ascertain which events occur before, after or during which other events. By the same principle, it is also possible to determine which actions occur before, after or during which others within an event. For instance, the shots [1, 25] and [20, 37] have the shot [20, 25] in common, i.e. $[1, 25] \cap [20, 37] = [20, 25]$. This means that [20, 25] has the content specified by the first shot *and* the content specified by the second shot. If [1, 25] and [20, 37] depict content of two different actions (or events), then those actions (or events) occur simultaneously during the shot [20, 25].

2.4.6 Integration of syntactic and semantic information

The use of semantic m-frames by an MMIS enables more than mere links to the system. Since the semantic m-frames reference the syntactic m-frames, the two are integrated together. This is specifically illustrated by the Description SEMs, which model descriptive semantic content-based information about a particular entity of interest. Such information is usually composed of objects which will also be modelled within the associated SYMs. For example, if a Description SEM models the fact that a telephone has buttons for a particular shot, then the SYMs associated with that shot will also have information concerning the presence and location of those buttons on-screen.

Furthermore, because the semantic m-frames integrate together all of the information an MMIS requires, an entity of interest hierarchy that is composed of Description, Events, and Actions SEMs, together with the associated SYMs, constitutes an MMIS's knowledge base or database. Thus the developed model becomes an integral part of the processing and functionality of the MMIS that uses it.

2.4.7 Direct user-media interaction

The frame is the smallest logical data unit of the full-scale semantic content-based model developed in this chapter. The video or audio may therefore not be interrupted *within* a frame's time interval (i.e. t), but may be interrupted *between* frames, with the system assuming that interaction is concerned with the last played frame (since it is improbable that the user would want to interact with a frame that they had not yet seen or heard). The definition of a frame as a sequence of video or audio of very short duration (e.g. 40 ms) within the model enables user-invoked or system-invoked interaction as frequently as at the end of every frame, providing a feeling of real-time interaction.

At the same time, the syntactic information component of each and every audio and video frame within the MMIS provides the system with the comprehensive detail necessary to enable interaction with all relevant on-screen objects. For example, the objects and co-ordinates detailed in the OBJECTS slot of the SYM enables the MMIS to determine what object has just been clicked on by the user or to ask the user to click on a particular object on screen.

Furthermore, the current context of interaction is provided by the details of the semantic m-frame currently being utilised, since it provides details of the current concept. The semantic m-frames use syntactic m-frames to link associated videos and audios to an interaction. As an example, consider the case where a user clicks on a house in a video/audio segment and is then presented with video/audio segments of the inside of the house. Further clicking on a vase inside the house might link the user to footage depicting famous antique vases.

2.5 SUMMARY

This chapter has presented a full-scale semantic content-based model that accommodates all of the seven semantic aspects of video and audio. The model achieves this through the use and integration of both syntactic and semantic multimedia frames (m-frames). Three types of semantic m-frames (SEMs) are used to represent an entity of interest: (1) a Description SEM describes the entity of interest, (2) an Events SEM models the events that are associated with the entity of interest, and (3) an

Actions *SEM* models each of the events represented in the Events *SEM* in terms of their constituent actions.

The following chapter presents a method for developing an interactive MMIS that uses the full-scale semantic content-based model.

Chapter Three

A METHOD FOR DEVELOPING INTERACTIVE MULTIMEDIA INFORMATION SYSTEMS THAT ENCOMPASS THE FULL-SCALE SEMANTIC CONTENT-BASED MODEL

"Nothing in progression can rest on its original plan.
We may as well think of rocking a grown man in the cradle of an infant."
— Edmund Burke

Thus far, this thesis has argued that a semantic content-based multimedia model that omits one or more of the seven semantic aspects of video and audio has devalued effectiveness within an interactive MMIS. The previous chapter developed a full-scale semantic content-based model that provides functionality for all seven of the semantic aspects, thereby addressing weaknesses in the structure-oriented approach of existing semantic content-based models. To this end, the model used an alternative approach, that of entities of interest, based on *multimedia frames* (*m-frames*): *syntactic m-frames* (SYMs) were used to model the syntactic content of the raw video and audio, while *semantic m-frames* (*SEM*s) were used to model the semantic content. The semantic content was modelled around an entity of interest in terms of a Description, an Events, and an Actions *SEM*.

This chapter presents a method for developing interactive MMISs that encompass the full-scale semantic content-based model, proposed in Chapter Two. The method has seven stages that take the developer from the initial description of entities of interest, through to the implementation of the multimedia support environment that uses the full-scale semantic content-based model. Figure 3.1 shows the stages of the method.

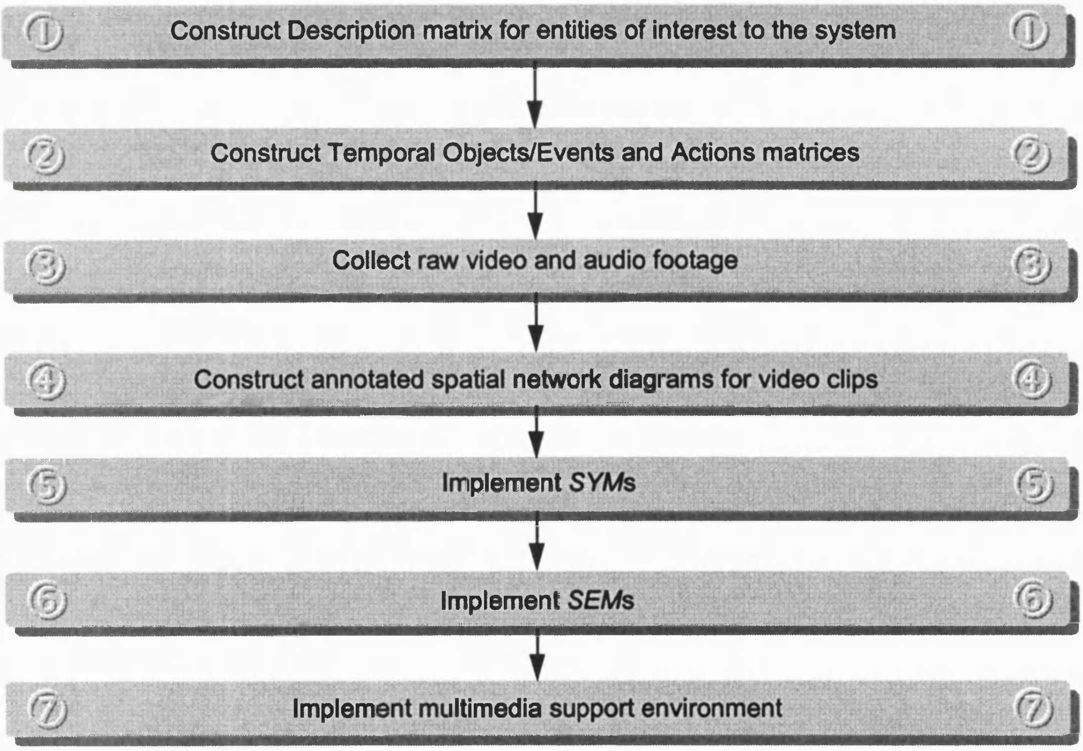

① Construct Description matrix for entities of interest to the system ①
② Construct Temporal Objects/Events and Actions matrices ②
③ Collect raw video and audio footage ③
④ Construct annotated spatial network diagrams for video clips ④
⑤ Implement *SYMs* ⑤
⑥ Implement *SEMs* ⑥
⑦ Implement multimedia support environment ⑦

Figure 3.1 A method for the development of an interactive MMIS that uses the full-scale semantic content-based model.

The method begins with the identification and description of the entities of interest to the system. Then, for each entity of interest, a matrix is constructed that details objects, events, and actions and the temporal relationships between those events and actions. The developer now has sufficient information to be able to know which raw video and audio footage is required. Once collected, the spatial relationships between objects on-screen may be determined and represented through annotated spatial network diagrams. These diagrams are then used to help with the implementation of the syntactic m-frames. Next, the semantic m-frames are implemented. Finally, the multimedia support environment that uses the full-scale semantic content-based model is implemented. The following sections discuss in detail the activity involved within each of the seven stages of the method.

3.1 STAGE 1: DESCRIBE ENTITIES OF INTEREST TO THE SYSTEM

The full-scale semantic content-based model is based on an entities of interest perspective, where the *SEMs* are organised around entities of interest to the system. Thus, the initial stage of the method focuses on the identification and description of entities of interest to the system. This provides the information needed to construct the Description *SEMs*, and thus indicates the descriptive audio and video footage that will be required in the system.

The entities of interest to be used are first organised into a structure, such as flat, hierarchical, or networked, depending on the viewpoint taken of the domain. Figure 3.2

shows a hierarchical arrangement of six possible entities of interest for a domain of antisocial behaviour.

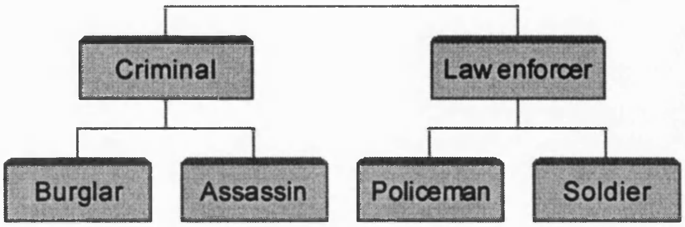


Figure 3.2 Entities of interest for a domain of antisocial behaviour, arranged hierarchically.

Each entity of interest within the structure is then described. The type and level of description for each entity of interest is, like the structure, determined by the viewpoint taken of the domain. At this stage, the events and actions that are associated with an entity of interest are not of concern, since they are the focus of Stage 2 of the method. A Description matrix is used to describe the entities of interest. The description perspectives used for the domain form the columns of the matrix, while the entities of interest form the rows. The instances are noted in the column-row intersections. Figure 3.3 provides an example Description matrix for the entities of interest in Figure 3.2. The example uses four description perspectives: Specialisation of, Characteristics, Wears, and Tools. Each entity of interest then uses one or more of these perspectives.

DESCRIPTIONS: ENTITIES:	Specialisation of	Characteristics	Wears	Tools
Criminal		Breaks law		
Law Enforcer		Enforces law Keeps order		
Burglar	Criminal		Mask Gloves	Torch Knife Pistol
Assassin	Criminal		Balaclava Gloves	Rifle
Policeman	Law enforcer		Helmet Uniform	Truncheon Radio Notepad
Soldier	Law enforcer		Military hat Uniform	Machine gun Knife Grenades

Figure 3.3 An example Description matrix, based on Figure 3.2.

3.2 STAGE 2: CONSTRUCT TEMPORAL OBJECTS/EVENTS AND ACTIONS (TOEA) MATRICES

The next stage involves the construction of Temporal Objects/Events and Actions (TOEA) matrices, which plot objects against the events and actions involving those objects. The TOEA matrix is temporal because it indicates the sequence in which events and actions occur. One TOEA matrix is constructed for each entity of interest identified in the previous stage. However, entities of interest that do not have associated events and actions, or whose events and actions are determined to be insignificant to the domain, are excluded from this stage. The TOEA matrices provide the information needed to construct the Events and Actions SEMs, and thus indicate the event- and action-oriented video and audio footage that will be required in the system.

Construction of a TOEA matrix begins by deciding upon the events to be included. These are placed in the topmost row, not necessarily in temporal order. Next, the objects that are to be included in the events are listed in the leftmost column of the matrix. Actions are then named in the row directly underneath the events. The individual actions that together constitute a particular event are grouped underneath that event. Since an action may occur more than once within the same event, the number of instances of each action is indicated by a list of letters in parentheses underneath the action name. Thus, if there is only one instance of an action then '(a)' will be placed underneath the action name, if there are two instances of an action then '(a, b)' will be placed underneath the action name, and so on.

Figure 3.4 shows an early TOEA matrix for the Burglar entity of interest that was described in the previous example. The matrix has plotted six objects, against three events and seven actions. The Fight event is composed of three unique actions; however, there are two instances of the Punches action, making four actions in total. The Burgle and Work events are composed of two actions each.

To indicate which objects are involved in which events and actions, a '1' or a '2' is placed at certain intersections in the matrix: a '1' indicates that this object is the one doing the action (within the event), whereas a '2' indicates that this object is the one that the action is being done to. For example, the matrix in Figure 3.4 shows a Grabs action between the Policeman and the Burglar objects. The '1' that is located where the Policeman row and Grabs column intersects indicates that it is the policeman that grabs the burglar, and not the burglar that grabs the policeman.

	EVENTS:	Fight			Work		Burgle	
	ACTIONS:	Grabs (a)	Kicks (a)	Punches (a, b)	Sells (a)	Collects (a)	Steals (a)	Arrests (a)
OBJECTS:	Cat						2_a	
	Burglar	2_a	1_a	1_a, 2_b	1_a	1_a	1_a	2_a
	Television						2_a	
	Policeman	1_a	2_a	2_a, 1_b				1_a
	Shoes				2_a			
	Pay					2_a		

Figure 3.4 Initial construction of a Temporal Objects/Events and Actions matrix for a Burglar entity of interest.

Where there are multiple objects acting on each other, a '1' or a '2' is used for each additional object, as appropriate. For example, the matrix in Figure 3.4 also indicates that a burglar steals a cat and a television. Thus, the Burglar object is the one performing the action and has a '1' placed at the intersection, whereas the Cat and the Television objects are the objects that the Stolen action is being performed on and therefore a '2' is placed at both of their intersections with the action.

A subscript letter appended to each number is used to indicate situations where the same objects are involved in different instances of an action. For example, the matrix in Figure 3.4 shows that there are two Punches actions (indicated by the '(a, b)' under the action name). Thus, the numbers that indicate one instance of the Punches action are labelled with a subscript 'a', whereas the numbers that constitute the other Punches action are labelled with a subscript 'b'. The matrix in Figure 3.4 therefore indicates that Punches action 'a' involves a burglar punching a policeman, and Punches action 'b' involves the policeman punching the burglar.

So far, the *sequence* in which events and actions take place *cannot* be determined from the initial construction of the matrix. That is, the order of events and actions is *not* specified by a left-to-right reading of the matrix. Consequently, the initial matrix is completed by the addition of temporal relationships between events and actions. The consideration of temporal relationships during Stage 2 of the method leads to a comprehensive understanding of the video and audio footage that will be required within the system. This understanding is invaluable during Stage 3, when the video and audio footage is collected. For example, without the TOEA matrix it would not be known whether it is necessary to film the policeman grabbing the burglar at the same time as the burglar is punching the policeman, or if it is sufficient to film the two actions separately.

Temporal event relationships specify the sequence in which events occur. Because the temporal order of certain events in relation to other events is not always important, the events on the matrix form one or more groups. Membership of a group is indicated by a capital letter in brackets, after the event name, e.g. '(A)'. If the temporal order between all events on the TOEA matrix is important, then all of the events will form only one group, and all events will be labelled '(A)'.

The temporal event relationships are then indicated in the topmost row of the TOEA matrix by numbering the events. The first event in a group is always numbered '1', with subsequent events numbered '2', '3', and so forth. Events which occur simultaneously will share the same number, e.g. if two events occur at the start of a group, they will both be numbered '1'. Their shots within the Events SEMs will thus intersect. A capital subscript letter is appended to each number to indicate the group of events to which the temporal ordering applies.

The completed TOEA matrix for the initial matrix given in Figure 3.4 is shown in Figure 3.5. The matrix indicates two groups of events. The first group (A) consists of two events, Fight and Burgle, which occur simultaneously, and thus must be filmed simultaneously. The second group (B) contains one event: Work. The events have been separated into two groups because the temporal order of Work in relation to the other two events is irrelevant to the system to be developed. Thus, the order in which the Work event is filmed in relation to the other events is not important. If it were important for the Work event to be filmed, for example, before the other two events, then all events would form one group. The Work event would then be numbered 1_A, and the Fight and Burgle events would each be numbered 2_A.

TEMPORAL EVENT RELATIONSHIPS:		1_A			1_B		1_A	
EVENTS:		Fight (A)			Work (B)		Burgle (A)	
ACTIONS:		Grabs (a)	Kicks (a)	Punches (a, b)	Sells (a)	Collects (a)	Steals (a)	Arrests (a)
OBJECTS:	Cat						2_a	
	Burglar	2_a	1_a	1_a, 2_b	1_a	1_a	1_a	2_a
	Television						2_a	
	Policeman	1_a	2_a	2_a, 1_b				1_a
	Shoes				2_a			
	Pay					2_a		
TEMPORAL ACTION RELATIONSHIPS:	a	2_A	4_A	3_A	1_B	2_B	1_A	5_A
	b			4_A				

Figure 3.5 The completed Temporal Objects/Events and Actions matrix, based on Figure 3.4.

Temporal action relationships specify the sequence in which the actions of a given group of events occur. They are represented on the bottom half of the TOEA matrix. In order to distinguish different action instances, a separate row is used for each action instance. The sequence of actions within each event group is then indicated by numbering the actions. The first action of each event is always numbered '1'. Actions which occur simultaneously will share the same number, e.g. if two actions take place at the same time at the start of a group of events, they will both be numbered '1'. To distinguish numbering for separate event groups, the numbers are appended with a capital subscript letter. The letter indicates the event group to which the temporal ordering of the actions applies.

For the actions detailed on the TOEA matrix in Figure 3.5, the temporal action relationship numbers for 'a' actions are noted in row 'a' at the bottom of the matrix, whereas the temporal action relationship numbers for 'b' actions are noted in row 'b'. The temporal action relationships for event group A show that the burglar steals the cat and the television (1), then the policeman grabs the burglar (2), then the burglar punches the policeman (3), then the burglar kicks the policeman at the same time as the policeman punches the burglar (4), and then the burglar is arrested by the policeman (5). Because the Burgle and Fight events occur simultaneously, their actions are intermingled. They are not, however, intermingled with the Work event, since it forms event group B. Thus, the numbering of actions within the Work event begins again with '1'. The temporal action relationships for the Work event show that the burglar sells shoes (1), and then the burglar collects his pay (2).

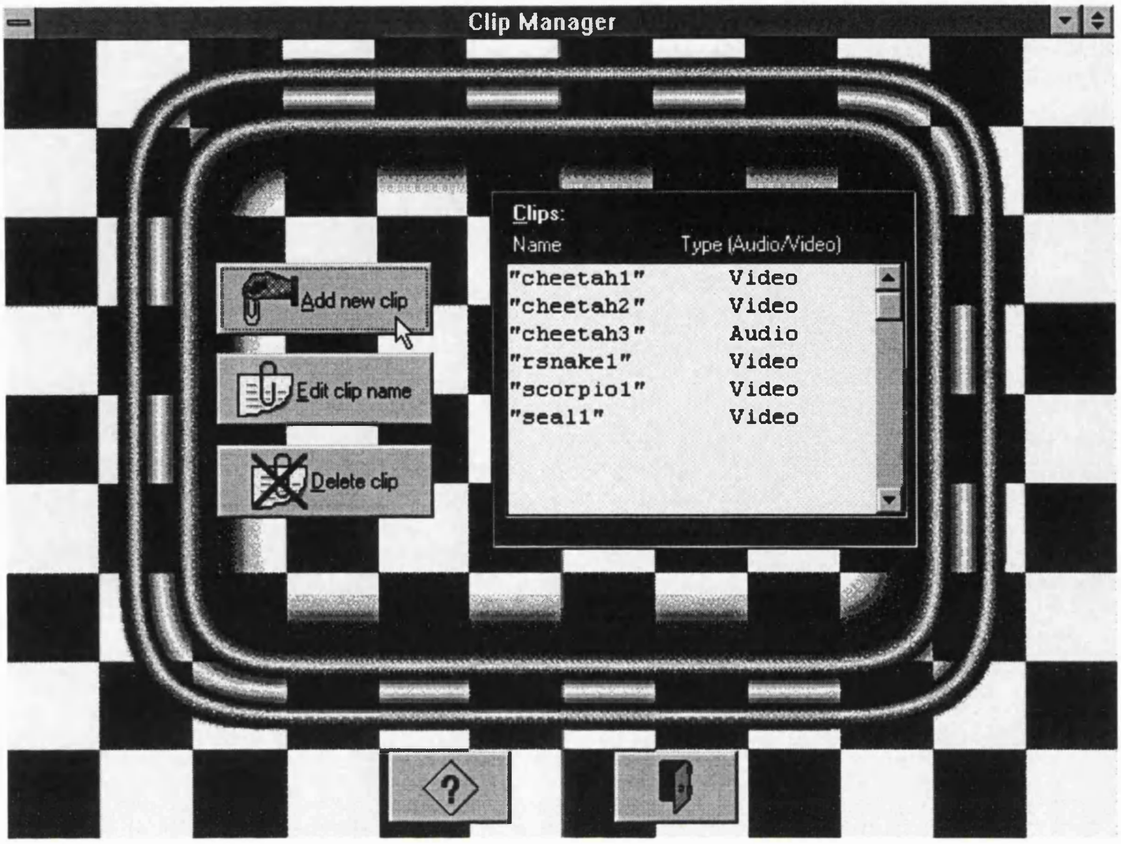
3.3 STAGE 3: COLLECT RAW VIDEO AND AUDIO FOOTAGE

The Description matrix and the TOEA matrices provide the developer with a specification of the video and audio footage that is required by the system. The

Description matrix indicates which properties of the entities of interest must be filmed. The TOEA matrices indicate the events and actions for which video and audio footage is required, and the order in which those events and actions are to be filmed. The TOEA matrices also provide details of the objects that must be included within each event and action that is filmed.

Description perspectives used in the Description matrix that have an audio theme indicate that audio footage in particular is required. For example, a Noise description perspective that has a *boom* instance would indicate a requirement for audio footage of a boom noise for the entity of interest. Similarly, audio-oriented events and actions on the TOEA matrix would also indicate a need for audio footage. For example, a Snoring event or action would indicate a requirement for audio footage of snoring.

Once filming has taken place, the video and audio are then captured digitally and edited into clips. A number of clips should be used to speed access to the shots. While all of the video and audio could be edited into one extended clip, having such a lengthy clip fetters the time it takes for shots to be cued up. To assist with the task of adding clips to a multimedia resource, a front-end software tool, such as the Clip Manager may be used (see Figure 3.6). The Clip Manager was developed to assist with the management of clips in ARISTOTLE, an interactive instructional MMIS that uses the full-scale semantic content-based model and is discussed in Chapter Four. The Clip Manager adds, modifies, and deletes clips within a given multimedia resource, and was developed in Asymetrix Multimedia ToolBook 4.0.



Clips:

Name	Type (Audio/Video)
"cheetah1"	Video
"cheetah2"	Video
"cheetah3"	Audio
"rsnake1"	Video
"scorpio1"	Video
"seal1"	Video

Figure 3.6 The Clip Manager.

3.4 STAGE 4: CONSTRUCT ANNOTATED SPATIAL NETWORK DIAGRAMS FOR VIDEO CLIPS

Once the video and audio footage to be used has been collected, the next stage is to model the spatial relationships of the objects within the video footage. This information is used when implementing the SYMs.

Chapter Two discussed the use of annotated spatial network diagrams to illustrate conceptually the spatial relationships between objects within a video frame. These diagrams are particularly useful during the development of the full-scale semantic content-based model as they serve as a quick way of representing all possible spatial relationships within each frame. During this stage, the video clips created in Stage 3 are taken in turn and played. One diagram is constructed for the first video frame, and then one additional diagram is constructed for each video frame where there is a change in the spatial relationships from the previous diagram.

The Description matrix and the TOEA matrices detail which objects of the many objects featured within the shots will be required by the system, and thus which objects are to be modelled within the SYMs. Other objects that feature in the video footage but were not detailed in the Description matrix or the TOEA matrices may also be modelled. Although these objects are incidental to the domain, their inclusion allows a more complete representation of the content of individual video frames.

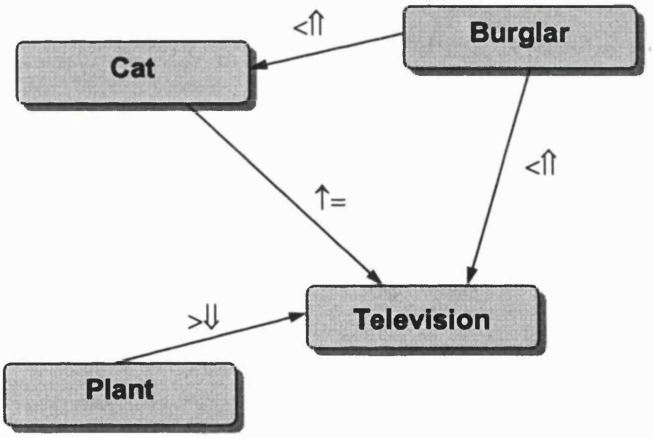


Figure 3.7 An example annotated spatial network diagram for a video frame from the Burgle event represented in Figure 3.5.

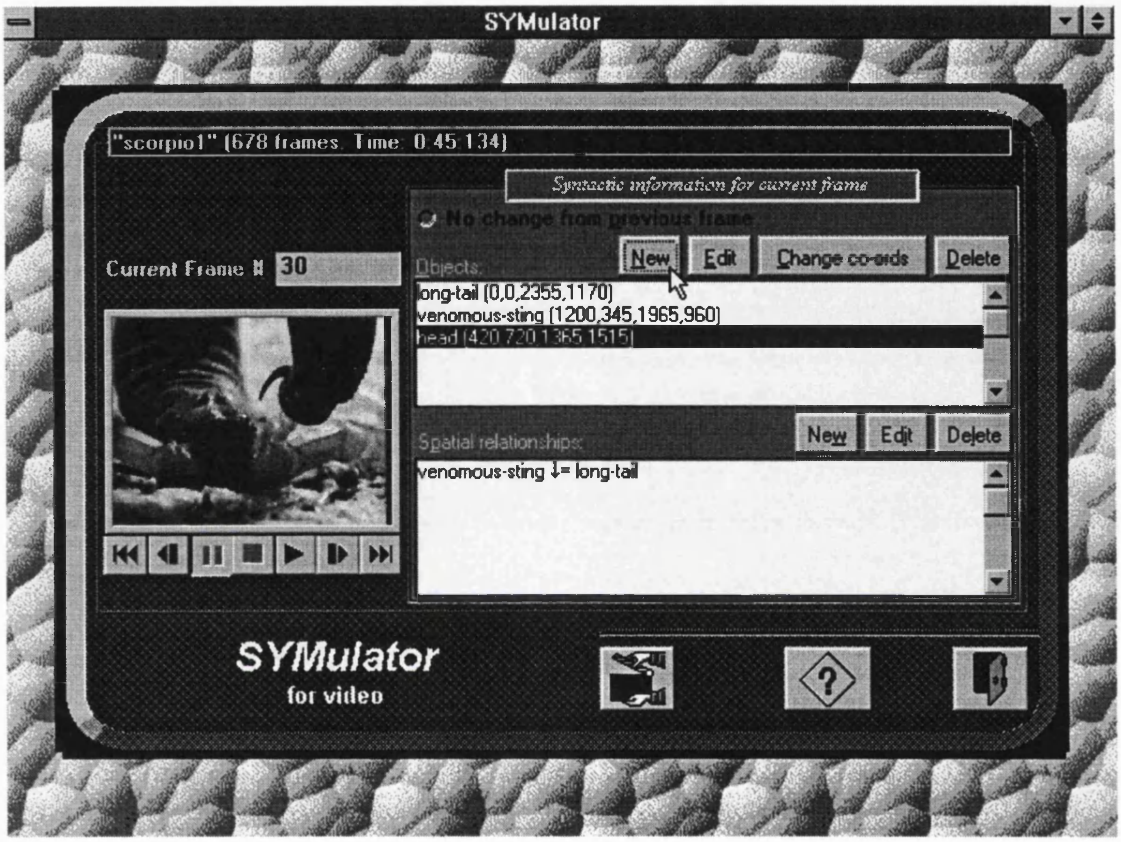
Figure 3.7 provides an example annotated spatial network diagram for a video frame that may feature in the Burgle event represented in the TOEA matrix in Section 3.2. The diagram indicates that the burglar is to the left of and in front of the cat and the television, and the cat is above and touching (i.e. on top of) the television. In addition, there is a plant behind and to the right of the television. The plant is an incidental object that was not represented on the TOEA matrix in Section 3.2.

3.5 STAGE 5: IMPLEMENT SYNTACTIC M-FRAMES

Once the annotated spatial network diagrams have been constructed, the SYMs may be developed. The diagrams should be used as the basis for developing the SYMs, together with the addition of on-screen co-ordinates for objects modelled within the diagrams. Video frames which refer to the same annotated spatial network diagram will only require separate associated SYMs if the on-screen co-ordinates of the objects modelled within the diagram change during those video frames.

While the SYMs may be created directly by the developer, the use of a front-end software tool proves practical for SYM implementation. An example of such a tool is the SYMulator (Figure 3.8), developed in Asymetrix Multimedia ToolBook 4.0. The SYMulator requires that video clips have already been added to the multimedia resource through the use of the Clip Manager. The SYMulator was developed to assist with the implementation of ARISTOTLE, an interactive instructional MMIS that is discussed in Chapter Four.

The SYMulator enables the developer to open a video clip from the multimedia resource, and then step through it frame-by-frame while adding the object co-ordinates and spatial relationships to each frame. The SYMs for a given video clip are stored in a Borland Paradox database, which has the same name as the associated clip. The database consists of three fields: (1) FrameNo, (2) Objects, and (3) SpatialRels. The FrameNo field is of type number, while the other two fields are of type memo (a text field of unlimited size). Each SYM is stored as a record within the database. In cases where a *SYM* is the same as a previous *SYM*, no record is stored. On determining the absence of a particular *SYM*, the MMIS using the database may then use a preceding *SYM*.



"scorpio1" (678 frames Time 0 45 134)

Current Frame # 30

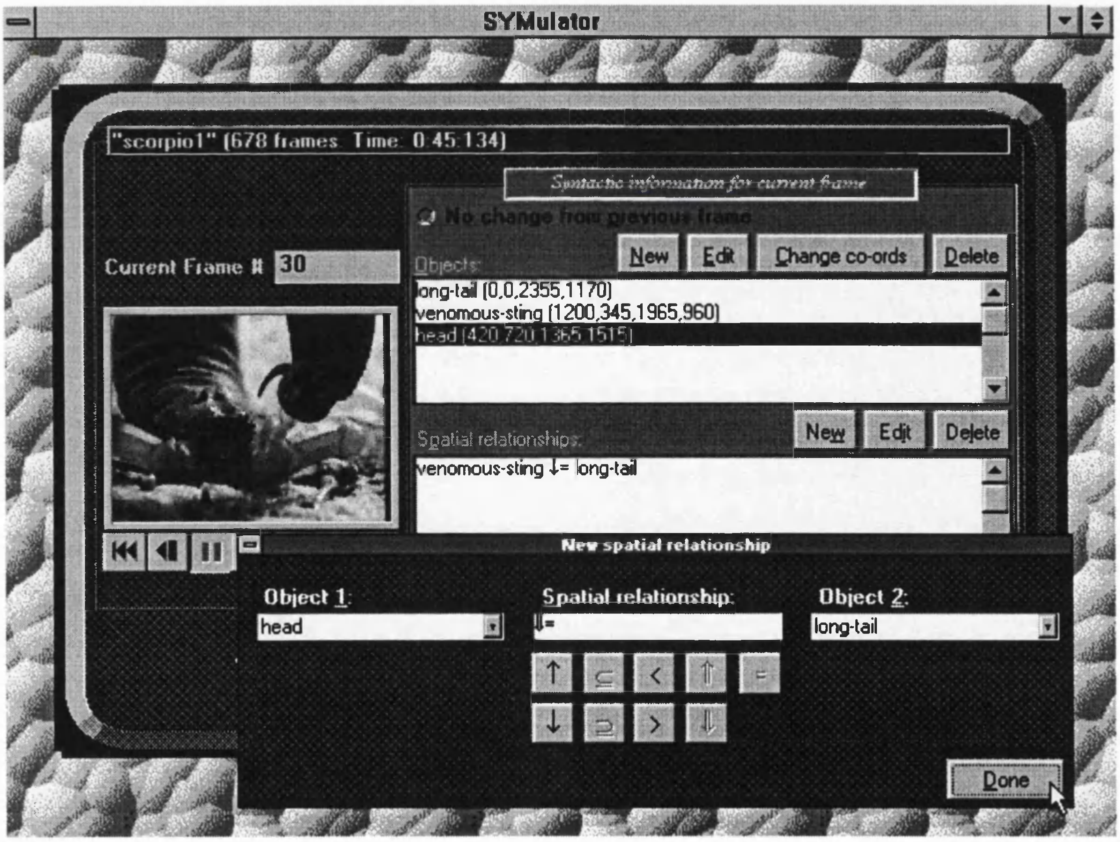
long-tail (0,0,2355,1170)
venomous-sting (1200,345,1965,960)
head (420,720,1365,1515)

venomous-sting ↓= long-tail

Figure 3.8 The SYMulator.

Every time the user creates a new object, they are given the option to attach on-screen co-ordinates to that object. In these cases, they may mark out on the video window, using the cursor, the location of the object. The co-ordinates used are page units, which are more accurate than screen-relative pixels. The conversion between the two differs depending on the video driver used; however, for a standard VGA screen (i.e. 640×480 pixels) there are 15 page units for every pixel on both the horizontal and the vertical planes. Each object together with its on-screen co-ordinates is stored on a separate textline within the Objects database field.

Spatial relationships may only be created between objects that have already been defined within the SYM. Figure 3.9 shows a spatial relationship being created in the SYMulator. Based on the existing symbols that have been used, the system disables those symbol buttons that would result in an invalid spatial relationship being constructed, e.g. it is not valid to use a relation together with its inverse. Each spatial relationship is stored on a separate textline within the SpatialRels database field.



SYMulator

"scorpio1" (678 frames Time: 0:45:134)

Syntactic information for current frame

No change from previous frame

Current Frame # 30

Objects:

long-tail (0,0,2355,1170)
venomous-sting (1200,345,1965,960)
head (420,720,1365,1515)

Spatial relationships:

venomous-sting ↓= long-tail

New spatial relationship

Object 1: head

Spatial relationship: ⇓=

Object 2: long-tail

Figure 3.9 Creating a new spatial relationship using the SYMulator.

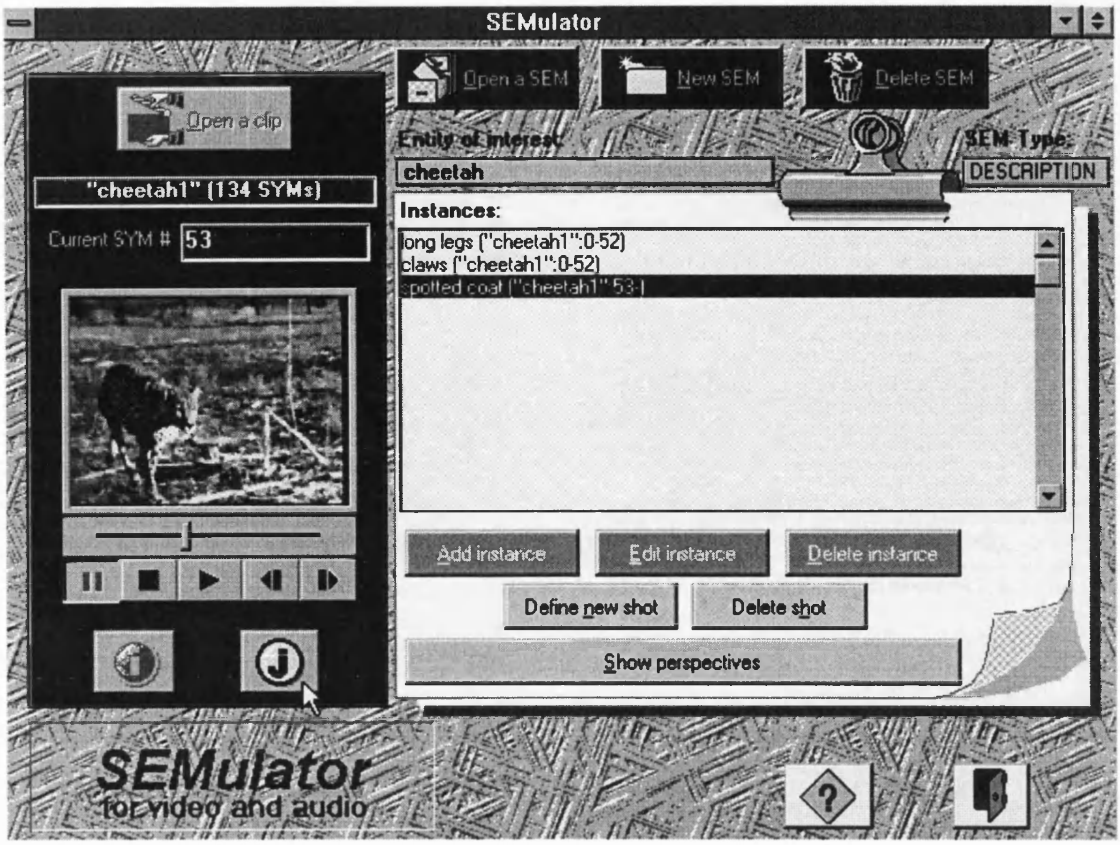
3.6 STAGE 6: IMPLEMENT SEMANTIC M-FRAMES

The *SEMs* are implemented after the *SYMs*. Their construction is based on the Description matrix and the TOEA matrices of Stages 1 and 2 respectively. The Description matrix is used to form the Description *SEMs* for each entity of interest. The description perspectives and instances map onto the perspectives and instances of the Description *SEMs*. The TOEA matrices are used as the basis for the Events and Actions *SEMs*. The events detailed on the matrices are grouped together according to the viewpoint taken of the domain. This grouping forms the basis of the Events *SEMs* perspectives, while the events become the Events *SEMs* instances. The actions that make up each event are then used to implement the Actions *SEMs*.

Based on the video and audio clips, suitable shots are then defined within the *SEMs* through the addition of shot references. Each reference refers to the *SYMs* associated with each shot.

While the *SEMs* may also be created directly by the developer, the use of a front-end software tool again proves practical. An example of such a tool is the SEMulator (Figure 3.10), developed in Asymetrix Multimedia ToolBook 4.0. As is the case with the Clip Manager and the SYMulator, the SEMulator was developed to speed the process of creating the *SEMs* that are used by ARISTOTLE (discussed in Chapter Four).

The SEMulator enables the developer to create the perspectives and instances for Description, Events, and Actions *SEMs*. Shots are physically defined by opening a video or audio clip and then indicating the corresponding *i* and *j SYMs* while the clip is playing or is paused.



"cheetah1" (134 SYMs)

Current SYM # 53

Entity of interest: cheetah

SEM Type: DESCRIPTION

Instances:

long legs ("cheetah1":0-52)
claws ("cheetah1":0-52)
spotted coat ("cheetah1":53-)

Figure 3.10 The SEMulator.

The construction of an Actions *SEM* for an entity of interest is based on an Events *SEM* for the same entity of interest, and the SEMulator embodies this in two ways. First, while the perspectives for Description and Events *SEM*s may be decided upon freely by the developer, the SEMulator only allows those perspectives to be added to an Actions *SEM* which already exist as instances in the corresponding Events *SEM*. Thus, the SEMulator constructs a list of events that have been defined in the corresponding Events *SEM*, and then invites the developer to choose perspectives from this list. Second, since actions serve to split up an event, the SEMulator ensures that a shot that is defined for an action is a valid sub-shot of one of the shots that was defined for the corresponding event (in the Events *SEM*).

Description, Events, and Actions SEMs are stored as Borland Paradox databases. The database consists of two fields: (1) Perspective (of type alphanumeric), and (2) Instances (of type memo). Each perspective and its associated instances are stored as separate records within the database. Each instance, together with its associated shots, is stored on a separate textline within the Instances memo field.

3.7 STAGE 7: IMPLEMENT MULTIMEDIA SUPPORT ENVIRONMENT

With the full-scale semantic content-based model fully implemented, it is now necessary to implement the multimedia support environment for the model. The multimedia support environment consists of the areas of the system that support the full-scale semantic content-based model. Consequently, the type of multimedia support environment that is implemented is dependent upon the type of MMIS being developed. For example, if an interactive instructional MMIS were being developed, then this stage would be concerned with the development of the domain, tutor and student modules of the architecture; whereas if a multimedia expert system were being developed, then this stage would involve the development of the knowledge base and the inferencing processes.

The functionality of the multimedia support environment is also dependent upon the manner in which the SYMs and SEMs have been implemented. For example, if the SYMulator and SEMulator were used, then the multimedia support environment would need to use the routines provided by the Borland Paradox Engine DLL (Dynamic Link

Library) in order to retrieve, search, and process the information contained within the various databases.

3.8 SUMMARY

This chapter has contributed a method for developing an interactive MMIS that uses the full-scale semantic content-based model proposed in the previous chapter. The method consists of seven stages: (1) construct Description matrix for entities of interest to the system; (2) construct Temporal Objects/Events and Actions (TOEA) matrices; (3) collect raw video and audio footage; (4) construct annotated spatial network diagrams for video clips; (5) implement SYMs; (6) implement SEMs; and (7) implement multimedia support environment. The chapter also discussed three front-end software tools that assist with the development process: the Clip Manager assists the user in the management of clips within the multimedia resource, while the SYMulator and SEMulator facilitate the creation of SYMs and SEMs respectively.

The following chapter demonstrates the use of the method in the development of ARISTOTLE, an interactive instructional MMIS. The chapter presents the architecture of the system, and discusses its behaviour in relation to the architecture.

Chapter Four

USING THE METHOD IN THE DEVELOPMENT OF ARISTOTLE, AN INTERACTIVE INSTRUCTIONAL MULTIMEDIA INFORMATION SYSTEM

"I pass with relief from the tossing sea of Cause and Theory to the firm ground of Result and Fact."

— Sir Winston Churchill

Chapter Three presented a method for developing interactive MMISs that use the full-scale semantic content-based model proposed in Chapter Two. The method provides a means of breaking down the development of such an interactive MMIS into a number of stages, running from the early planning and design of the model, through to the model's implementation, and ending with the implementation of the multimedia support environment that uses the model. It was explained in the previous chapter that this final stage was dependent upon the type of interactive MMIS being developed.

In this chapter, the method is applied to the development of ARISTOTLE, a prototype interactive instructional MMIS that fully utilises the full-scale semantic content-based model proposed in Chapter Two. Interactive instructional MMISs place stringent demands on the use of video and audio (Agius, 1996; Angelides and Demosthenous, 1996). For pedagogic purposes, they require comprehensive and structured content-based information and links between segments of related content, whether this be audio, video, or information (Agius and Angelides, 1997b). Such requirements provide a sound basis for demonstrating the practical use of the full-scale semantic content-based model and its seven semantic aspects.

ARISTOTLE was developed with Asymetrix Multimedia ToolBook 4.0 under Microsoft Windows. The system tutors a knowledge of basic zoology to young school children, hence it is named after the famous Greek philosopher who was one of the earliest writers on zoology. ARISTOTLE has been designed to fit within the framework of the National Science Curriculum for England and Wales (Department for Education, 1995). The Curriculum encourages the implementation and use of multimedia because it is useful for teaching about visual and aural phenomena, such as movement, observable differences between living things, growth, finding different animals in different habitats, and distinguishing variation in the noises of different animals.

The following section discusses the development of ARISTOTLE as it took place within each of the stages of the method. Then, the chapter explains the functionality and behaviour of the system in relation to its architecture.

4.1 ARISTOTLE'S DEVELOPMENT

This section discusses ARISTOTLE's development as it took place within the stages of the method presented in Chapter Two. Stage 7 of the method is a large stage which, in this case, is specific to an interactive instructional MMIS. It thus serves as a method by which interactive instructional MMISs that use the full-scale semantic content-based model may be developed.

4.1.1 Stage 1: Constructing the Description matrix for entities of interest to ARISTOTLE

ARISTOTLE's domain is that of zoology, and therefore the entities of interest to the system are animals. To create a system that would be able to demonstrate the full-scale semantic content-based model sufficiently, three animals were chosen and grouped into classes. Animals were split into vertebrates and invertebrates. Vertebrates were further divided into mammals and reptiles; arthropods were chosen as a specific type of invertebrate. Figure 4.1 details the entities of interest to ARISTOTLE.

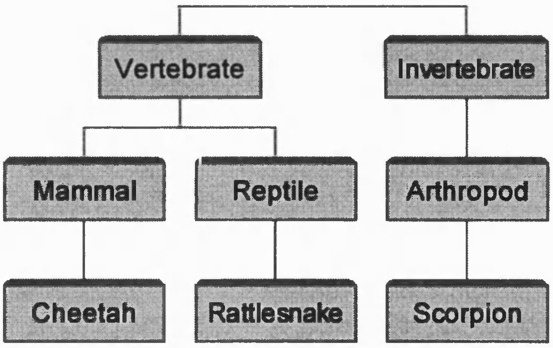
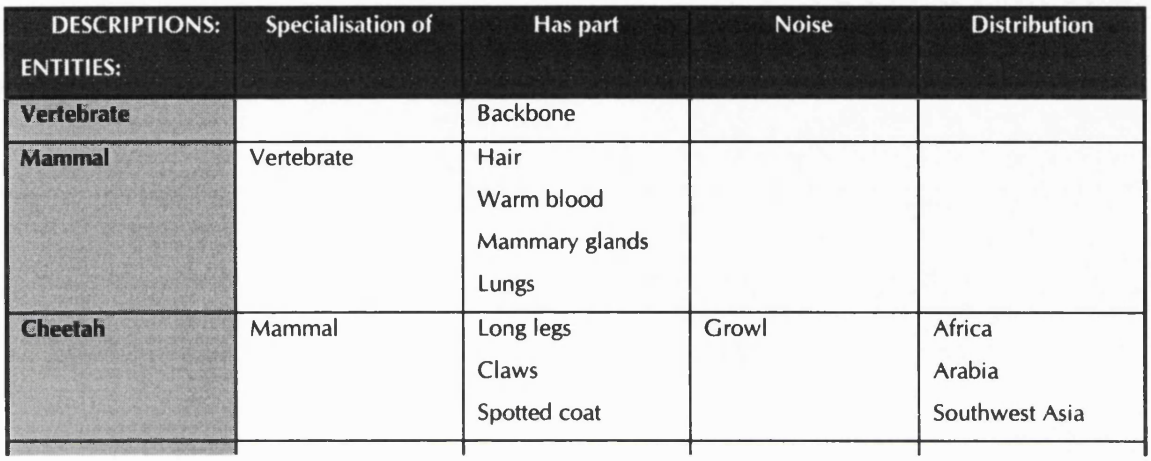


Figure 4.1 The entities of interest to ARISTOTLE.

The animals and animal classes were then described on a Description matrix, according to four description perspectives: Specialisation of, Has part, Noise, and Distribution. Figure 4.2 shows a portion of this Description matrix. It provides descriptions of the Vertebrate, Mammal and Cheetah entities of interest.



DESCRIPTIONS: ENTITIES:	Specialisation of	Has part	Noise	Distribution
Vertebrate		Backbone		
Mammal	Vertebrate	Hair Warm blood Mammary glands Lungs		
Cheetah	Mammal	Long legs Claws Spotted coat	Growl	Africa Arabia Southwest Asia

Figure 4.2 A portion of the Description matrix for ARISTOTLE.

4.1.2 Stage 2: Constructing the TOEA matrices

The next stage of ARISTOTLE's development was to decide upon animal events and actions that would be used in teaching. A TOEA matrix was constructed for the Mammal, Cheetah, Reptile, Rattlesnake and Scorpion entities of interest. The events for these entities centred on what the animal or animal class was able to do (e.g. hunting), while the actions broke down the events into their constituent activities (e.g. observing prey, catching prey). Figure 4.3 shows the TOEA matrix for the Cheetah entity of interest.

TEMPORAL EVENT RELATIONSHIPS:		1_A					1_A	
EVENTS:		Hunt					Kill	
ACTIONS:		Observing (a)	Testing for slow (a)	Catching (a)	Suffocating (a)	Chasing (a)	Catching (a)	Suffocating (a)
OBJECTS:	Cheetah	1_a	1_a	1_a	1_a	1_a	1_a	1_a
	Prey	2_a	2_a	2_a	2_a	2_a	2_a	2_a
TEMPORAL ACTION RELATIONSHIPS:	a	1_A	1_A	3_A	4_A	2_A	3_A	4_A

Figure 4.3 The TOEA matrix for the Cheetah entity of interest in ARISTOTLE.

4.1.3 Stage 3: Collecting the raw video and audio footage

The TOEA matrices were used to guide the filming process. Initially, a day was spent filming the animals at London Zoo in Regent's Park, NW1. However, the generally unpredictable nature of animals (compared with human actors) meant that not all the required footage could be obtained. For example, the animals frequently failed to perform their characteristic noises. In the latter case, even when animal noises *were* recorded, playback of the footage revealed that the noise of the crowds at the Zoo had drowned out the recorded animal sounds. It also proved impossible to obtain the events footage required, e.g. the Cheetah TOEA matrix dictated that footage was required of a cheetah hunting, however the cheetahs at the Zoo do not have to hunt for their food.

In order to overcome these problems, a number of wildlife programmes were recorded onto VHS tape as they were broadcast on television. Since these programmes included professionally shot footage of animals in their natural habitats, they featured many of the animals' characteristic behaviours. In addition, because they were broadcast in NICAM stereo, the quality of the animal noises proved to be much better than that obtained from London Zoo.

Suitable video and audio segments from the video tapes were then captured into Microsoft AVI (Audio/Video Interleave) format using Microsoft VidCap, and edited into clips using Microsoft VidEdit (both applications are components of Microsoft Video for Windows). In cases where the video segment included commentary by the programme narrator that met the needs of the system, or uninterrupted sounds of nature, this was kept as part of the segment. However, captured video segments that included a nonsensical audio component (e.g. because the footage captured had been edited in such a way that the commentary no longer made sense), had this audio component replaced with suitable sections of music.

All of the clips were then added to a single multimedia resource using the Clip Manager front-end software tool (discussed in Chapter Three).

4.1.4 Stage 4: Constructing the annotated spatial network diagrams for the video clips

The annotated spatial network diagrams were then constructed in order to represent the spatial relationships of entities of interest within the video frames of the clips. Here the Description and TOEA matrices helped identify which objects of the many objects featured within the video frames were of interest to the system. Additional objects that would be useful to ARISTOTLE during the teaching-learning interaction were also included. It was found that, on average, spatial relationships did not change that much between video frames, and were often uniform for up to 100 video frames.

Figure 4.4 shows the annotated spatial network diagram that was drawn for frame 0 of the "seal1" clip within ARISTOTLE. This clip is used when teaching about vertebrates. The frame therefore models the spatial relationships existing between the backbone (the distinguishing part of the vertebrate), the back and the head. While the latter two objects were not included on the Description matrix, their inclusion here will enable the student to be taught the location of the backbone in vertebrates in relation to other parts of the body, e.g. the backbone is inside the back.

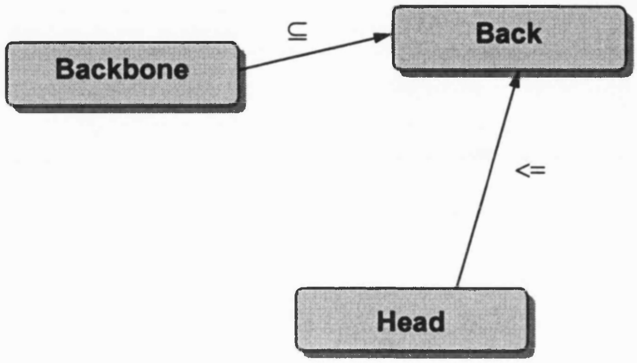
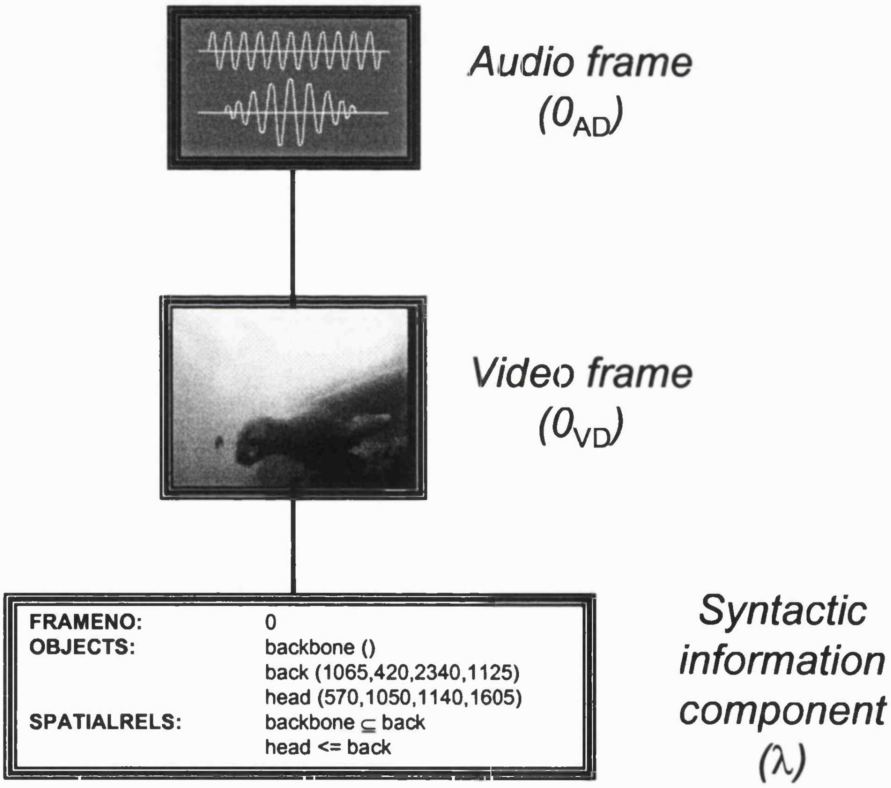


Figure 4.4 An annotated spatial network diagram drawn for frame 0 of the "seal1" clip within ARISTOTLE.

4.1.5 Stage 5: Implementing the SYMs

The annotated spatial network diagrams were then used in conjunction with the SYMulator front-end software tool (discussed in Chapter Three) to create the syntactic m-frames. Adding the spatial relationships to the SYMs was a relatively quick process, in this particular case taking only a few hours, since it merely involved transforming the detail of the diagrams into the SYMs. However, it took one week to add the on-screen co-ordinates to all of the SYMs for all of the clips. While the spatial relationships did not differ frequently within the clips, the co-ordinates of the objects rarely stayed the same for more than two or three video frames. On average, each SYM modelled five or six objects and six or seven spatial relationships. The number of unique SYMs needed for a particular video clip differed greatly depending on the length of the clip, and the speed of movement within the clip. For example, one of the clips shows video footage of a cheetah running which required more unique *SYMs* than the clip which shows footage of a rattlesnake slithering slowly through grass.

Figure 4.5 provides a conceptual representation of one of ARISTOTLE's *SYMs*. The *SYM* is representing a video and audio frame (160×120) from a clip depicting a seal. It represents the final *SYM* that is based on the spatial relationships depicted in Figure 4.4.



FRAMENO:	0
OBJECTS:	backbone () back (1065,420,2340,1125) head (570,1050,1140,1605)
SPATIALRELS:	backbone ⊆ back head <= back

Figure 4.5 Conceptual representation of a *SYM* from ARISTOTLE.

4.1.6 Stage 6: Implementing the *SEMs*

The *SEMs* used in ARISTOTLE were constructed according to the Description and TOEA matrices for each entity of interest. They were organised hierarchically, via a SPECIALISATION OF perspective.

The SEMulator front-end software tool (discussed in Chapter Three) was used to create the *SEMs*. First the perspectives and instances were created for each of the Description, Events, and Actions *SEMs*. Then, appropriate shots were defined for the various instances within these *SEMs*. Figure 4.6 shows the implemented Description, Events, and Actions *SEMs* for the Cheetah entity of interest.

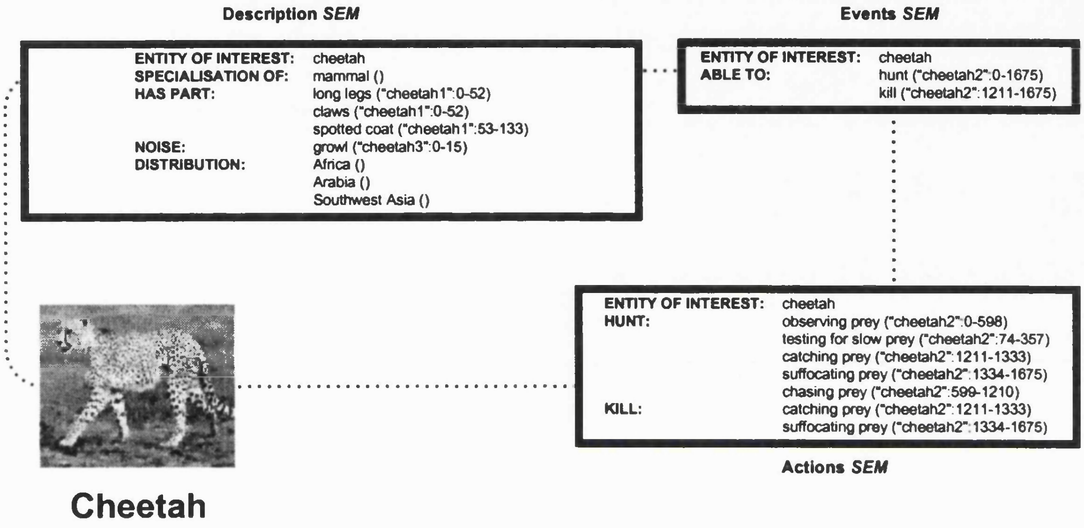

Description *SEM*
ENTITY OF INTEREST: cheetah
SPECIALISATION OF: mammal ()
HAS PART: long legs ("cheetah1":0-52)
claws ("cheetah1":0-52)
spotted coat ("cheetah1":53-133)
NOISE: growl ("cheetah3":0-15)
DISTRIBUTION: Africa ()
Arabia ()
Southwest Asia ()
Events *SEM*
ENTITY OF INTEREST: cheetah
ABLE TO: hunt ("cheetah2":0-1675)
kill ("cheetah2":1211-1675)
ENTITY OF INTEREST: cheetah
HUNT: observing prey ("cheetah2":0-598)
testing for slow prey ("cheetah2":74-357)
catching prey ("cheetah2":1211-1333)
suffocating prey ("cheetah2":1334-1675)
chasing prey ("cheetah2":599-1210)
KILL: catching prey ("cheetah2":1211-1333)
suffocating prey ("cheetah2":1334-1675)
Actions *SEM*
Cheetah

Figure 4.6 Conceptual representation of the Cheetah Description, Events, and Actions *SEMs* from ARISTOTLE.

4.1.7 Stage 7: Developing ARISTOTLE's multimedia support environment

Development of the full-scale semantic content-based model is one part of a larger development process for an interactive MMIS. In the case of an interactive instructional MMIS that uses the full-scale semantic content-based model, such as ARISTOTLE, a number of additional stages are required after the model has been developed, which are concerned with the implementation of the various modules of the architecture.

Interactive instructional MMISs are based on an architecture that incorporates three modules (Angelides, 1995; Woolf and Hall, 1995; Siemer and Angelides, 1996; Woolf, 1996; Siemer and Angelides, 1997a; Siemer and Angelides, 1997b; Siemer and Angelides, 1997c): a *domain module*, which contains the knowledge of the domain to be taught to the student-user; a *tutor module*, which contains the pedagogic strategies that guide the teaching of the student-user; and a *student module*, which infers and models the status of the student-user. These three modules together make up the multimedia support environment of an interactive instructional MMIS. ARISTOTLE uses such an architecture, depicted in Figure 4.7. The teaching techniques that are used in the architecture are adaptations of common practice in instructional systems.

All of ARISTOTLE's SEMs (with the exception of the remedial strategy and teaching strategy SEMs) are stored as Borland Paradox databases, that each consist of two fields: (1) Perspective (of type alphanumeric), and (2) Instances (of type memo). Each perspective and its associated instances are stored as separate records within the database. Each instance is stored on a separate textline within the Instances memo field. The domain SEMs and the SYMs were developed using the SEMulator and SYMulator front-end systems respectively. All of the other SEMs, e.g. the remedial SEMs and the teaching goal SEMs, were created directly as Borland Paradox databases, using Borland Paradox 5.0 for Windows.

The remedial strategy and teaching strategy SEMs are procedural in nature and thus are stored as OpenScript code (OpenScript is the programming language of Multimedia ToolBook). OpenScript permits the run-time compilation and execution of code text; thus the code could have been stored as text within the Paradox databases, and then compiled and executed each time the remedial and teaching strategies were used by ARISTOTLE. However, this feature of OpenScript is only practical when used with a few lines of code text. The remedial and teaching strategy routines are very large in size and thus their run-time compilation and execution would have increased greatly the response time of the system.

The following sections discuss the development of ARISTOTLE's domain, tutor and student modules.

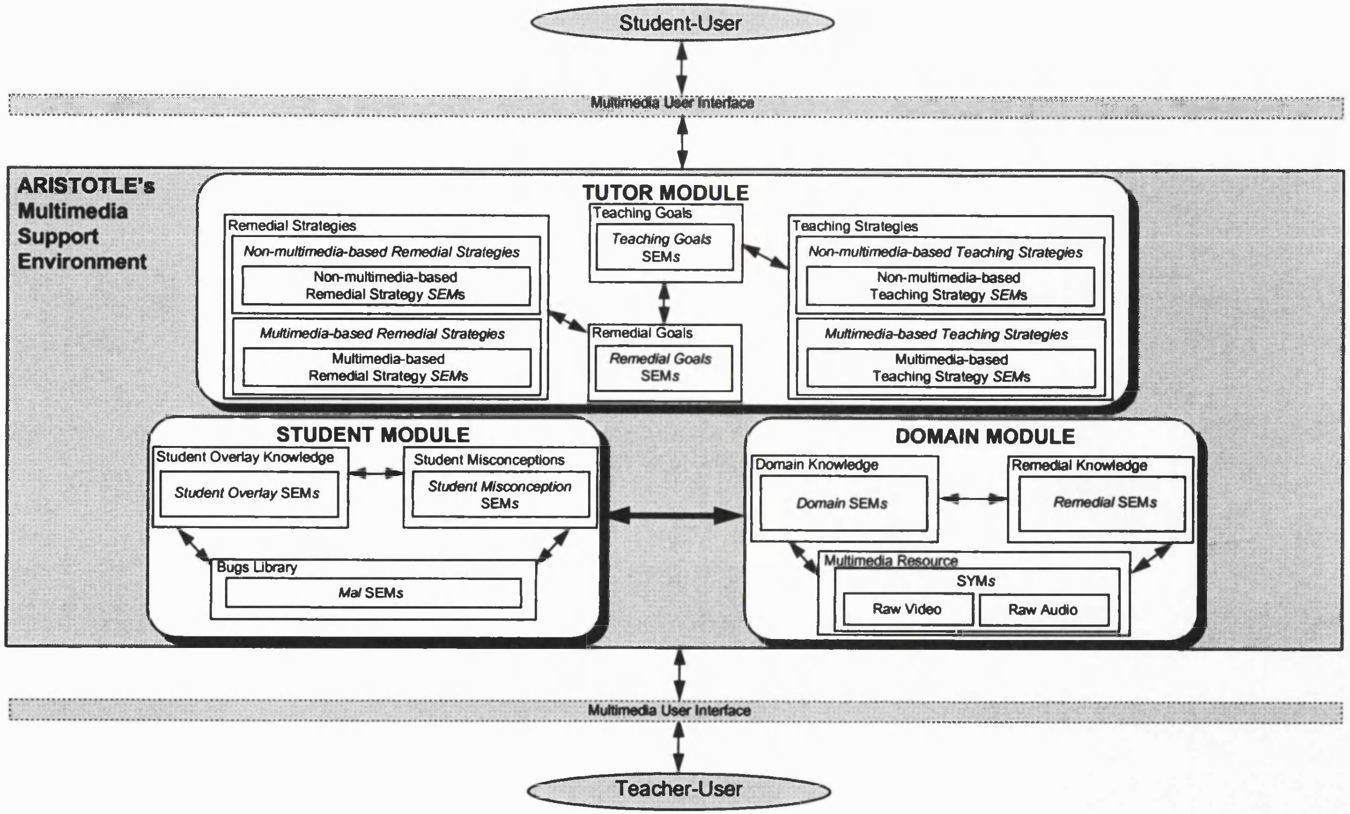

Student-User
Multimedia User Interface
ARISTOTLE's Multimedia Support Environment
TUTOR MODULE
Remedial Strategies
Non-multimedia-based Remedial Strategies
Non-multimedia-based Remedial Strategy SEMs
Multimedia-based Remedial Strategies
Multimedia-based Remedial Strategy SEMs
Teaching Goals
Teaching Goals SEMs
Remedial Goals
Remedial Goals SEMs
Teaching Strategies
Non-multimedia-based Teaching Strategies
Non-multimedia-based Teaching Strategy SEMs
Multimedia-based Teaching Strategies
Multimedia-based Teaching Strategy SEMs
STUDENT MODULE
Student Overlay Knowledge
Student Overlay SEMs
Student Misconceptions
Student Misconception SEMs
Bugs Library
Mal SEMs
DOMAIN MODULE
Domain Knowledge
Domain SEMs
Remedial Knowledge
Remedial SEMs
Multimedia Resource
SYMs
Raw Video
Raw Audio
Multimedia User Interface
Teacher-User

Figure 4.7 The architecture of ARISTOTLE.

Implementing ARISTOTLE's domain module

The first module to be developed in an interactive instructional MMIS is the domain module. The domain module may be considered the fundamental module of the architecture since it provides the knowledge to be imparted to the student-user, and the structure of that knowledge. ARISTOTLE's domain module consists of three components: a multimedia resource, domain knowledge, and remedial knowledge.

The *multimedia resource* consists of the SYMs, and their associated raw video and audio, as developed using the Clip Manager and SYMulator in Stages 3 and 5 of the method.

The *domain knowledge* consists of the SEMs that were developed in Stage 6. In order for ARISTOTLE to be able to describe the unessential content of a shot being presented, an ANNOTATION perspective was added to each of the Description SEMs where the associated entity of interest used video or audio footage. Figure 4.8 provides a conceptual representation of the Cheetah Description, Events and Actions SEMs given in Figure 4.6 with the additional ANNOTATION perspective. The instances within the ANNOTATION perspective provide incidental information about the content of the various shots that have been used within the SEMs. These instances are used to enable ARISTOTLE to provide an introductory textual description to a video shot that does not include any of the answers expected from the student-user. The use of an ANNOTATION perspective in this way marks out clearly which information within the Description, Events, and Actions SEMs is unrelated to the teaching goals of the system, i.e. the annotations, while also keeping all of this information together.

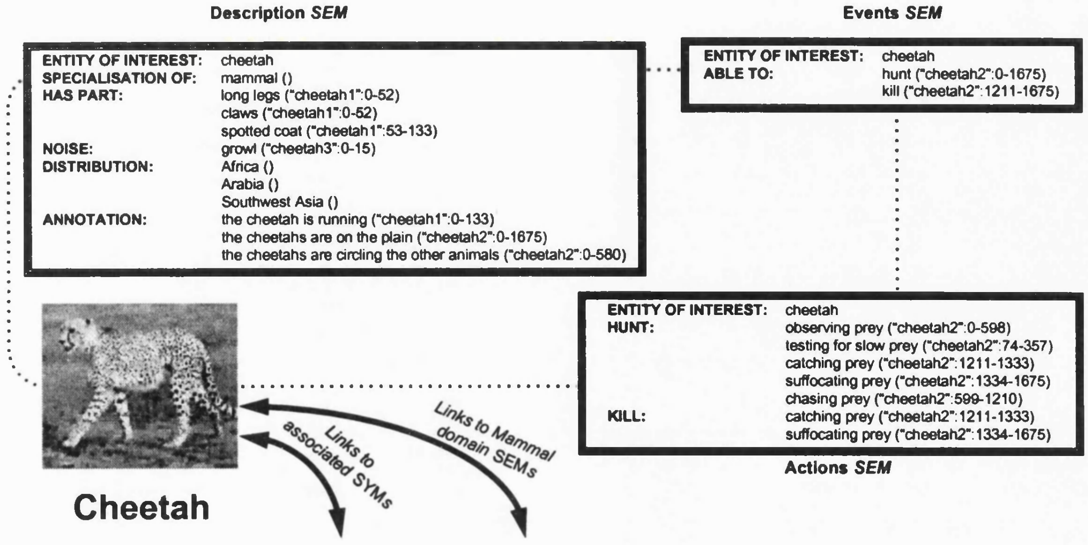
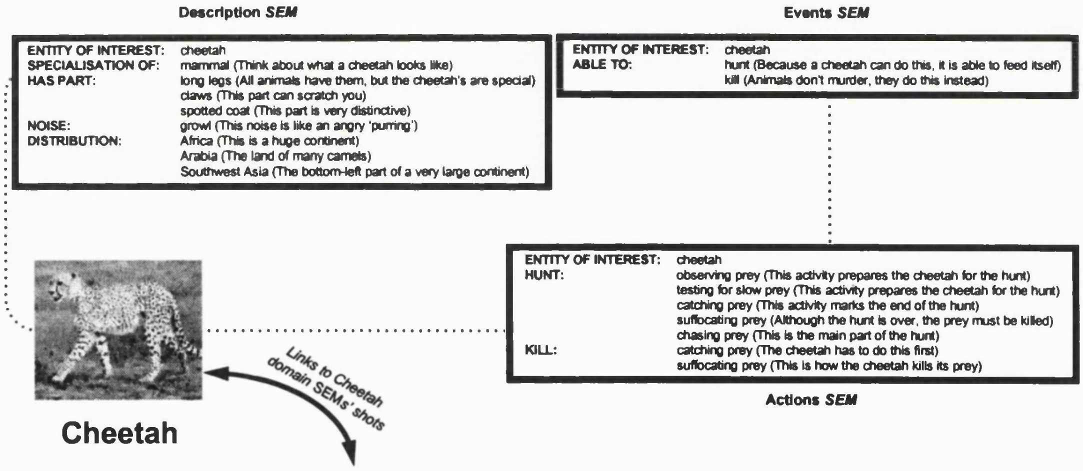

Description SEM
ENTITY OF INTEREST: cheetah
SPECIALISATION OF: mammal ()
HAS PART: long legs ("cheetah1":0-52)
claws ("cheetah1":0-52)
spotted coat ("cheetah1":53-133)
NOISE: growl ("cheetah3":0-15)
DISTRIBUTION: Africa ()
Arabia ()
Southwest Asia ()
ANNOTATION: the cheetah is running ("cheetah1":0-133)
the cheetahs are on the plain ("cheetah2":0-1675)
the cheetahs are circling the other animals ("cheetah2":0-580)
Events SEM
ENTITY OF INTEREST: cheetah
ABLE TO: hunt ("cheetah2":0-1675)
kill ("cheetah2":1211-1675)
ENTITY OF INTEREST: cheetah
HUNT: observing prey ("cheetah2":0-598)
testing for slow prey ("cheetah2":74-357)
catching prey ("cheetah2":1211-1333)
suffocating prey ("cheetah2":1334-1675)
chasing prey ("cheetah2":599-1210)
KILL: catching prey ("cheetah2":1211-1333)
suffocating prey ("cheetah2":1334-1675)
Actions SEM
Cheetah
Links to associated SYMs
Links to Mammal domain SEMs

Figure 4.8 Conceptual representation of ARISTOTLE's Cheetah domain *SEMs* with the annotations.

The links to the *SYMs* that are associated with the shots used within the Cheetah domain *SEMs* are dynamic and are established during the course of interaction. These links are established by linking the *SYM* database for the shot currently being delivered with the entity of interest at hand through the manipulation of a database alias. The Borland Paradox Engine uses an alias to refer to a database table, as a substitute for the table's physical name. Thus, when a new *SYM* database is required, the link to it is established by changing the physical name associated with the *SYM* database alias, while leaving the alias name itself intact.

Links to the domain *SEMs* that are one level up in the hierarchy (e.g. the Mammal domain *SEMs* in the case of the Cheetah represented in Figure 4.8) are also established dynamically. Such links are set up by the creation of three database aliases, which serve as links to the Description, Events, and Actions *SEMs* (i.e. tables) for the required entity of interest.

The *remedial knowledge* stores the information that is used by the tutor module to remedy the student-user when they do not provide the correct answer to a question and also when the student-user requests assistance. ARISTOTLE's remedial knowledge is a hierarchy of remedial *SEMs*, which mirrors the domain *SEMs* hierarchy. However, the remedial *SEMs* are oriented towards the provision of remediation for the student-user. The remedial *SEMs* thus model information intended to guide the student-user towards the remedial goals, and thus the teaching goals, rather than reflecting perspectives on the multimedia content as the domain *SEMs* do. Figure 4.9 shows the Cheetah remedial *SEMs* in ARISTOTLE's remedial knowledge. ARISTOTLE's remedial *SEMs* store textual information that is intended to assist the student-user with their misconceptions.



ENTITY OF INTEREST: cheetah
SPECIALISATION OF: mammal (Think about what a cheetah looks like)
HAS PART: long legs (All animals have them, but the cheetah's are special)
claws (This part can scratch you)
spotted coat (This part is very distinctive)
NOISE: growl (This noise is like an angry 'purring')
DISTRIBUTION: Africa (This is a huge continent)
Arabia (The land of many camels)
Southwest Asia (The bottom-left part of a very large continent)

ENTITY OF INTEREST: cheetah
ABLE TO: hunt (Because a cheetah can do this, it is able to feed itself)
kill (Animals don't murder, they do this instead)

ENTITY OF INTEREST: cheetah
HUNT: observing prey (This activity prepares the cheetah for the hunt)
testing for slow prey (This activity prepares the cheetah for the hunt)
catching prey (This activity marks the end of the hunt)
suffocating prey (Although the hunt is over, the prey must be killed)
chasing prey (This is the main part of the hunt)
KILL: catching prey (The cheetah has to do this first)
suffocating prey (This is how the cheetah kills its prey)

Figure 4.9 Conceptual representation of ARISTOTLE's Cheetah remedial *SEMs*.

Remedial *SEMs* for a particular entity of interest are used in conjunction with the domain *SEMs* for the same entity of interest. Thus, specific misconceptions may be illustrated to the student-user through the use of suitable video and audio shots found within the corresponding domain *SEMs*. This is indicated by the arrow labelled 'Links to Cheetah domain *SEMs*' shots' in Figure 4.9. These links are established as and when they are needed. For example, if the student-user is having difficulty identifying a cheetah's spotted coat, then ARISTOTLE's multimedia-based remedial strategies will gather the shot reference for *spotted coat* given in the Cheetah domain Description *SEM* (i.e."cheetah1":53-133) and use the shot to show the student-user the cheetah's spotted coat.

Once the domain *SEMs* and *SYMs* were implemented, various procedures for retrieving their information were developed. These procedures perform such tasks as retrieving all or some of the instances and shots for a given *SEM* perspective, returning a depth-first or breadth-first path through the domain knowledge (i.e. a list of animals), and retrieving all animals within the domain knowledge that have a particular instance value for a particular perspective.

Procedures for manipulating the information within the domain *SEMs* and *SYMs* were then implemented. These procedures use the procedures for retrieving information from the *SEMs* and *SYMs* that were previously implemented. They are concerned with various tasks, such as determining which objects within a given *SYM* match a given set of co-ordinates. This is used when determining what object a student-user has clicked on. The co-ordinates of each object within the *SYM* are checked against the given set of co-ordinates. A match occurs each time the given set of co-ordinates is determined to be located inside the co-ordinates of the object. In cases where there are no co-ordinates associated with an object in the *SYM*, the object that obscures this object is found through the spatial relationships, and then its co-ordinates are used as the co-ordinates of the hidden object.

The domain module also determines where a given object is located spatially with reference to other objects in a given *SYM*. This is achieved by first rearranging all of the spatial relationships within the *SYM* so that the given object appears first in each relationship. This involves swapping the position of the objects within the spatial relationship and then replacing the spatial relationship symbols with their inverses. Those relationships in which the given object does not appear are then excluded and the remaining relationships used to determine the object's spatial location, based on the symbols used in these relationships.

Another important task is that of determining the next group of actions within a given event, as actions occurring simultaneously must be taught about at the same time. All constituent actions of an event (that is, those actions which occur within the specific shot used to teach about the event) are ordered within one array, according to the *i* values of their associated shots. Thus, the next group of actions is determined by taking the first action in the array and then adding to the group those actions whose *j* values are not greater than the *j* value of the shot of the first action. These actions are thus those which occur simultaneously with the first action.

Similarly, the domain module also determines if a given action occurs before or after a given group of actions within an event, so that the tutor module may inform the student-user of this. If the *i* value of the given action is less than the *i* value of the first action within the given group of actions, then the action occurs before. If the *j* value of the given action is greater than the *j* value of the last action within the given group of actions, then the action occurs afterwards.

Similar techniques are also used to order and group events, and to determine whether events occur before or after other events.

The domain module also collects all the ANNOTATION instances whose associated shots intersect with a given shot. These shots may overlap considerably, and thus will intersect at various places. Five situations of two-shot intersection are illustrated in Figure 4.10. In all five situations, $A=[i_1, j_1]$ and $B=[i_2, j_2]$. In the case of Figure 4.10(a), $i_1 = i_2$, and $j_1 < j_2$; in Figure 4.10(b), $i_1 < i_2$, and $j_1 = j_2$; in Figure

4.10(c), $i_1 < i_2$, and $j_1 < j_2$; in Figure 4.10(d), $i_1 = i_2$ and $j_1 = j_2$; and in Figure 4.10(e), $i_1 < i_2$ and $j_1 > j_2$. Once the annotations are collected, they are ordered according to each associated shot's *i* value (as with the constituent actions of an event).

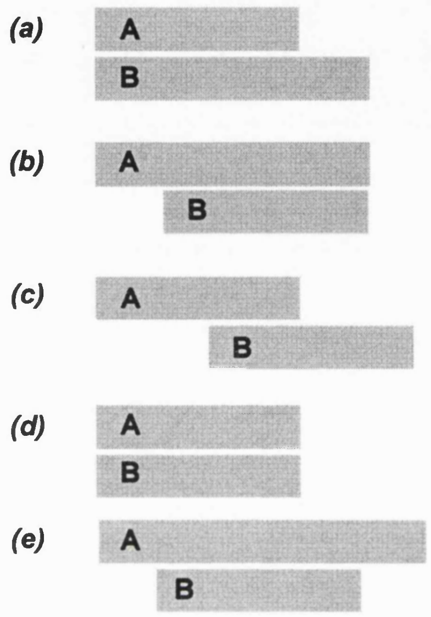


Figure 4.10 Intersecting shots: **(a)** A and B have the same starting frame, but different ending frames; **(b)** A and B have different starting frames, but share the same ending frame; **(c)** A and B have different starting and ending frames; **(d)** A and B share the same starting and ending frames; **(e)** A and B have different starting and ending frames, but B is a proper sub-shot of A.

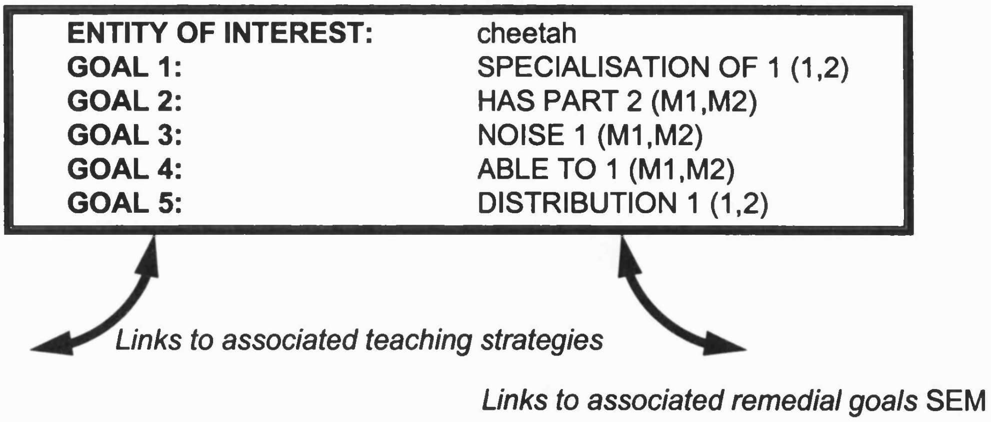
All of the implemented procedures within the domain module provide services to the tutor and student modules who use them during their processing.

Implementing ARISTOTLE's tutor module

The pedagogic processes of the interactive instructional MMIS are the next to be developed. ARISTOTLE's tutor module consists of three main components: teaching goals, remedial goals, and teaching strategies.

The *teaching goals* determine what the individual student-user should be instructed on. Goal attainment by the student-user is achieved through the use of associated teaching strategies. ARISTOTLE has one teaching goals *SEM* for every animal (i.e. entity of interest) within the domain knowledge.

Figure 4.11 shows ARISTOTLE's Cheetah teaching goals *SEM*. Each goal within the *SEM* consists of the name of the perspective that the goal is concerned with, the minimum number of instances that a student-user must name in order to satisfy the goal, and the teaching strategies which may be used to try to achieve the goal (the order in which the teaching strategies are to be used is determined by the student module). Where the perspective named for the goal is event-oriented (e.g. ABLE TO), the goal is achieved if the student-user names at least the number of events specified by the goal *and* all of the constituent actions of these events. The specified teaching strategies are used both for teaching about the events and their actions.



ENTITY OF INTEREST:	cheetah
GOAL 1:	SPECIALISATION OF 1 (1,2)
GOAL 2:	HAS PART 2 (M1,M2)
GOAL 3:	NOISE 1 (M1,M2)
GOAL 4:	ABLE TO 1 (M1,M2)
GOAL 5:	DISTRIBUTION 1 (1,2)

Links to associated teaching strategies
Links to associated remedial goals SEM

Figure 4.11 Conceptual representation of the Cheetah teaching goals *SEM* from ARISTOTLE.

The links to the associated teaching strategies are dynamic and are established during the course of interaction by calling the appropriate teaching strategy *SEM* based on the teaching strategy code (i.e. 1, 2, M1, or M2). The links to the associated remedial goals *SEM* are established when the tutor module first begins to teach about the entity of interest associated with the teaching goals *SEM*. The link is established through the use of a database alias to the remedial goals *SEM*.

Next, the *teaching strategies* were implemented. Teaching strategies determine how a student-user should be taught a particular subset of knowledge. A teaching strategy presents material that will allow student-users to master a particular teaching goal, and also evaluates the student-user's reaction to the instruction. ARISTOTLE uses teaching strategies that fall into one of two categories: (1) non-multimedia-based teaching strategies, and (2) multimedia-based teaching strategies. Both types of strategy are guided by the teaching goals of the system and the student-user's individual needs.

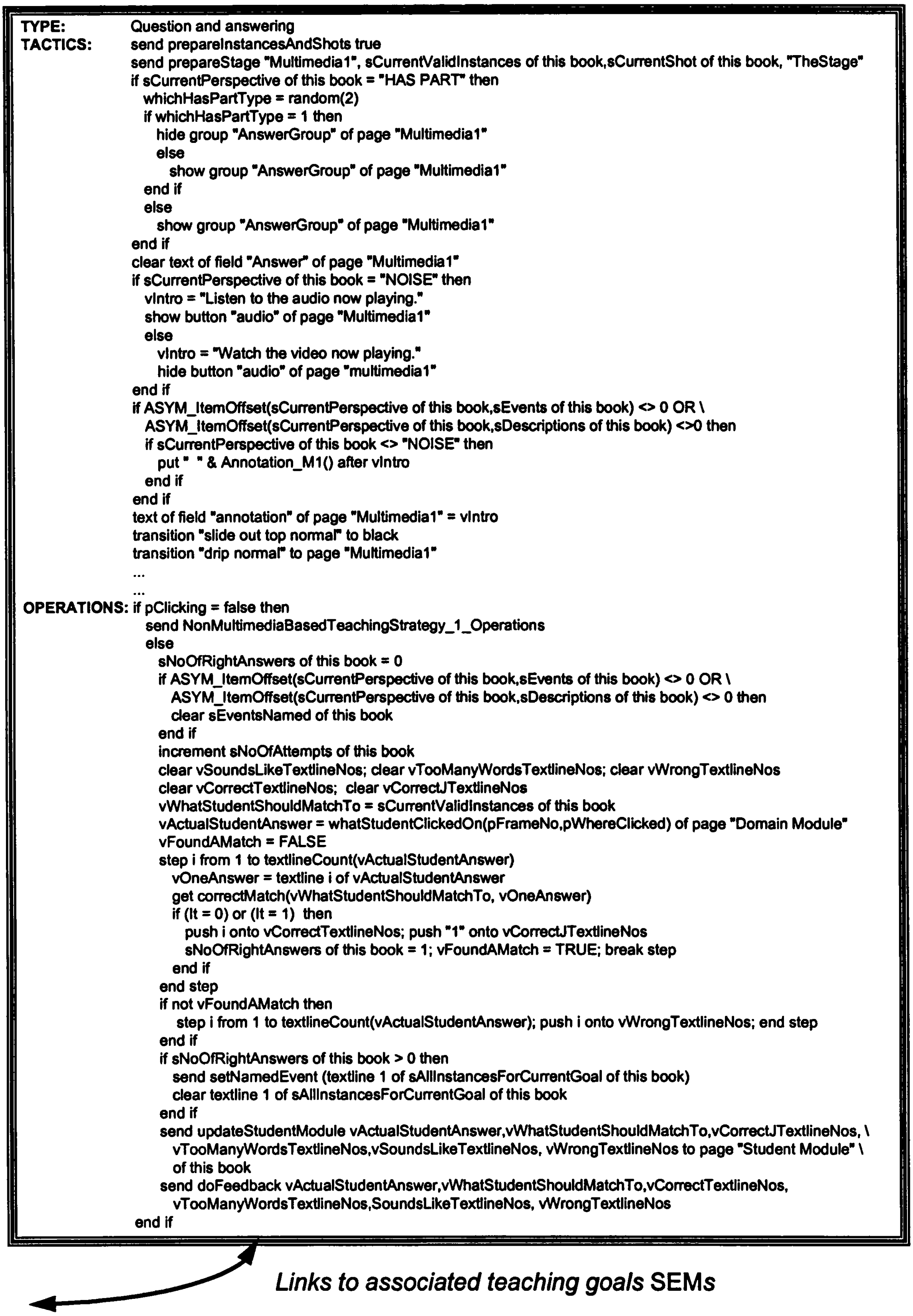
Non-multimedia-based teaching strategies provide ways in which to teach the student-user about aspects of an animal or animal class that cannot be taught through the use of multimedia. For example, ARISTOTLE teaches about animal categorisations with text only because no videos or audios could be found to illustrate this concept.

Conversely, *multimedia-based teaching strategies* are those that are concerned with the use of multimedia components within the teaching-learning interaction. More specifically, these strategies provide different ways to teach a particular subset of knowledge with the appropriate use of multimedia. For example, teaching that a cheetah is able to hunt by showing a video of a cheetah hunting. Most of the teaching in ARISTOTLE uses multimedia-based teaching strategies. For example, to teach about body parts that particular animals have, or teach that a particular animal is able to do certain activities, appropriate shots for that animal will be used (where they are available).

A teaching strategy *SEM* has three perspectives: (1) TYPE, which indicates the nature of the teaching strategy; (2) TACTICS, which provides the procedures for presenting the teaching goal to the student-user; and (3) OPERATIONS, which provides the procedures for evaluating the student-user's responses.

ARISTOTLE has two non-multimedia-based and two multimedia-based teaching strategies:

- **Non-multimedia-based teaching strategy 1** is a simple question and answering strategy that asks the student-user a question, and waits for a response.
- **Non-multimedia-based teaching strategy 2** is a multiple choice strategy that asks the student-user to select the right answer from three alternatives.
- **Multimedia-based teaching strategy 1** is a multimedia-based question and answering strategy. It presents a video or audio shot to the student-user, and asks a question related to the content of the shot. Student-user responses may be through clicking on an object that is present in the shot (e.g. a cheetah's claws), or through typing an answer into an input line. A conceptual representation of the *SEM* for this teaching strategy is depicted in Figure 4.12. Because the code within the TACTICS perspective is very long, only the main portions of this code have been provided. The code that is not given concerns preparations for the playing of video or audio shots, that is, it deals with the opening and cueing-up of videos or audios.



```
TYPE:        Question and answering
TACTICS:     send prepareInstancesAndShots true
             send prepareStage "Multimedia1", sCurrentValidInstances of this book,sCurrentShot of this book, "TheStage"
             if sCurrentPerspective of this book = "HAS PART" then
               whichHasPartType = random(2)
               if whichHasPartType = 1 then
                 hide group "AnswerGroup" of page "Multimedia1"
                 else
                   show group "AnswerGroup" of page "Multimedia1"
               end if
               else
                 show group "AnswerGroup" of page "Multimedia1"
             end if
             clear text of field "Answer" of page "Multimedia1"
             if sCurrentPerspective of this book = "NOISE" then
               vIntro = "Listen to the audio now playing."
               show button "audio" of page "Multimedia1"
               else
                 vIntro = "Watch the video now playing."
                 hide button "audio" of page "multimedia1"
             end if
             if ASYM_ItemOffset(sCurrentPerspective of this book,sEvents of this book) <> 0 OR \
               ASYM_ItemOffset(sCurrentPerspective of this book,sDescriptions of this book) <>0 then
               if sCurrentPerspective of this book <> "NOISE" then
                 put " " & Annotation_M1() after vIntro
               end if
             end if
             text of field "annotation" of page "Multimedia1" = vIntro
             transition "slide out top normal" to black
             transition "drip normal" to page "Multimedia1"
             ...
             ...
OPERATIONS: if pClicking = false then
               send NonMultimediaBasedTeachingStrategy_1_Operations
               else
                 sNoOfRightAnswers of this book = 0
                 if ASYM_ItemOffset(sCurrentPerspective of this book,sEvents of this book) <> 0 OR \
                   ASYM_ItemOffset(sCurrentPerspective of this book,sDescriptions of this book) <> 0 then
                   clear sEventsNamed of this book
                 end if
                 increment sNoOfAttempts of this book
                 clear vSoundsLikeTextlineNos; clear vTooManyWordsTextlineNos; clear vWrongTextlineNos
                 clear vCorrectTextlineNos; clear vCorrectJTextlineNos
                 vWhatStudentShouldMatchTo = sCurrentValidInstances of this book
                 vActualStudentAnswer = whatStudentClickedOn(pFrameNo,pWhereClicked) of page "Domain Module"
                 vFoundAMatch = FALSE
                 step i from 1 to textlineCount(vActualStudentAnswer)
                   vOneAnswer = textline i of vActualStudentAnswer
                   get correctMatch(vWhatStudentShouldMatchTo, vOneAnswer)
                   if (It = 0) or (It = 1) then
                     push i onto vCorrectTextlineNos; push "1" onto vCorrectJTextlineNos
                     sNoOfRightAnswers of this book = 1; vFoundAMatch = TRUE; break step
                   end if
                 end step
                 if not vFoundAMatch then
                    step i from 1 to textlineCount(vActualStudentAnswer); push i onto vWrongTextlineNos; end step
                 end if
                 if sNoOfRightAnswers of this book > 0 then
                   send setNamedEvent (textline 1 of sAllInstancesForCurrentGoal of this book)
                   clear textline 1 of sAllInstancesForCurrentGoal of this book
                 end if
                 send updateStudentModule vActualStudentAnswer,vWhatStudentShouldMatchTo,vCorrectJTextlineNos, \
                   vTooManyWordsTextlineNos,vSoundsLikeTextlineNos, vWrongTextlineNos to page "Student Module" \
                   of this book
                 send doFeedback vActualStudentAnswer,vWhatStudentShouldMatchTo,vCorrectTextlineNos,
                   vTooManyWordsTextlineNos,SoundsLikeTextlineNos, vWrongTextlineNos
             end if
```

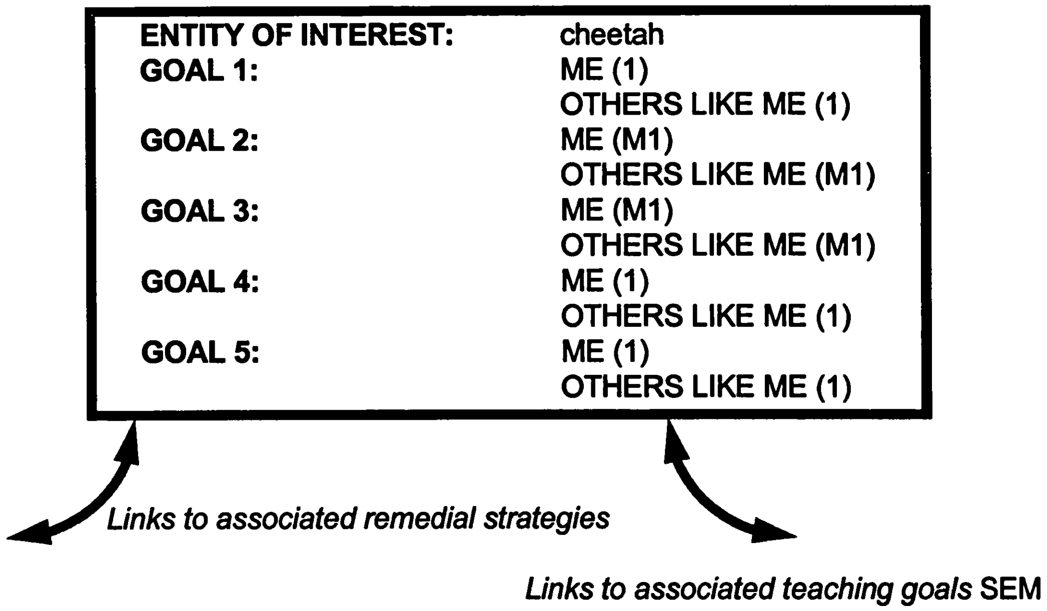
Links to associated teaching goals SEMs

Figure 4.12 Conceptual representation of one of ARISTOTLE's multimedia-based teaching strategy *SEMs*.

- **Multimedia-based teaching strategy 2** is a multimedia-based multiple choice strategy. It presents three video or audio shots to the student-user, and asks the student-user to select, from three alternatives, the shot that represents the correct answer to the question. For example, if the question were "Click on the cheetah's long legs.", the student-user would click on the legs within the video shot that showed footage of a cheetah's legs.

The *remedial goals* are used to provide remedial assistance to the student-user, on request by the student-user. There is one remedial goals *SEM* for each and every teaching goals *SEM* within ARISTOTLE. The Cheetah remedial goals *SEM* from ARISTOTLE is depicted in Figure 4.13. Each remedial goal within a remedial goals *SEM* consists of a number of sub-goals, each of which has an associated remedial strategy (indicated in brackets in the figure), which is used to carry out the remediation. Each time the student-user asks for assistance, the next remedial sub-goal for the current teaching goal is executed. The remedial sub-goals are executed in the order in which they appear within the remedial goals *SEM*. When the sub-goals are exhausted, the student-user may no longer request assistance for the current teaching goal.

The remedial sub-goals take one of two forms: "ME" or "OTHERS LIKE ME". The "ME" remedial sub-goal indicates that the remedial information from the corresponding remedial *SEM* should be used to provide the remediation. The "OTHERS LIKE ME" remedial sub-goal indicates that remediation should take place by informing the student-user of other animals (entities of interest) that have at least one of the current instance values (i.e. those answers that the student-user is currently expected to respond with in order to satisfy the teaching goal) in common with the animal currently being taught about.



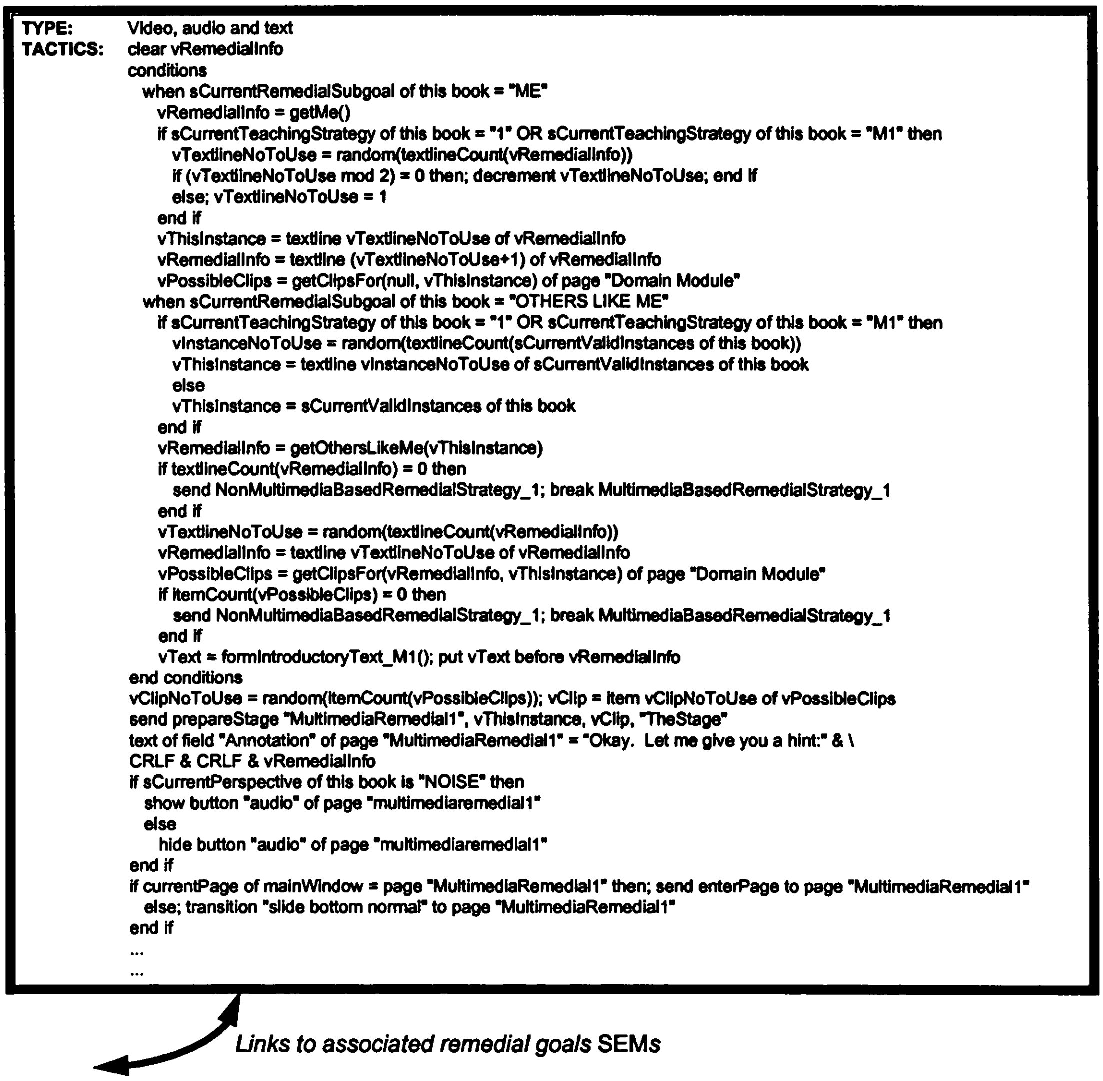
ENTITY OF INTEREST:	cheetah
GOAL 1:	ME (1) OTHERS LIKE ME (1)
GOAL 2:	ME (M1) OTHERS LIKE ME (M1)
GOAL 3:	ME (M1) OTHERS LIKE ME (M1)
GOAL 4:	ME (1) OTHERS LIKE ME (1)
GOAL 5:	ME (1) OTHERS LIKE ME (1)

Links to associated remedial strategies
Links to associated teaching goals SEM

Figure 4.13 Conceptual representation of the Cheetah remedial goals *SEM* in ARISTOTLE.

As was the case with the teaching strategy *SEM*s, the links to the associated remedial strategies are dynamic and are established during the course of interaction by calling the appropriate remedial strategy *SEM* based on the remedial strategy code (i.e. 1 or M1).

The *remedial strategies* were the last component of ARISTOTLE's tutor module to be implemented. ARISTOTLE has one non-multimedia-based and one multimedia-based remedial strategy. The non-multimedia-based remedial strategy uses the textual information from the corresponding remedial *SEM*, whereas the multimedia-based remedial strategy also uses the logical video and audio segments of the corresponding domain *SEM* during the remediation. Remedial strategy *SEM*s have a TYPE perspective which describes the strategy, and a TACTICS perspective which provides the routines for presenting the remedial information to the student-user. Figure 4.14 shows the multimedia-based remedial strategy *SEM* from ARISTOTLE. Because the code held within the TACTICS perspective is very long, only the main portions of the code are shown. The missing code deals with the opening and cueing-up of the video or audio shot for presentation.



```
TYPE:       Video, audio and text
TACTICS:    clear vRemedialInfo
            conditions
              when sCurrentRemedialSubgoal of this book = "ME"
                vRemedialInfo = getMe()
                if sCurrentTeachingStrategy of this book = "1" OR sCurrentTeachingStrategy of this book = "M1" then
                  vTextlineNoToUse = random(textlineCount(vRemedialInfo))
                  if (vTextlineNoToUse mod 2) = 0 then; decrement vTextlineNoToUse; end if
                  else; vTextlineNoToUse = 1
                end if
                vThisInstance = textline vTextlineNoToUse of vRemedialInfo
                vRemedialInfo = textline (vTextlineNoToUse+1) of vRemedialInfo
                vPossibleClips = getClipsFor(null, vThisInstance) of page "Domain Module"
              when sCurrentRemedialSubgoal of this book = "OTHERS LIKE ME"
                if sCurrentTeachingStrategy of this book = "1" OR sCurrentTeachingStrategy of this book = "M1" then
                  vInstanceNoToUse = random(textlineCount(sCurrentValidInstances of this book))
                  vThisInstance = textline vInstanceNoToUse of sCurrentValidInstances of this book
                  else
                  vThisInstance = sCurrentValidInstances of this book
                end if
                vRemedialInfo = getOthersLikeMe(vThisInstance)
                if textlineCount(vRemedialInfo) = 0 then
                  send NonMultimediaBasedRemedialStrategy_1; break MultimediaBasedRemedialStrategy_1
                end if
                vTextlineNoToUse = random(textlineCount(vRemedialInfo))
                vRemedialInfo = textline vTextlineNoToUse of vRemedialInfo
                vPossibleClips = getClipsFor(vRemedialInfo, vThisInstance) of page "Domain Module"
                if itemCount(vPossibleClips) = 0 then
                  send NonMultimediaBasedRemedialStrategy_1; break MultimediaBasedRemedialStrategy_1
                end if
                vText = formIntroductoryText_M1(); put vText before vRemedialInfo
            end conditions
            vClipNoToUse = random(itemCount(vPossibleClips)); vClip = item vClipNoToUse of vPossibleClips
            send prepareStage "MultimediaRemedial1", vThisInstance, vClip, "TheStage"
            text of field "Annotation" of page "MultimediaRemedial1" = "Okay.  Let me give you a hint:" & \
            CRLF & CRLF & vRemedialInfo
            if sCurrentPerspective of this book is "NOISE" then
              show button "audio" of page "multimediaremedial1"
              else
                hide button "audio" of page "multimediaremedial1"
            end if
            if currentPage of mainWindow = page "MultimediaRemedial1" then; send enterPage to page "MultimediaRemedial1"
              else; transition "slide bottom normal" to page "MultimediaRemedial1"
            end if
            ...
            ...
```

Links to associated remedial goals SEMs

Figure 4.14 Conceptual representation of ARISTOTLE's multimedia-based remedial strategy *SEM*.

Once all of the tutor module SEMs had been created, the procedures to retrieve and manipulate their information were developed. These procedures perform various tasks, such as retrieving teaching and remedial goals, and moving between teaching goals SEMs through the setting of the physical name of the table for the database alias. The tutor module must also formulate textual annotations and questions for a given teaching goal and teaching strategy. This is achieved through the use of procedures that are called from within the teaching strategy SEMs. For example, a 'HAS PART' teaching goal may be presented to the student-user in two forms (determined randomly): the student-user is asked to click on the part on the screen, or the student-user is asked to name the part displayed in the video. In the second case, the domain module procedures are used to determine which objects are spatially located around the part in question, and then the part that the student-user must name is described in terms of its spatial relationship with other objects on-screen.

Implementing ARISTOTLE's student module

The last module to be implemented is the student module. The student module provides valuable information to the tutor module about the status of the student-user so that the tutor module may alter its tutoring processes accordingly. Unlike the domain and tutor modules, the majority of the student module is constructed during the course of the student-user's interaction with the system as student overlay knowledge of the domain knowledge and diagnosed student misconceptions. This stage, then, consists mainly of providing the processes for allowing this real-time construction to take place. However, the student module also includes knowledge regarding common misconceptions which are used during diagnosis. This 'bugs library' is *not* constructed during interaction with the system.

The *bugs library* was the first component of the student module to be developed in ARISTOTLE. In consists of mal SEMs which record common misconceptions related to values or classes for a particular entity of interest. This information is used to inform the student-user that their mistake is a common one. Figure 4.15 provides a conceptual representation of the Scorpion mal SEM from ARISTOTLE.

ENTITY OF INTEREST:	scorpion
CLASS EXCEPTIONS:	insect reptile
VALUE EXCEPTIONS:	wings

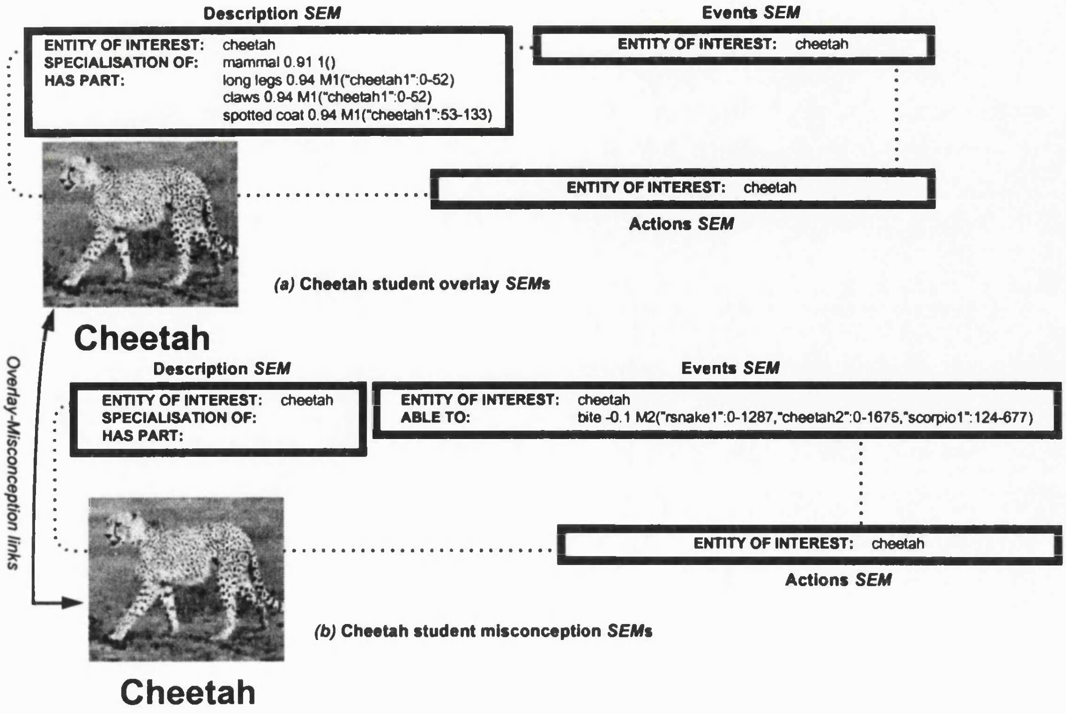
Figure 4.15 Conceptual representation of ARISTOTLE's Scorpion mal *SEM*.

CLASS EXCEPTIONS cater for situations were a student-user perceives certain animals as looking or sounding similar to each other. In these cases, it is probable that a student-user may perceive one animal as belonging to a generic animal class of which it is not actually a member. Thus, the Scorpion mal *SEM* records the fact that a scorpion may be confused with a reptile or an insect by the student-user, since scorpions look like reptiles or insects.

Similarly, VALUE EXCEPTIONS are used to record instance values that are commonly misconceived as belonging to the entity of interest. Thus, the Scorpion mal *SEM* records the fact that the student-user may suggest that a scorpion has wings (perhaps because scorpions are often thought to be insects).

The *student overlay knowledge* is a representation of the current status of the student-user in terms of correct knowledge attained. This is represented as student overlay *SEMs*. Figure 4.16(a) shows a conceptual representation of typical Cheetah student overlay *SEMs* in ARISTOTLE.

The student overlay *SEMs* mirror the structure of the domain *SEMs*, but include different information. Each instance in a student overlay *SEM* records the correct answer the student-user provided, the strength of the acquired knowledge (rated between 0 and 1), the successful teaching strategy used to elicit this response, and the shots that were used if a multimedia-based teaching strategy was used.


Description *SEM*
ENTITY OF INTEREST: cheetah
SPECIALISATION OF: mammal 0.91 1()
HAS PART: long legs 0.94 M1("cheetah1":0-52)
claws 0.94 M1("cheetah1":0-52)
spotted coat 0.94 M1("cheetah1":53-133)
Events *SEM*
ENTITY OF INTEREST: cheetah
ENTITY OF INTEREST: cheetah
Actions *SEM*
Cheetah
Overlay-Misconception links
Description *SEM*
ENTITY OF INTEREST: cheetah
SPECIALISATION OF:
HAS PART:
Events *SEM*
ENTITY OF INTEREST: cheetah
ABLE TO: bite -0.1 M2("rsnake1":0-1287,"cheetah2":0-1675,"scorpio1":124-677)
ENTITY OF INTEREST: cheetah
Actions *SEM*
Cheetah

(a) **Cheetah student overlay *SEMs***

(b) **Cheetah student misconception *SEMs***

Figure 4.16 Conceptual representation of: **(a)** typical Cheetah student overlay *SEMs*, and **(b)** the corresponding Cheetah student misconception *SEMs* in ARISTOTLE.

There are two main ways in which the student knowledge differs from the domain knowledge: missing conceptions and misconceptions. A missing conception is an item of knowledge that the domain has but the student-user does not. In this case the student overlay knowledge is a proper subset of the domain knowledge. Missing conceptions are determined by 'subtracting' the student overlay knowledge from the domain knowledge.

Student misconceptions are represented with student misconception *SEMs*. These are similar to the student overlay *SEMs*, but the appended numbers indicate the *seriousness* of the bad knowledge (on a scale between -1 and 0). Figure 4.16(b) shows a conceptual representation of typical Cheetah student misconception *SEMs* in ARISTOTLE.

ARISTOTLE's student overlay knowledge and student misconceptions together hold the inferred knowledge of the student-user. They are linked through the use of database aliases. In addition to the student overlay and misconception SEMs, ARISTOTLE records the path the student-user has so far taken through the domain knowledge, together with the type of path he or she has taken (either depth-first or breadth-first), and the aggregate knowledge weights the various teaching strategies have yielded, together with an aggregate usage score. This information is stored in this way to avoid the need to calculate these values from the entire student overlay and misconception SEMs, and thus to speed the process of determining the effectiveness of various teaching strategies. The *aggregate knowledge weight* of each teaching strategy is determined by adding the overlay and misconception knowledge weights from the current student-user's response to the existing value stored. The *aggregate usage score* is incremented every time the teaching strategy has achieved a teaching goal, and decremented every time it has not.

The final development of the student module is that of developing the procedures for retrieving and storing the information contained within the overlay and misconception SEMs and the mal SEMs. Procedures were also developed to calculate the knowledge weights to be attached to a particular overlay or misconception instance. The knowledge weight to be attached to a particular overlay instance is calculated by subtracting the following from 1.0: the number of attempts made by the student-user multiplied by 0.06, the number of times the user asked for assistance multiplied by 0.1, and a value based on how well the student-user provided the answer required (e.g. if the student-user misspelled the answer then a further 0.05 would be subtracted). If the end result is less than 0, then 0 is the value used.

The knowledge weight to be attached to a particular misconception instance is calculated by subtracting the following from 0: the number of attempts made by the student-user multiplied by 0.06, the number of times the user asked for assistance multiplied by 0.1, and a value based on how badly the student-user provided the answer required (e.g. if the student-user provided an answer found in the corresponding mal *SEM* then a further 0.1 would be subtracted). If the end result is less than -1.0, then -1.0 is the value used.

The student module also orders the teaching strategies to be used by the tutor module according to their successfulness with the current student-user. To achieve this, the teaching strategies' names and successfulness values are stored in a two-dimensional array. The successfulness value of each teaching strategy is calculated by multiplying its aggregate knowledge weight and its aggregate usage score. The array is then 'quicksorted', according to the successfulness values, in descending order so that the most successful teaching strategy appears first in the array. Those teaching strategies which are not to be used for the achievement of the current teaching goal are then deleted from the array.

4.2 ARISTOTLE IN ACTION

ARISTOTLE greets the student-user with the screen shown in Figure 4.17. When the student-user clicks on the 'Let's go!' sign, they are asked for their name so that new student-users may be initialised, or previous student-users may continue their tuition where they left off previously.

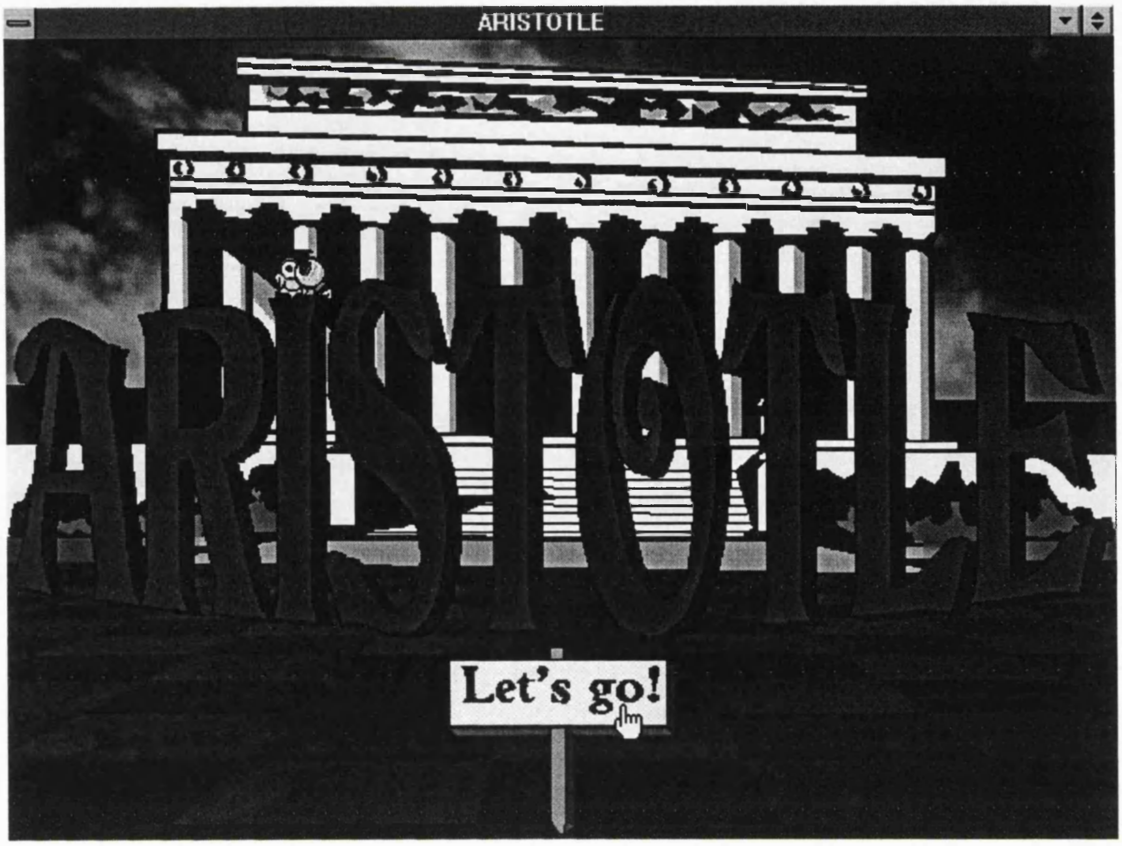
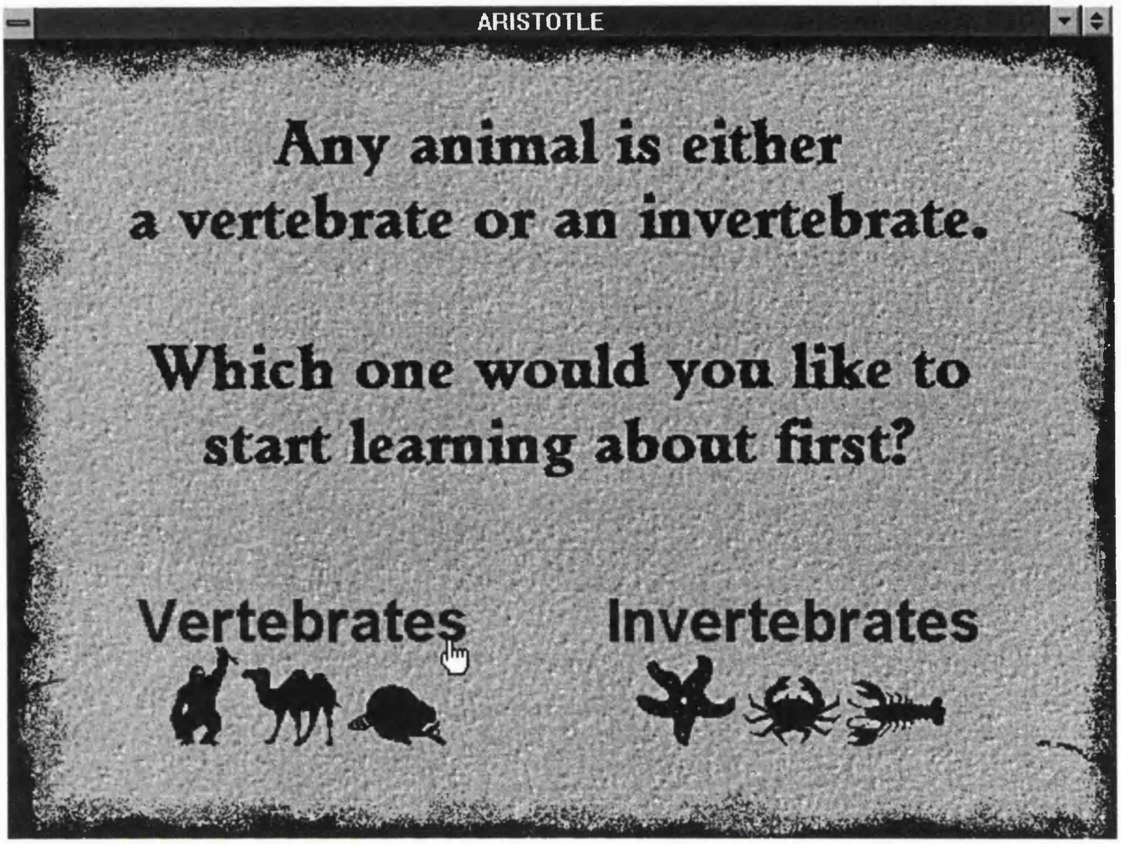


Figure 4.17 ARISTOTLE's opening screen.

The student-user must then decide whether they wish to be first taught about vertebrates or invertebrates (Figure 4.18). On selection, the domain module prepares a stack that contains all animals within the domain knowledge in either depth-first or breadth-first order (decided randomly). This stack is then used by the tutor module to determine the order in which the teaching goals are to be attempted, and thus the order in which the teaching goals *SEMs* are to be used.



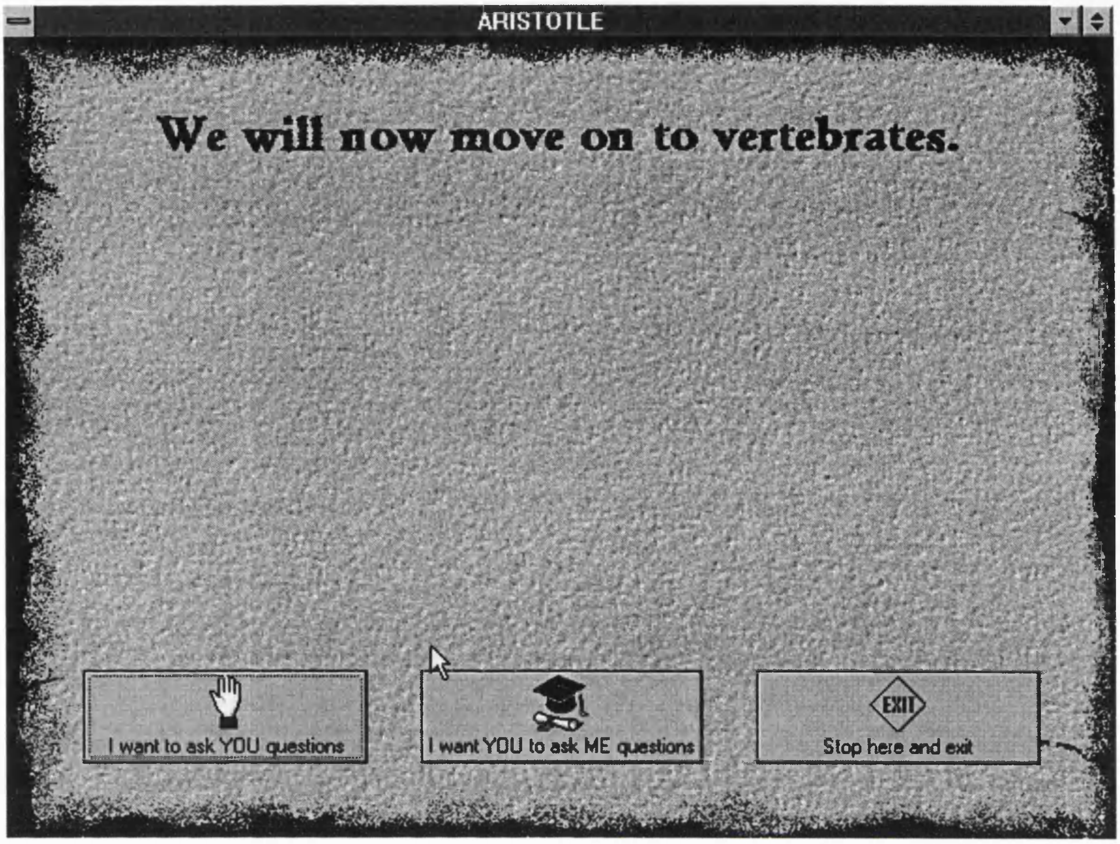
ARISTOTLE

Any animal is either
a vertebrate or an invertebrate.

Which one would you like to
start learning about first?

Figure 4.18 Student-user chooses between vertebrates and invertebrates in ARISTOTLE.

At the start of each goal, the student-user is presented with the option to: (1) ask ARISTOTLE questions, (2) let ARISTOTLE ask questions, or (3) stop here and exit (Figure 4.19).

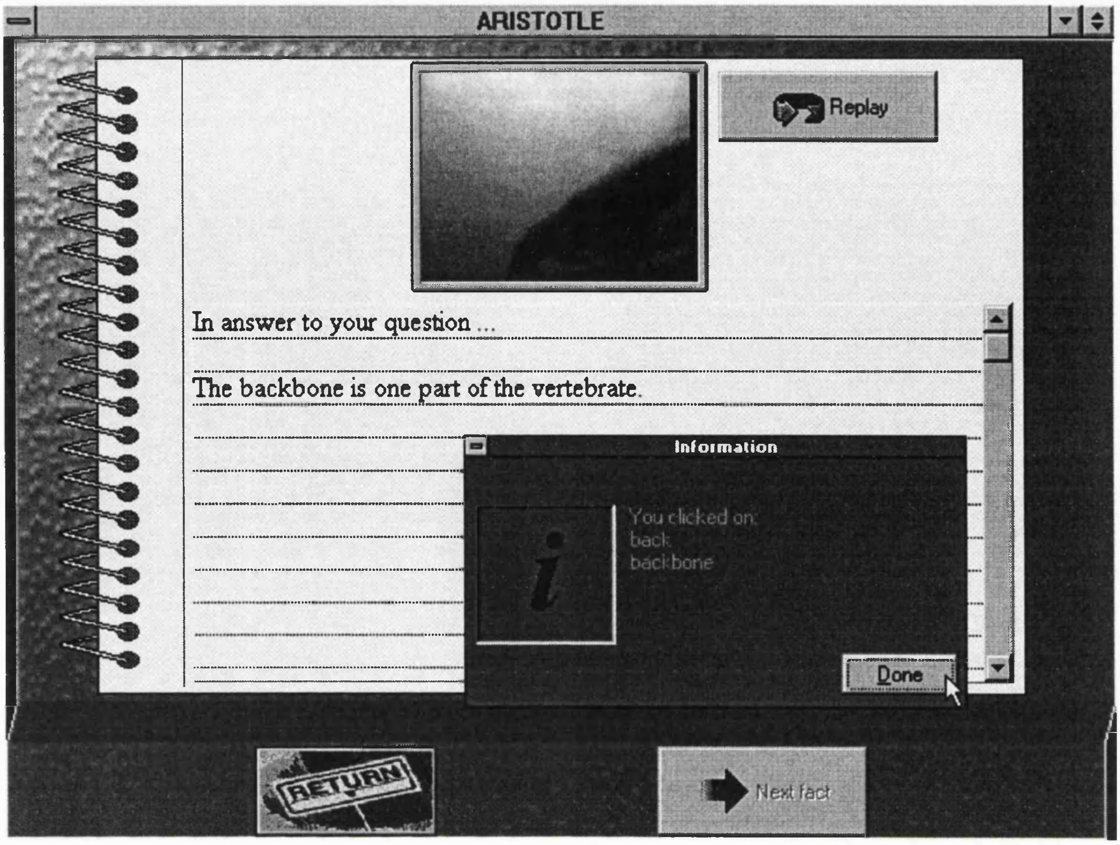


ARISTOTLE

We will now move on to vertebrates.

Figure 4.19 ARISTOTLE begins the teaching with vertebrates.

If the student-user chooses the first option, ARISTOTLE formulates a list of possible questions that may be asked, based on the current teaching goal *SEM*, e.g. “What distinguishing parts does a vertebrate have?”. The student-user then selects a question from those available, and ARISTOTLE provides the answer. If the question relates to a perspective within a domain Description *SEM* then the student-user may click on an object while the video is playing (Figure 4.20). The domain module routines are then used to determine which object the student-user has clicked on so that they may be informed of this.



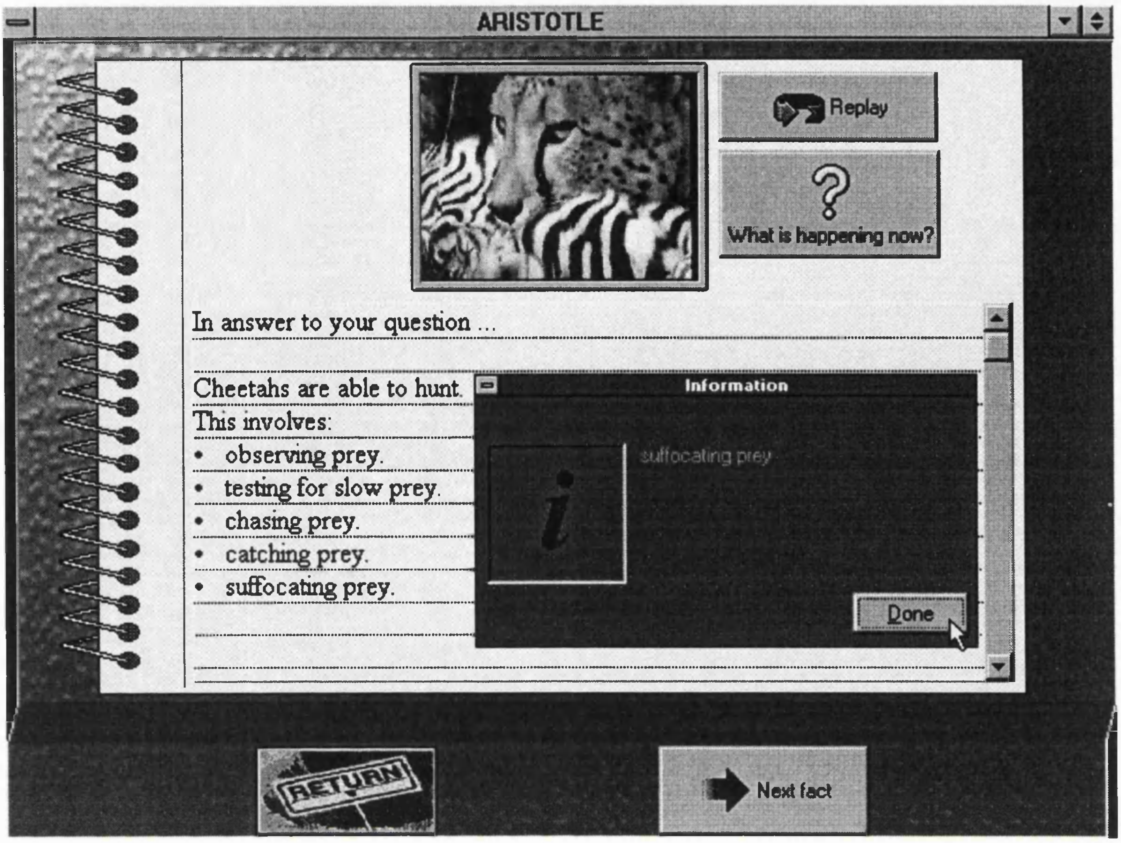
In answer to your question ...

The backbone is one part of the vertebrate.

You clicked on:
back
backbone

Figure 4.20 ARISTOTLE provides the answer to the student-user's question "What distinguishing parts does a vertebrate have?".

If the question relates to a domain Events *SEM*, then the student-user is played a video that depicts the event, and is informed of the actions that constitute the event, as they occur within the video (Figure 4.21). In addition, the student-user may click on a 'What is happening now?' button, at any time, to ask about the actions that are being shown in the video at that moment.



In answer to your question ...

Cheetahs are able to hunt.
This involves:
- observing prey.
- testing for slow prey.
- chasing prey.
- catching prey.
- suffocating prey.

suffocating prey

Figure 4.21 ARISTOTLE provides the answer to the student-user's question "What is a cheetah able to do?".

When the student-user has finished asking a question, they are returned to the screen given in Figure 4.19, from which the same three options are available again. The student-user may only progress to another animal after ARISTOTLE has asked questions about that animal.

When the student-user chooses to have ARISTOTLE ask questions, the first teaching goal is taken from the teaching goal *SEM* and the student module then orders the associated teaching strategies according to their success rate with the current student-user. The student module ensures that teaching strategies that have never been used before are given highest priority.

Figure 4.22 shows non-multimedia-based teaching strategy 1 being used with a Reptile teaching goal. As well as attempting to answer the question, the student-user may also ask why they are being asked a particular question, via the 'Why should I?' button. In this case they would be told, "I want to see if you know what type of animal a reptile is." They may also request assistance, in which case the first remedial sub-goal for the SPECIALISATION OF perspective from the Reptile remedial goals *SEM* is executed. Once all the sub-goals for a particular perspective have been exhausted, the student-user may no longer request assistance, and the assistance button is disabled.

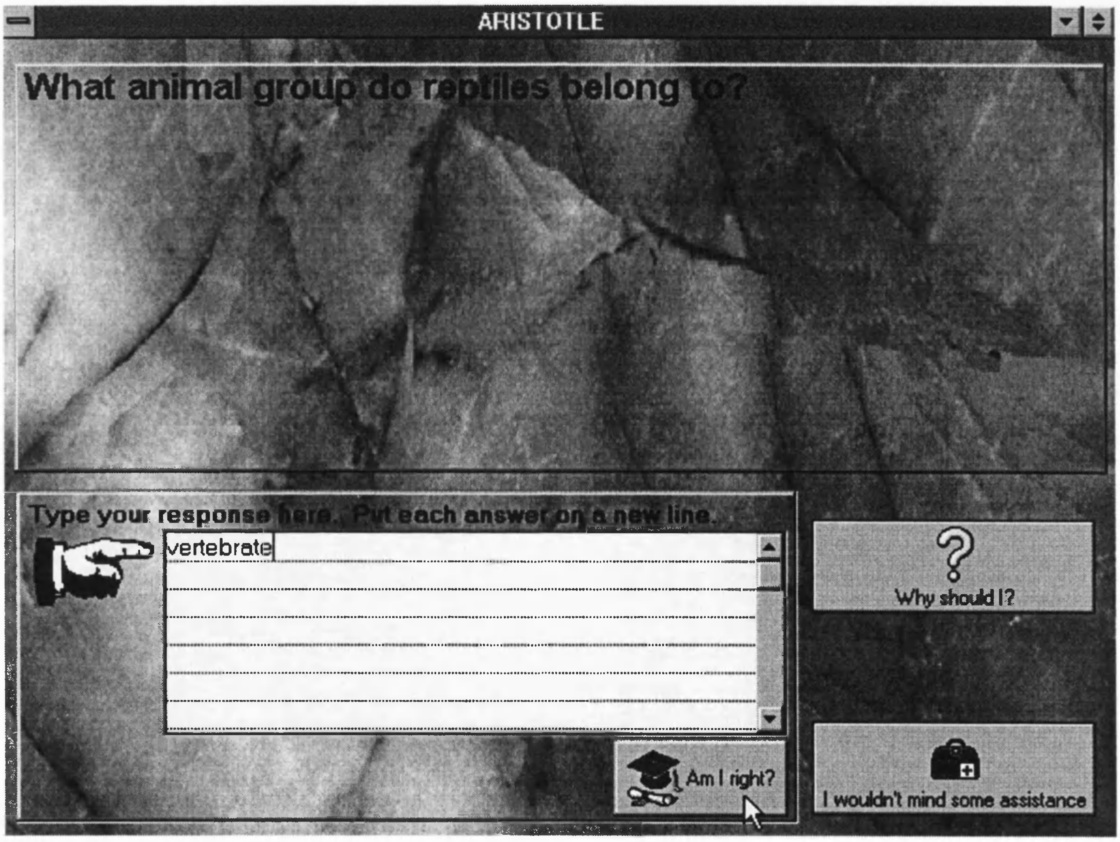
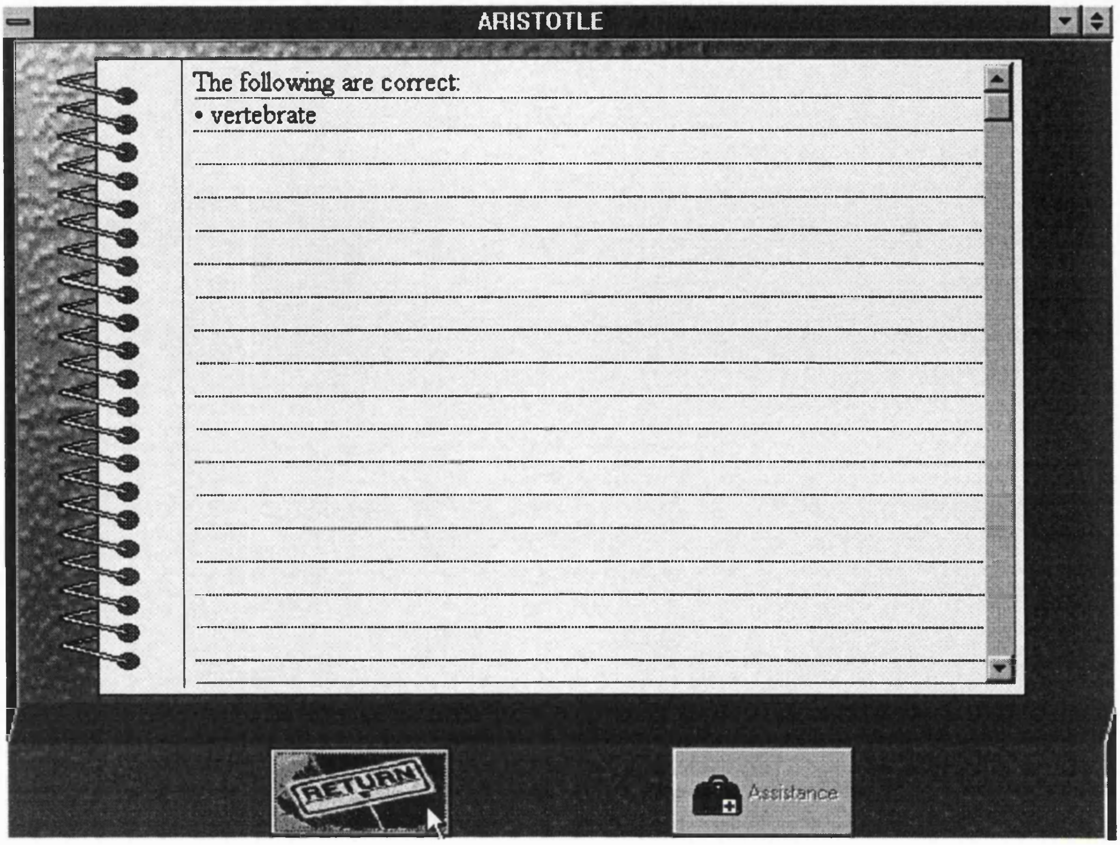

ARISTOTLE
What animal group do reptiles belong to?
Type your response here. Put each answer on a new line.
vertebrate
Why should I?
Am I right?
I wouldn't mind some assistance

Figure 4.22 Non-multimedia-based teaching strategy 1 is used.

Once the student-user has attempted to answer the question, the student module updates the student overlay and misconception *SEMs* for the current entity of interest, and then updates the aggregate knowledge weight and aggregate usage score for the teaching strategy that was used. Then, the tutor module provides feedback based on the student-user's response. This feedback informs the student-user which of their answers are correct, which are correct but are spelt incorrectly, and which are wrong. ARISTOTLE will accept answers from the student-user which sound similar to those expected when pronounced (e.g. 'vertebrait' is an acceptable response if the correct answer is 'vertebrate'), as well as answers which are full sentences that include the correct answer (e.g. 'got to be a vertebrate'). Figure 4.23 shows a feedback screen from ARISTOTLE based on the response provided in Figure 4.22. From here, the student-user may return to teaching (via the 'Return' button) or, if at least one incorrect answer was given or the student-user failed to name the minimum number of instances required to achieve the teaching goal, they may request assistance (in which case the next remedial sub-goal for the current teaching goal is executed).

On returning to teaching, If the student has provided at least one correct answer, then the question is asked again using the same teaching strategy as was used previously. If the student has failed to provide *any* correct answers to the question asked, then the tutor module will ask the question again using the next best teaching strategy from those specified for the current goal in the current teaching goals SEM. If all teaching strategies have been exhausted, then the tutor module moves on to the next teaching goal.

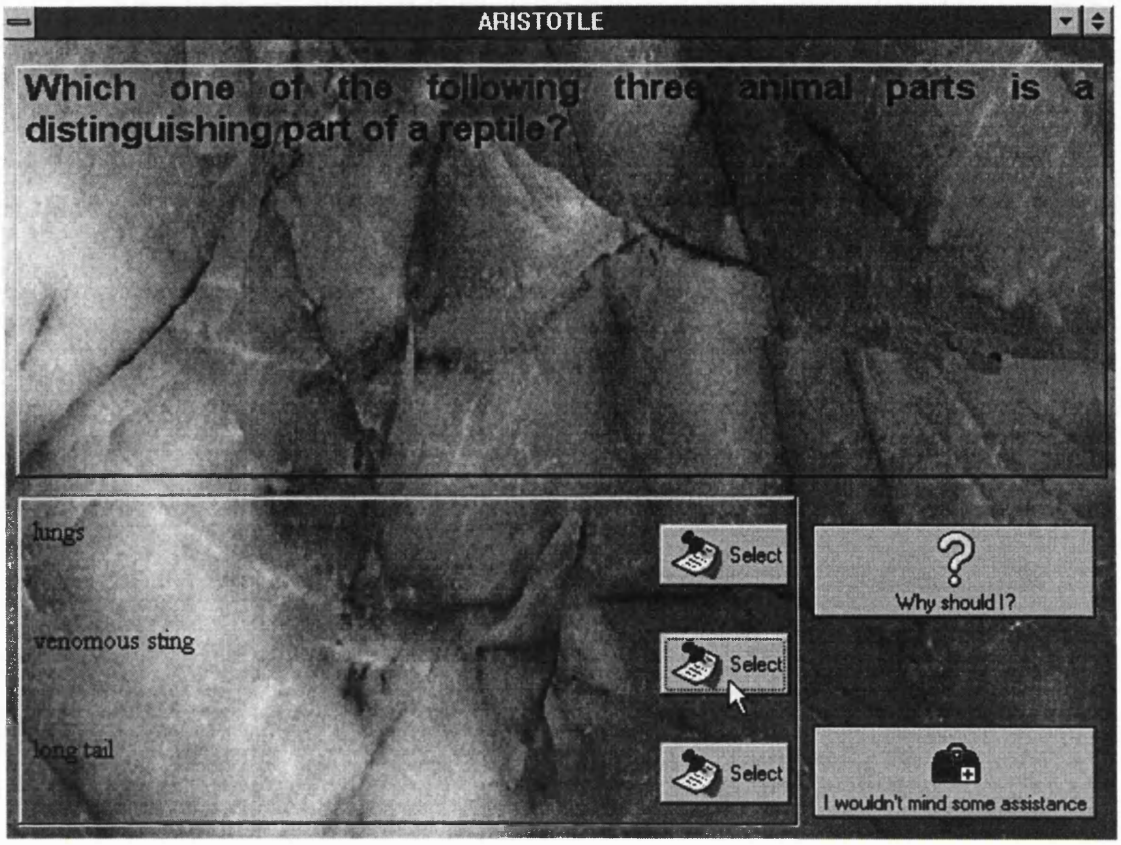


The following are correct:

- vertebrate

Figure 4.23 ARISTOTLE provides feedback to the student-user based on their responses.

Figure 4.24 shows ARISTOTLE using non-multimedia-based teaching strategy 2 for one of the teaching goals for reptiles. Here, the domain module returns two alternative answers to the correct one to the tutor module. To achieve this it searches through the domain *SEM*s and retrieves all the instances within each one that have a perspective that matches the current teaching goal. The two alternatives are then drawn at random by the tutor module. The position of the correct answer in relation to the two incorrect answers is also determined randomly.



ARISTOTLE

Which one of the following three animal parts is a distinguishing part of a reptile?

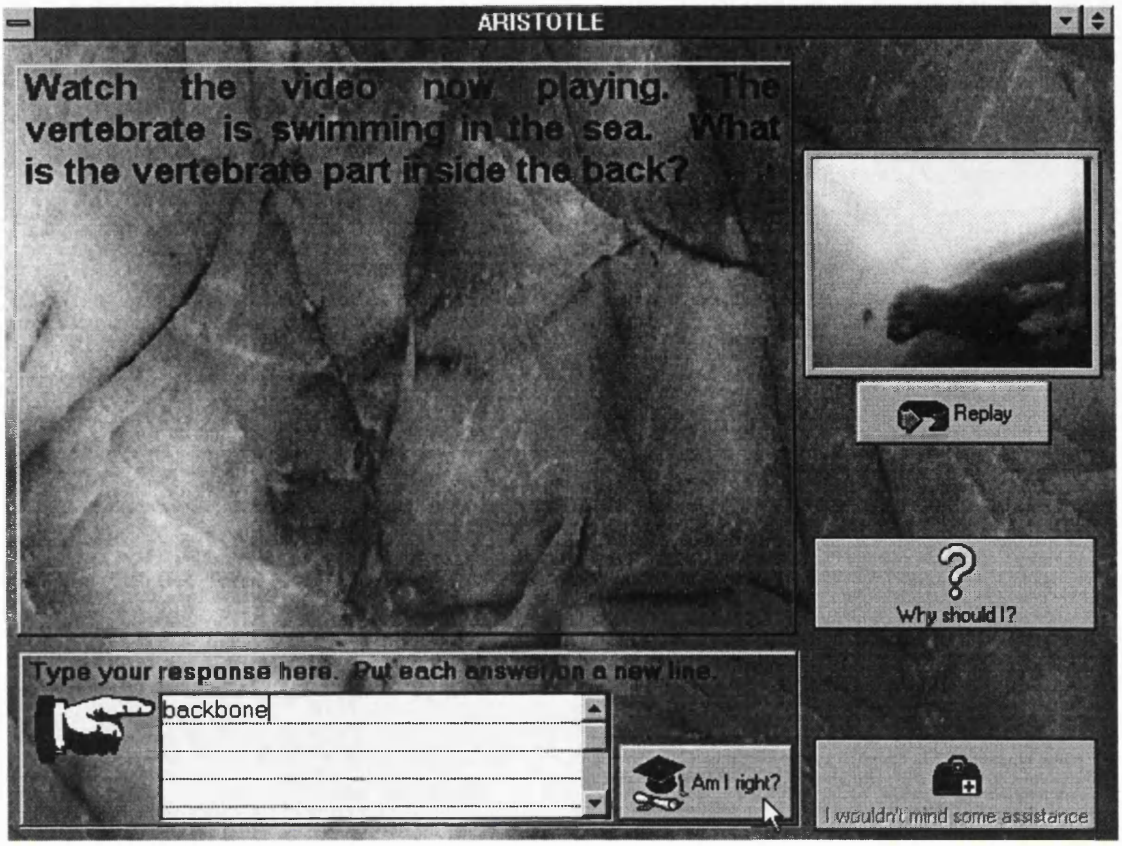
lungs

venomous sting

long tail

Figure 4.24 ARISTOTLE uses non-multimedia-based teaching strategy 2.

Figure 4.25 shows ARISTOTLE using multimedia-based teaching strategy 1 for the 'HAS PART' teaching goal for vertebrates. The teaching strategy, in conjunction with the procedures in the domain module, plays a video or audio shot, formulates a textual annotation that describes the incidental content of the video or audio (based on the ANNOTATION perspective for the domain Description *SEM* of the current entity of interest), and requests a response from the student-user.



Watch the video now playing. The vertebrate is swimming in the sea. What is the vertebrate part inside the back?

Type your response here. Put each answer on a new line.

backbone

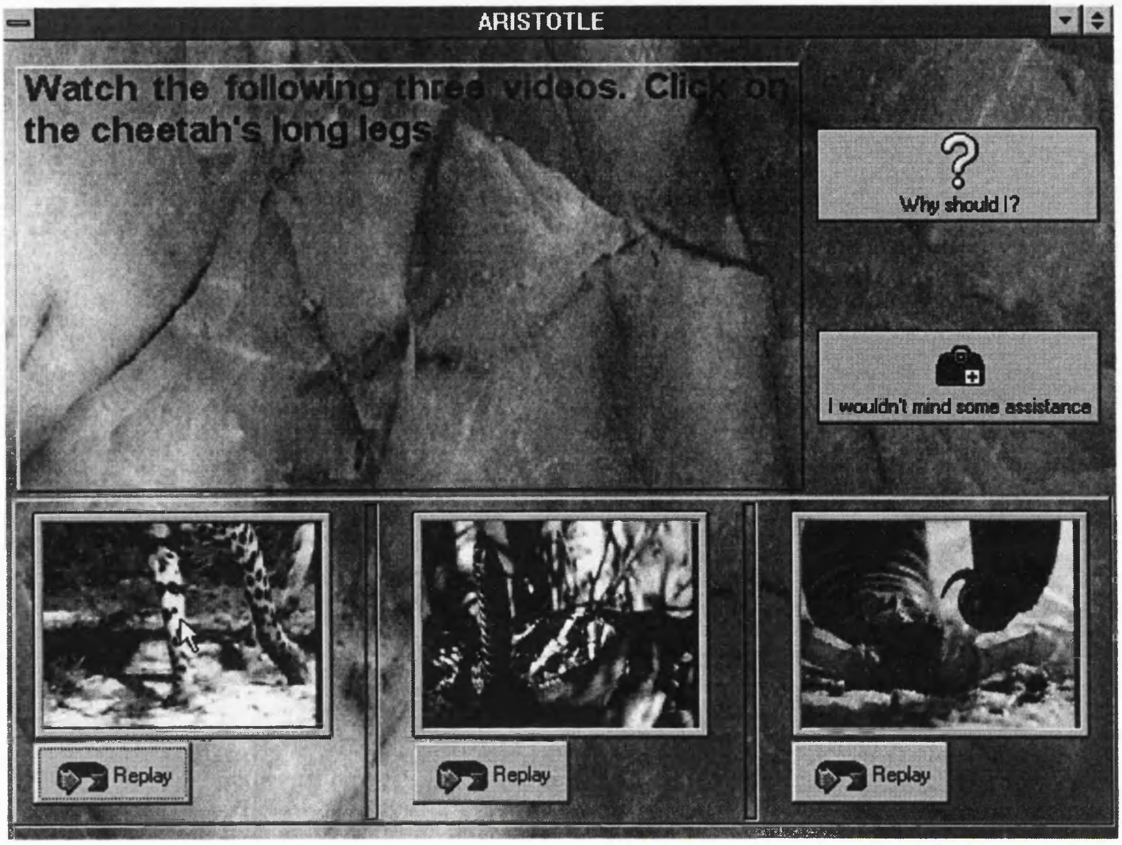
Figure 4.25 ARISTOTLE uses multimedia-based teaching strategy 1.

The textual annotation is based on all the annotations within the domain knowledge whose associated shots intersect with the shot being used by the teaching strategy. The domain module collects and orders the annotations (according to the *i* values of their associated shots) and passes them to the tutor module. The tutor module then retrieves from the annotations a suitable description of the incidental content of the shot.

Depending on the teaching goal that is to be achieved using this teaching strategy, the student-user is invited to either: (1) name the animal part that is described in relation to the other objects present in the video frame, (2) click on a named animal part, (3) provide a textual response to a question based on the content of the video or audio. If the student-user clicks on an animal part, the teaching strategy then provides the domain module with the co-ordinates of where the student-user clicked, together with the clip name and SYM they correspond to. The domain module then returns a list of objects that have been defined for that location of the video. If the correct answer is contained within that list then the student-user is assumed to be correct.

If the student-user must provide a textual response to a 'HAS PART' teaching goal, the teaching strategy uses the spatial relationships for the SYM that the video is paused on to formulate a question. This question will ask the student-user to name the part that is located spatially next to another object (or objects), e.g. inside, to the left, between, above and touching, and so on. The question in Figure 4.25 is of this type.

Figure 4.26 shows ARISTOTLE using multimedia-based teaching strategy 2. This teaching strategy uses the procedures of the domain module to retrieve two alternative shots. It then presents these two shots to the student-user, together with the shot that depicts the teaching goal in question. The student-user is then invited to either: (1) click on an animal part to elicit a response, or (2) choose the video or audio that best depicts the goal in question.



ARISTOTLE

Watch the following three videos. Click on the cheetah's long legs.

Figure 4.26 ARISTOTLE uses multimedia-based teaching strategy 2.

4.3 SUMMARY

This chapter has discussed the use of the method in the development of a prototype interactive instructional MMIS called ARISTOTLE that uses the full-scale semantic content-based model. After discussing the system's development and architecture, the chapter discussed the functionality and behaviour that the system exhibits in relation to its architecture.

The following chapter discusses how ARISTOTLE implements and uses the seven semantic aspects of video and audio.

Chapter Five

IMPLEMENTATION OF THE SEVEN SEMANTIC ASPECTS OF VIDEO AND AUDIO WITHIN ARISTOTLE

"Aristotle maintained that women have fewer teeth than men; although he was twice married, it never occurred to him to verify this statement by examining his wives' mouths."

— Bertrand Russell, *Impact of Science on Society* (1952)

Through the implementation of the full-scale semantic content-based model, and the use of the seven semantic aspects that the model provides for, an interactive MMIS enhances its use of video and audio, compared to systems that do not use the model. The previous chapter discussed the development, architecture and functionality of ARISTOTLE, an interactive instructional MMIS that uses the full-scale semantic content-based model.

This chapter discusses the implementation and use of the seven semantic aspects, both individually and collectively, within ARISTOTLE. It is organised into two main sections. The following section discusses how each of the seven semantic aspects are implemented within ARISTOTLE's architecture, and presents the individual benefits that arise from each aspect's use. Then, the chapter argues that the consolidated implementation and use of all seven of the semantic aspects in ARISTOTLE is required in order for the system to function as it does.

5.1 DISTINCT BENEFITS ARISING FROM EACH OF THE SEVEN SEMANTIC ASPECTS

Each of the seven semantic aspects of video and audio offers unique benefits to the interactive MMIS that uses them. This section takes each of the seven semantic aspects in turn and discusses their implementation within ARISTOTLE's architecture, together with the benefits that arise from their use within the system. Table 5.1 provides an overview and summary of the discussion.

5.1.1 Explicit media structure

The video footage for ARISTOTLE was filmed in PAL and captured at 15 fps (frames per second). Thus, the video frame duration, t_{VD}, is $1/15$ s. Audio frames are also assumed to occur at a rate of 15 fps. Thus, one audio frame occurs every 66.67 ms, such that $t_{AD} = 1/15 \text{ s} = t_{VD}$. Frames are grouped into shots that are associated with particular content-based information within ARISTOTLE's *SEM*s.

No distinction is made between audio and video shots within the shots in the *SEM*s. Both video and audio shots are therefore stored in the same manner, i.e. "clipname":*i-j*. It is the perspective within the *SEM* that the shot is defined under that prescribes whether video and/or audio should be used. For example, the use of shots within a NOISE perspective dictates the use of just audio, even if the shot is actually one defined within a video and audio clip. This enables ARISTOTLE to use just the audio stream contained within a video clip, without the need to also present the video.

Table 5.1 The seven semantic aspects within ARISTOTLE.

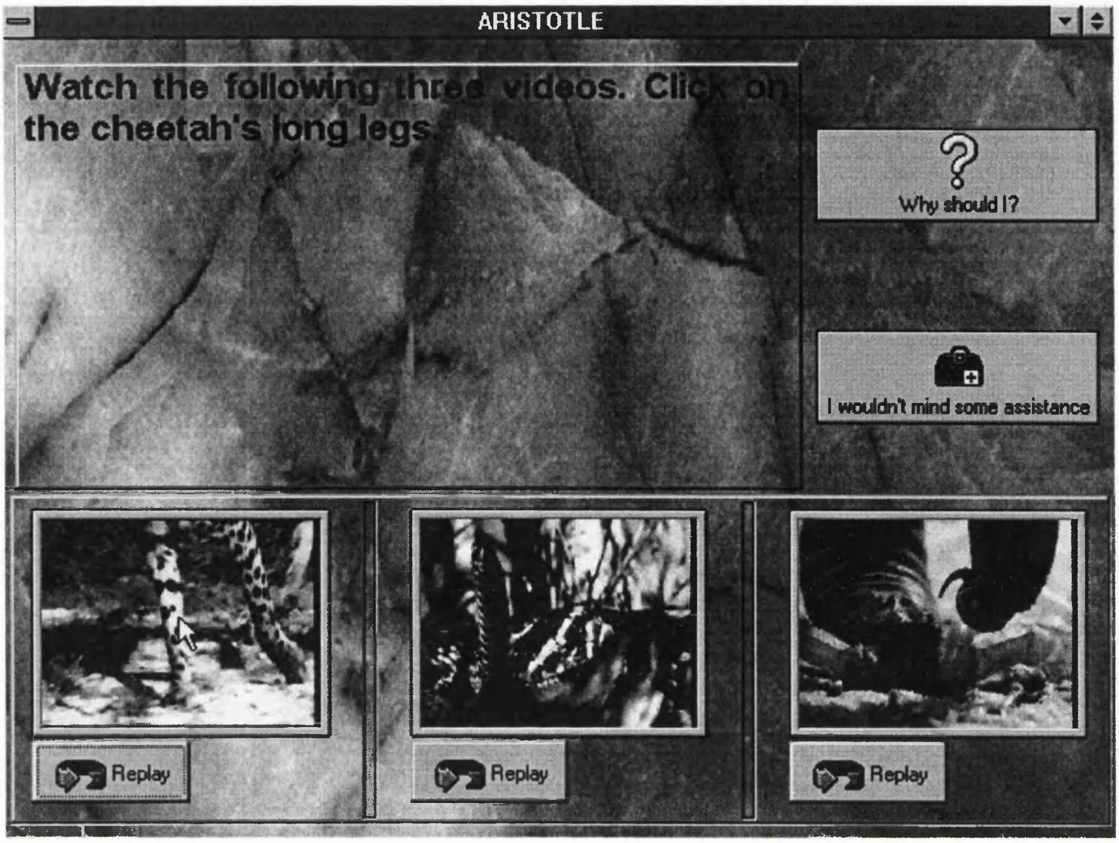
Semantic aspect	Implementation in ARISTOTLE	Benefits from use within ARISTOTLE
Explicit media structure	$t = 1/15$ s, frames grouped into shots associated with content-based information in *SEMs*.	• Separate use of video and audio streams. • Clear representation of media structure, resulting in quick determination of suitable shots.
Objects	Via the OBJECTS slot in the *SYMs*.	• Can ask student-user to click on particular animal part to register recognition. • Student-user can ask questions about an on-screen object by clicking on it. • Misconceptions related to incorrect recognition of objects may be diagnosed.
Spatial relationships between objects	Via the SPATIALRELS slot in the *SYMs*.	• Can teach about the relative location of certain objects in comparison to other objects. • Can ask student-user to name animal parts on-screen, without the need to reveal their identity. • Can ask student-user to identify objects that are not visible on-screen.
Events and actions involving objects	Via Events and Actions *SEMs*.	• Can use multimedia components to teach about animal behaviour as well as animal properties. • Can describe incidental activity taking place within the media stream.
Temporal relationships between events and actions	Determined from shots in Events and Actions *SEMs*.	• Can teach simultaneously about events/actions that occur at the same time. • Can notify student-user when an event/action they have named occurs before or after the current event/action(s) being taught about. • Can inform the student-user of what is happening within an event shot as the actions occur, or whenever the student-user asks.
Integration of syntactic and semantic information	Via combined use of *SEMs* and *SYMs*.	• Can combine domain knowledge with knowledge of the media stream, thus can use videos/audios that are relevant to the current concepts being taught.
Direct user-media interaction	Via combined use of *SEMs* and *SYMs*.	• Enables interactive teaching with videos/ audios. • Enables student-led learning through interaction with the media stream.

The shots within a single clip are permitted to intersect with each other, thus providing multiple content perspectives. For example, the shots for the constituent actions of an event intersect with the shots for that event. This simple media structure of frames and shots enables ARISTOTLE to quickly determine which shots are suitable for assisting in the achievement of a particular teaching or remedial goal. A complex hierarchy, e.g. where shots are aggregated into further structures, would have meant that ARISTOTLE would have had to search downwards in the hierarchy before finding the particular shot required.

5.1.2 Objects

Objects are represented within the OBJECTS slot of the SYMs. Animals are the entities of interest to ARISTOTLE and so the video and audio footage was captured and edited carefully so that there are as few objects in the footage as possible which are not relevant pedagogically. In other words, there are few objects in the SYMs which are not animals or animal parts. However, some objects modelled are those which are unrelated to animals, e.g. a tree in the background of a video frame, in order for ARISTOTLE to have a more complete understanding of the media streams.

Information within the SYMs about object presence and location within video frames enables ARISTOTLE to ask the student-user to click on a particular animal part within a video frame in order to register their recognition of the object. Figure 5.1 provides an example of ARISTOTLE asking the student-user to click on the long legs of a cheetah.

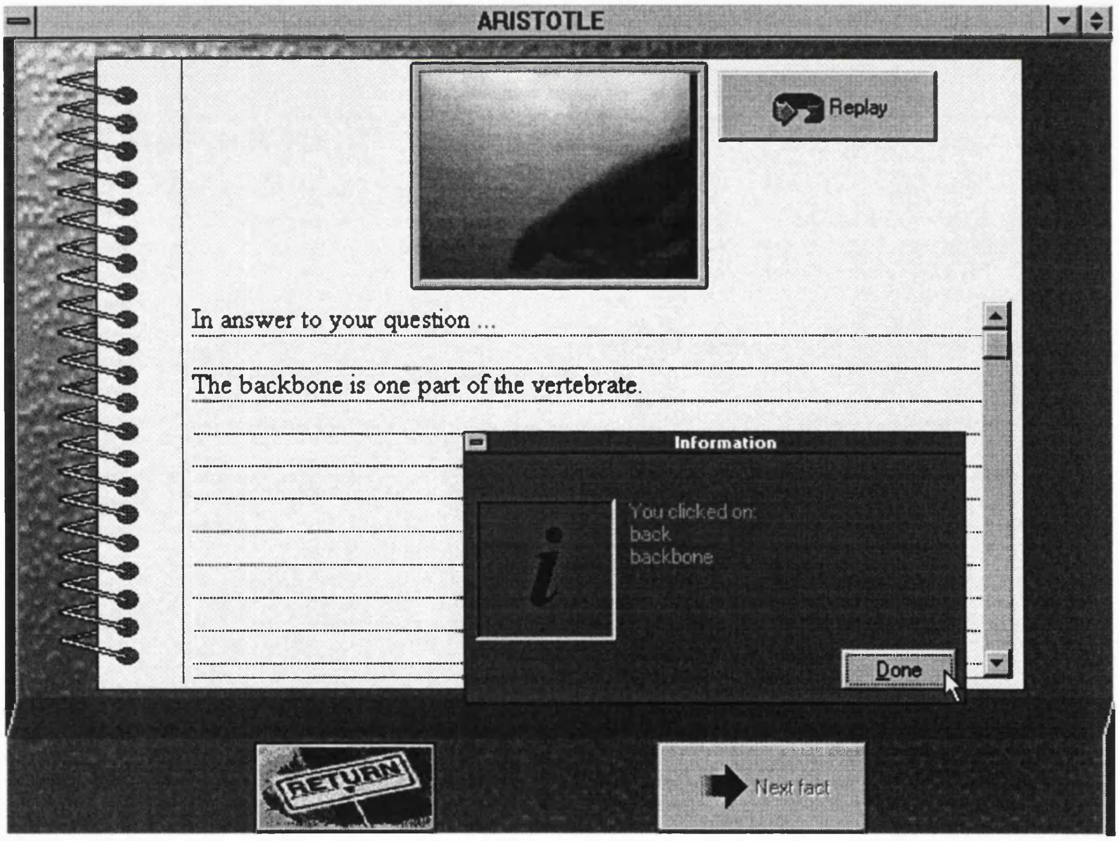


Watch the following three videos. Click on the cheetah's long legs.

Figure 5.1 ARISTOTLE uses the 'objects' semantic aspect during system-led teaching.

Through the associated object co-ordinates, ARISTOTLE can determine whether the object clicked on is indeed the correct one. If it is not the correct object, ARISTOTLE is able to store the incorrect object's name as a student misconception. The modelling of objects incidental to the teaching goals, such as trees, plants, and so forth, provides ARISTOTLE with full details of a misconception. Without this, ARISTOTLE would only know that the object clicked on is not the right one, but would not know exactly *what* that object was. For example, if the student-user were to click on the grass in the first video in Figure 5.1, ARISTOTLE would be able to determine this from the *SYM* for the associated video frame, and thus store 'grass' as a misconception in the student misconception *SEM*.

Such information also makes it possible for the student-user to ask questions about a particular object that is present on-screen by clicking on it with the cursor. ARISTOTLE can then use the object co-ordinates for the *SYM* associated with the relevant video frame to determine which object was clicked on and then react accordingly. Figure 5.2 shows a situation where the student-user is being presented with a video depicting a vertebrate's backbone. The student-user has clicked on an area of the screen in order to find out the name of an object. They are then told that they clicked on the vertebrate's back and backbone (since the backbone is inside the back).



In answer to your question ...

The backbone is one part of the vertebrate.

You clicked on:
back
backbone

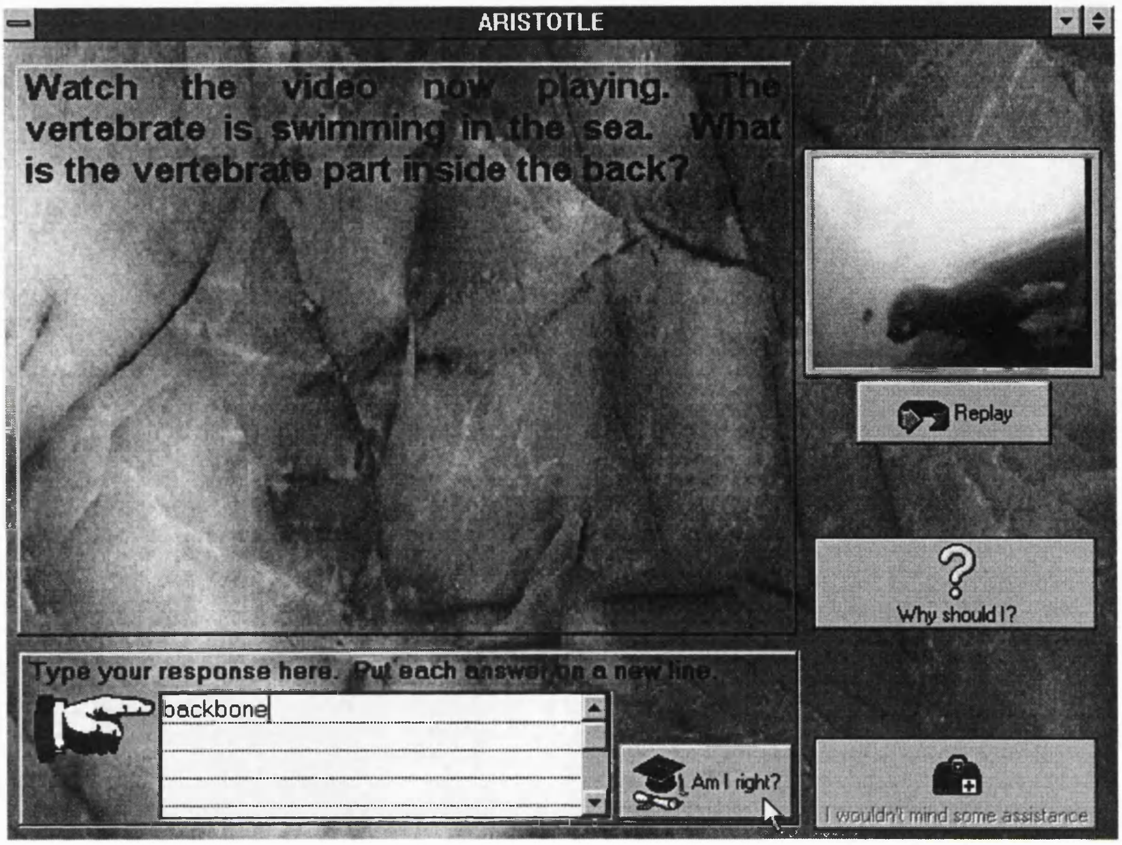
Figure 5.2 ARISTOTLE uses the 'objects' semantic aspect during student-led teaching.

5.1.3 Spatial relationships between objects

Spatial relationships between objects are represented within the SPATIALRELS slot of the SYMs. Because spatial relationships are used by ARISTOTLE in the phrasing of questions to the student-user, those spatial relationships that would be advantageous for such a task were modelled. For example, it was not necessary to record the fact that one tree was situated to the left of another tree, since ARISTOTLE would not ask the student-user to identify a tree in a video sequence because trees are not animal parts.

The modelling of spatial relationships within the SYMs allowed ARISTOTLE to phrase questions based on the position of objects. This enabled ARISTOTLE to teach about the relative location of certain objects in comparison to other objects, e.g. teaching that a cheetah's claws are situated at the end of their long legs by asking the student-user to name the animal part that is below and touching the long legs. It also enabled ARISTOTLE to ask the student-user to name animals or animal parts that are on-screen, without the need to reveal their identity.

The spatial relationships also enable ARISTOTLE to ask the student-user to identify objects that are not visible on-screen. For example, asking the student-user to name the body part that is located inside the backs of vertebrates. This is illustrated in Figure 5.3.



ARISTOTLE

Watch the video now playing. The vertebrate is swimming in the sea. What is the vertebrate part inside the back?

Type your response here. Put each answer on a new line.

backbone

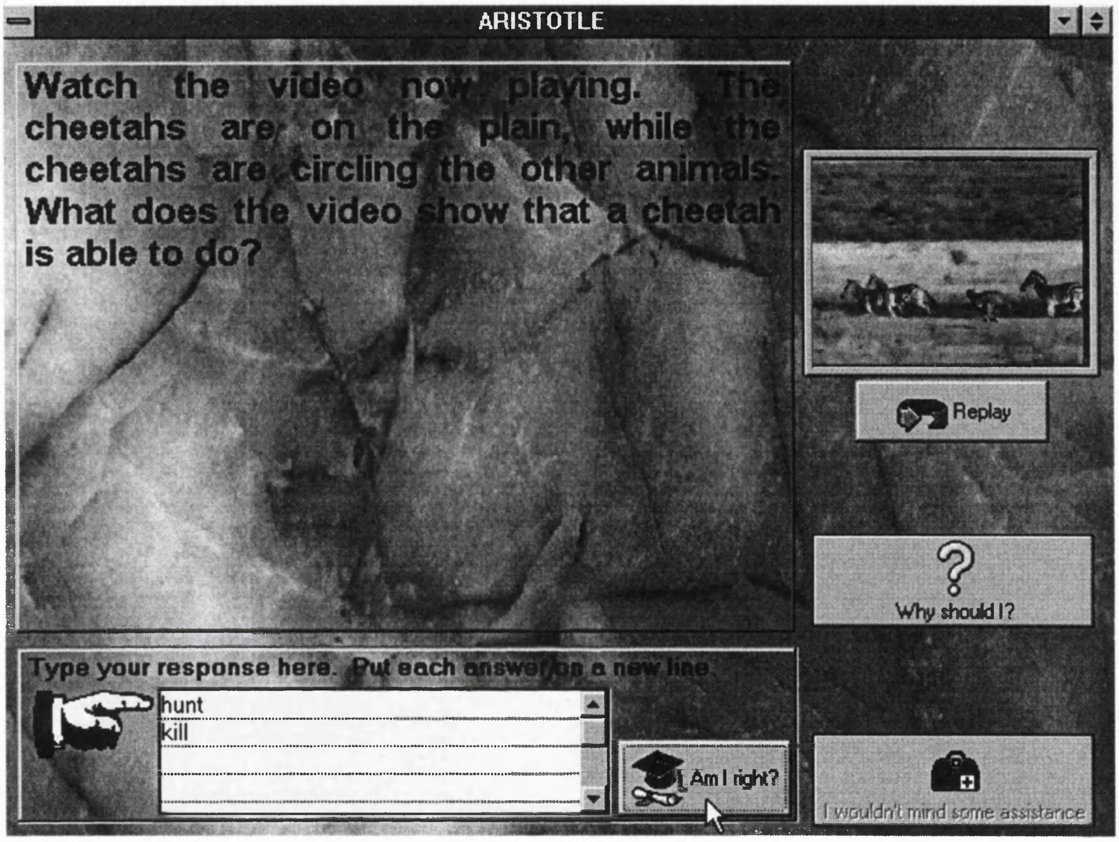
Figure 5.3 ARISTOTLE uses the 'spatial relationships between objects' semantic aspect.

5.1.4 Events and actions involving objects

Events and actions involving objects are implemented as Events *SEM*s and Actions *SEM*s, respectively. Events described what animals were able to do, such as 'hunt' in the case of cheetahs, and 'bite' in the case of rattlesnakes, via an ABLE TO perspective. The events are then split up into a number of actions, which characteristically make up such events. For example, 'observing prey', 'testing for slow prey', 'catching prey', 'suffocating prey', and 'chasing prey' in the case of cheetahs hunting.

These events and actions involved animals and so, whenever possible, shots were associated with the events and actions so that multimedia-based teaching strategies could be used for teaching. This enabled ARISTOTLE to use multimedia components to teach about animal behaviour as well as to teach about animal properties.

Figure 5.4 shows ARISTOTLE using the events and actions semantic aspect to ask the student-user about the events that the cheetah in the full-motion video sequence is involved in.



Watch the video now playing. The cheetahs are on the plain, while the cheetahs are circling the other animals. What does the video show that a cheetah is able to do?

Type your response here. Put each answer on a new line.

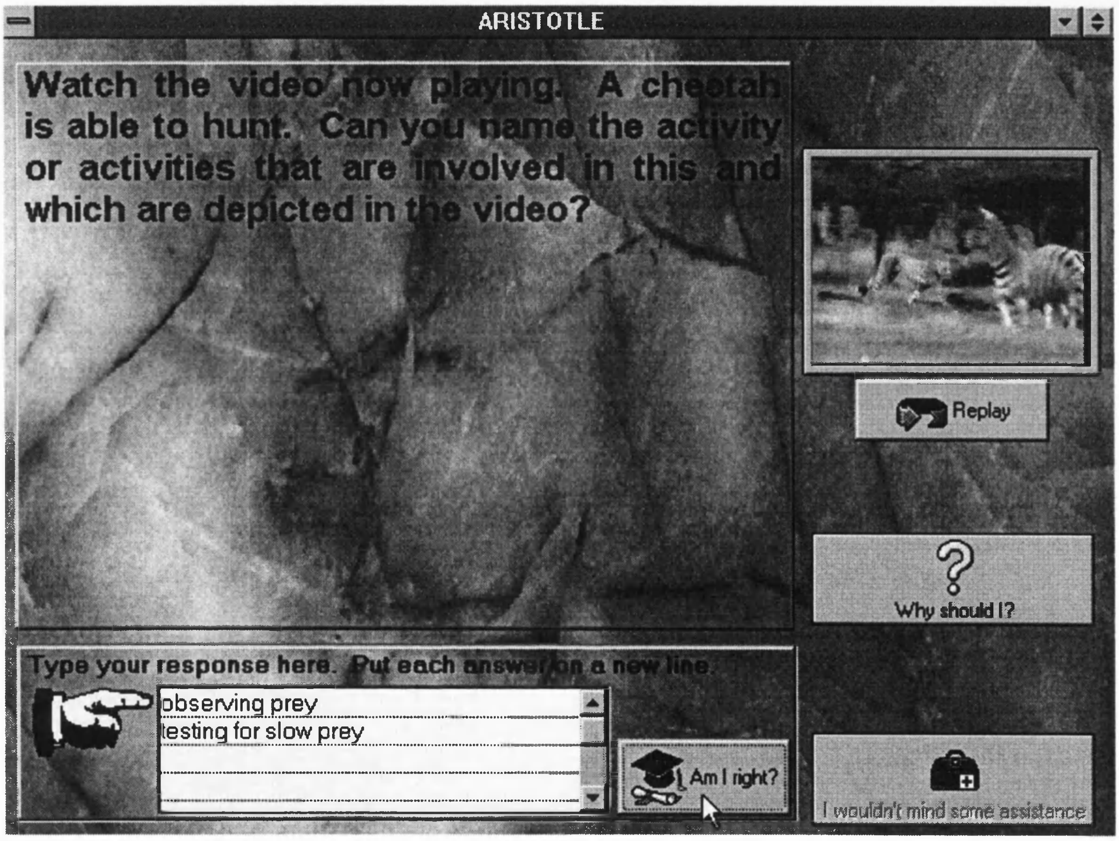
hunt
kill

Figure 5.4 ARISTOTLE uses the 'events and actions involving objects' semantic aspect.

5.1.5 Temporal relationships between events and actions

Events within ARISTOTLE are taught and presented according to their temporal order within the media stream. For example, if two events within an Events *SEM* occur simultaneously within the media stream (as with the hunt and kill events in the Cheetah Events *SEM*), ARISTOTLE teaches about the two events at the same time. This was illustrated in Figure 5.4.

Actions are treated in a similar fashion. The shots that are associated with each constituent action of an event enable ARISTOTLE to deduce which actions occur before, after, or during which others. Thus, actions which occur simultaneously are taught about simultaneously within ARISTOTLE. Figure 5.5 shows ARISTOTLE using the temporal relationships to teach about two actions that occur simultaneously within a hunting event.


Watch the video now playing. A cheetah is able to hunt. Can you name the activity or activities that are involved in this and which are depicted in the video?

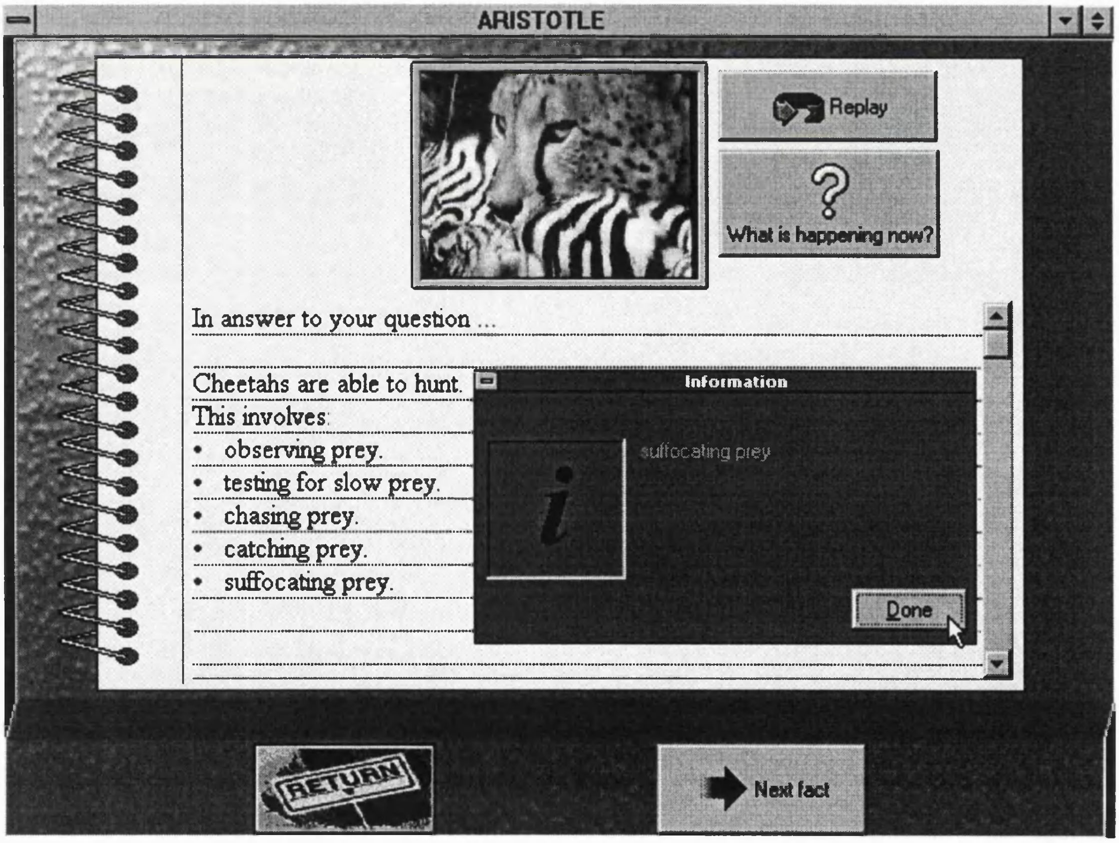
Type your response here. Put each answer on a new line.

observing prey
testing for slow prey

Figure 5.5 ARISTOTLE uses the 'temporal relationships between events and actions' semantic aspect during system-led teaching.

ARISTOTLE uses the temporal relationships between the actions to determine whether an action that the student-user has named actually occurs before or after the group of actions that is currently being taught about. This knowledge is used to inform the student-user of their error during feedback.

When the student-user asks questions that are related to events (e.g. "What is a cheetah able to do?"), ARISTOTLE provides the answer by playing an appropriate shot that depicts the event, and then naming the actions as and when they occur during the presentation of the shot. At any time, the student-user may also interrupt the presentation of the shot to ask which actions are occurring within the shot at that precise moment. ARISTOTLE uses the shots associated with the actions in the Actions *SEM* to determine the order in which the actions in the media stream occur. Figure 5.6 shows ARISTOTLE detailing the actions that constitute a hunting event for cheetahs.



In answer to your question ...

Cheetahs are able to hunt.
This involves:
- observing prey.
- testing for slow prey.
- chasing prey.
- catching prey.
- suffocating prey.

suffocating prey

Figure 5.6 ARISTOTLE uses the 'temporal relationships between events and actions' semantic aspect during student-led teaching.

5.1.6 Integration of syntactic and semantic information

Integration of syntactic and semantic information is achieved through the combined use of SEMs and SYMs. In the case of ARISTOTLE, this occurs through the use of the multimedia-based teaching strategies, and through student-led teaching. Multimedia-based teaching strategy 1 uses this semantic aspect by combining the semantic information regarding animals and their properties with the syntactic information regarding their spatial location in relation to other animal properties and non-animal objects. Figure 5.3 provides one example of this, where the semantic information that a vertebrate has a backbone is combined with the syntactic information that the backbone is located in the back.

Multimedia-based teaching strategy 2 also uses an integration of syntactic and semantic information. Figure 5.1 provides an example of this, where the semantic information that cheetahs have long legs is used in combination with the syntactic information regarding the co-ordinates of the object (i.e. the long legs) within the video frame, so that the student-user is asked to click on the long legs to register their response.

During student-led teaching, answers to the student-user's questions are provided by using the semantic information contained within the SEMs together with suitable multimedia content. When the student-user clicks on an object located in the video presented to them, ARISTOTLE uses an integration of syntactic and semantic information to provide the name of the object. This was illustrated in Figure 5.2.

This integration of syntactic and semantic information links domain knowledge, and thus teaching goals, with the content represented within video and audio streams. This enables the full use of videos and audios that are relevant to the current concepts being taught.

5.1.7 Direct user-media interaction

Direct user-media interaction is achieved through the combined use of SEMs and SYMs. In the case of ARISTOTLE, this occurs through the use of the multimedia-based teaching strategies, and through student-led teaching. Both multimedia-based teaching strategies use direct user-media interaction, for the HAS PART perspective, to elicit responses from the student-user. These teaching strategies use the information contained with the SEMs to determine *when* the student-user has interacted, and the information contained with the SYMs to determine *what* the student-user has interacted with. Figure 5.1 provides an example of this within multimedia-based teaching strategy 2. ARISTOTLE also uses the SEMs and SYMs in this manner when the student-user enquires about the media stream presented to him or her. An example of this was provided in Figure 5.2.

The student-user may also interact with the media stream during student-led teaching of events and actions, by asking which actions are happening at the moment of interaction. In Figure 5.6, the student-user is being presented with a shot depicting a cheetah hunting. The student-user has clicked the 'What is happening now?' button, and is told that the cheetah is currently suffocating its prey.

Direct user-media interaction enables ARISTOTLE to use interactive teaching with videos and audios, rather than merely asking the student-user to identify objects that they have recognised by typing their names into an input line. It also enables ARISTOTLE to offer interactive student-led learning through inquisitive student-user interaction with the media stream by, for example, clicking on objects with the cursor in order to find out what they are.

5.2 JUSTIFYING THE CONSOLIDATED USE OF THE SEVEN SEMANTIC ASPECTS

In the previous section, the distinct benefits of implementing and using the seven semantic aspects of video and audio within ARISTOTLE were discussed. This section argues that ARISTOTLE is only able to function as it does because of its *collective* implementation and use of all seven of the semantic aspects.

In order for the student-user to interact with a given media stream, the media stream must have a media structure that can be manipulated by the system. The moment that the interaction occurred can then be mapped onto a point within the defined media structure.

In order to know *what* has been interacted with, the system must also have knowledge of the presence of objects, and their on-screen location (through the use of co-ordinates), for each point within the explicit media structure. The representation of content-based object information relies on the existence of an explicit media structure, so that the precise moments at which objects are present within the media stream can be modelled.

Spatial relationships provide details of relative location and are needed for interaction with objects that are not physically visible on-screen. Thus, the on-screen location for hidden objects may be determined from the spatial relationships. However, the spatial relationships rely on the knowledge the system has of the objects within the media stream, since a spatial relationship can only be defined between two distinct objects.

In order to understand the context of interaction, the system must have knowledge of the events and actions occurring within the media stream at the time of interaction. For this to be possible, the events and actions must be tied to the explicit media structure, so that the system can determine which shots depict which events and actions.

To know which events and actions occur before, after, or during the moment of interaction, knowledge of the temporal relationships between the events and actions is also required. However, the existence of temporal relationships between events and actions is dependent upon an effective modelling of events and actions, and an explicit media structure. Thus, the temporal relationships between events and actions can be determined from the explicit media structure that events and actions have been defined on.

For the system to link the knowledge of what has been interacted with and the context of interaction, integration between syntactic and semantic information must exist. Then, the system is able to deduce the relevance of one fragment of semantic information to one fragment of syntactic information (and vice versa). If the two are unrelated, then the context of interaction cannot be related to a point within the media structure, or an object on-screen. The integration between syntactic and semantic information is bound by the integration between: (1) the events and actions, and the temporal relationships between them, and (2) the objects, and the spatial relationships between the objects. This integration is provided by the explicit media structure, which ties the semantic and syntactic representations to a common time-based representation.

Therefore, all seven semantic aspects are necessary for enabling full interaction with video and audio. If one or more of the semantic aspects were removed, the functionality of the MMIS would be restricted. For example, without a representation of objects, spatial relationships between objects cannot be determined, and thus there is no representation of syntactic information. It is not, therefore, possible to integrate syntactic and semantic information. Consequently, the MMIS cannot determine what has been interacted with, and it is unable to determine the context of interaction.

Similarly, if the explicit media structure were not present, the precise moments at which objects were depicted in the media stream could not be modelled. Thus, spatial relationships between objects could not be determined. Without the explicit media structure, events and actions could not be defined on specific shots, and temporal relationships between events and actions could not be determined. Consequently, syntactic and semantic information would not exist and could not, therefore, be integrated. The MMIS would then be unable to provide direct user-media interaction since it would be unable to determine what had been interacted with and the context of the interaction.

5.3 SUMMARY

This chapter has discussed the implementation and use of the seven semantic aspects of video and audio within ARISTOTLE. The implementation and use of each semantic aspect within ARISTOTLE has revealed that each semantic aspect offers unique benefits to the system that uses it. However, it was argued that it is only through the consolidated implementation and use of all of the seven semantic aspects that direct user-media interaction is enabled. Hence, while an interactive MMIS may implement only a select few of the seven semantic aspects, the implementation of all seven results in a system that can use video and audio to their full potential.

The following chapter concludes this thesis with a summary of each chapter. Then it discusses the two contributions believed to have been made by this research, namely the full-scale semantic content-based model, which was proposed in Chapter

Two, and the method for developing interactive MMISs that use the model, which was proposed in Chapter Three. Finally, the chapter identifies areas for further research and development.

Chapter Six

CONCLUDING DISCUSSION

"Only a little more
I have to write,
Then I'll give o'er
And bid the world good-night."
—Robert Herrick, 'His Poetry his Pillar'

This thesis has proposed a full-scale semantic content-based model for interactive MMISs that caters for the seven semantic aspects of video and audio. A method for the development of interactive MMISs that use the model was also discussed. Both the model and the method were demonstrated through the development of ARISTOTLE, an interactive instructional MMIS that teaches young children about zoology. The previous chapter discussed the implementation and use of the seven semantic aspects within ARISTOTLE, both individually and collectively.

This chapter begins by summarising the previous five chapters of the thesis. Then, it discusses the two contributions made by this thesis: the full-scale semantic content-based model, and the method for the development of interactive MMISs that uses the model. The chapter concludes by discussing future research and development that arises from this thesis.

6.1 THESIS SUMMARY

Chapter One began the thesis by distinguishing between the syntax and semantics of video and audio in multimedia information systems. It was argued that issues of syntax have dominated research within the field of multimedia. However, multimedia content-based semantics must be considered in order for video and audio to be used and interacted with. Existing research on semantic content-based modelling was then reviewed within four groups: physical models, techniques for locating content objects, stratification-based techniques, and formal techniques. This review highlighted seven semantic aspects of video and audio: (1) explicit media structure, (2) objects, (3) spatial relationships between objects, (4) events and actions involving objects, (5) temporal relationships between events and actions, (6) integration of syntactic and semantic information, and (7) direct user-media interaction. Weaknesses in how existing research had addressed these seven semantic aspects were revealed. Moreover, it was found that none of the models encompassed all seven aspects.

Chapter Two presented a full-scale semantic content-based model that caters for all seven of the semantic aspects of video and audio. To achieve this, the model uses the multimedia frame, or m-frame, as the representation framework that stores the syntactic and semantic content-based information about the video and audio. Two types of m-frames are used. Syntactic m-frames (SYMs) model the syntactic content of the video and audio, and represent objects and spatial relationships between objects. Semantic m-frames (SEMs) model the semantic content of the video and audio, and model descriptions, events, and actions associated with the entity of interest. Both m-frames were described in detail. The chapter concluded by discussing how the model catered for the seven semantic aspects of video and audio.

Chapter Three presented a method for developing interactive MMISs that encompass the full-scale semantic content-based model. The method is composed of seven stages: (1) construct Description matrix for entities of interest to the system, (2) construct Temporal Objects/Events and Actions matrices, (3) collect raw video and audio footage, (4) construct annotated spatial network diagrams for video clips, (5) implement *SYMs*, (6) implement *SEMs*, and (7) implement multimedia support environment. The chapter described in detail the tasks to be undertaken in each of these stages. It also described three front-end software tools that were developed to assist with the development process: the Clip Manager for the management of clips within the multimedia resource, and the SYMulator and SEMulator to facilitate the creation of *SYMs* and *SEMs*.

Chapter Four discussed the use of the method in the development of ARISTOTLE, an interactive instructional MMIS that uses the full-scale semantic content-based model. ARISTOTLE's development within each stage of the method was described. Then, the chapter presented the architecture of the system, and discussed the domain, tutor, and student modules of the architecture in detail. The chapter concluded by discussing the behaviour of ARISTOTLE in relation to this architecture.

Chapter Five discussed the implementation and use of the seven semantic aspects of video and audio within ARISTOTLE. It began by discussing how each of the seven semantic aspects were implemented within ARISTOTLE's architecture, and the individual benefits that arose from the use of each aspect. Then, it discussed the combined use of the seven semantic aspects, and argued that all seven aspects are required in order for the system to function as it does.

6.2 THESIS CONTRIBUTIONS

This thesis makes two contributions: the full-scale semantic content-based model, and the method for the development of interactive MMISs that use the model. This section discusses each of these contributions in turn.

6.2.1 The full-scale semantic content-based model

The full-scale semantic content-based model caters for the seven semantic aspects of video and audio, namely: (1) explicit media structure, (2) objects, (3) spatial relationships between objects, (4) events and actions involving objects, (5) temporal relationships between events and actions, (6) integration of syntactic and semantic information, and (7) direct user-media interaction. To achieve this, the model adopts an entities of interest approach where the relevant semantic content-based information about video and audio is organised around, and integrated with information about, the entities of interest to the system.

The model is based on an explicit media structure that divides both video and audio into frames. These frames are grouped into shots, where a shot is defined as an arbitrary sequence of contiguous frames that have continuity of meaning in time. A shot is formally expressed by a pair $[i, j]$, where i is the starting frame of the shot and j is the ending frame.

The multimedia frame (m-frame) is used as the representation framework that stores the syntactic and semantic content-based information within the model. A syntactic m-frame (SYM) consists of an audio frame, a video frame, and a syntactic information component. The syntactic information component models the syntactic content of the associated video frame, in terms of objects present in the frame, together with their on-screen co-ordinates, and spatial relationships between those objects. It consists of three slots: the FRAMENO slot stores the frame number of the related audio and video, the OBJECTS slot stores the names and on-screen co-ordinates of the pertinent objects present within the associated video frame, and the SPATIALRELS slot stores the spatial relationships between those objects. Nine primitives are used to model the spatial relationships: *touches* (=), *above* (↑) and *beneath* (↓), *inside* (⊆) and *outside* (⊇), *left* (<) and *right* (>), and *before* (⇑) and *behind* (⇓).

Semantic m-frames (SEMs) model information about the semantic content of shots that are related to an entity of interest to the MMIS using the model. Each slot within a SEM represents a particular perspective on the multimedia content. Each slot value represents a more specific instance of the perspective. The perspectives and instances are defined by the domain of discourse of the MMIS. Shots are associated with instances. Shots are permitted to intersect with each other, thus permitting overlaps of content representation. SEMs are therefore not restricted to representing only one view of the content depicted within media streams.

Three SEMs collectively model an entity of interest. The Description SEM describes the entity of interest. Its perspectives are therefore description-oriented, e.g. HAS PART. The Events SEM models the events that are associated with the entity of interest. Its perspectives are event-oriented, e.g. ABLE TO, and serve to group together one or more events that are modelled as instances. The Actions SEM models the constituent actions of the events represented in the Events *SEM*. Each perspective within the Actions *SEM* therefore corresponds to an instance within the Events *SEM*. Thus, a perspective serves to group together the actions that constitute an event, which are modelled as instances. Consequently, the shots segment each of the event shots into specific actions.

The syntactic and semantic m-frames together form the full-scale semantic content-based model, as illustrated by Figure 2.5. The *SYMs* of the model are timely because they maintain the original continuity of the media stream, and thus their ordering is important. The *SEMs*, however, are timeless, since they model content in a manner unrelated to the original placement of shots within the media stream.

The model is representational, and not computational, and thus provides a structured representation of the information needed for the MMIS using the model to use the seven semantic aspects of video and audio. The model enables the MMIS to understand the content of the media currently being used, and thus to use media according to its goals and objectives. More specifically, the model enables the MMIS to know what is depicted within the media at a specific moment in time, to know what is being interacted with at a specific moment in time, to know the relative context of the interaction (i.e. what else has, is, and will be going on within the media), and to know which media have footage of the object, events, or actions currently being used within the system.

The full-scale semantic content-based model caters for the seven semantic aspects as follows: the explicit media structure is composed of audio and video frames, which are grouped into shots; objects are represented within the OBJECTS slot of the SYMs; spatial relationships are represented within the SPATIALRELS slot of the SYMs; events and actions involving objects are represented by perspectives and instances within the Events and Actions SEMs; temporal relationships between events and actions may be determined from the shots associated with the events and actions; the integration of syntactic and semantic information is provided by the tight linking between SEMs and SYMs within the model, which are tied to a common explicit media structure; and direct user-media interaction is provided by the combination of SEMs and SYMs, which together provide details of what has been interacted with, and the current context of interaction.

ARISTOTLE, an interactive instructional MMIS, was developed to demonstrate that the full-scale semantic content-based model is implementable.

6.2.2 The method for the development of an interactive MMIS that uses the full-scale semantic content-based model

The method consists of seven stages that prescribe the tasks to be undertaken in the development of an interactive MMIS that uses the full-scale semantic content-based model. These stages take the developer from the initial description of entities of interest to the system, through to the implementation of the multimedia support environment that uses the model. The development of ARISTOTLE demonstrates that the method is workable in practice. The seven stages of the method are as follows:

Stage 1: Construct Description matrix for entities of interest to the system. The first stage of the method involves the identification and description of entities of interest to the system. First, the entities of interest are identified and organised into a structure, such as flat, hierarchical, or networked. Entities of interest are then described on a Description matrix according to various description-oriented perspectives.

Stage 2: Construct Temporal Objects/Events and Actions matrices. The second stage of the method involves the construction of Temporal Objects/Events and Actions (TOEA) matrices for each entity of interest identified during Stage 1. The TOEA matrix plots objects against the events and actions involving those objects. It is temporal because it indicates the sequence in which events and actions occur.

Stage 3: Collect raw video and audio footage. Once the entities of interest have been described, and their associated events and actions identified and ordered, the developer is now in a position to collect the raw video and audio footage. The Description matrix indicates which properties of the entities of interest are to be filmed. The TOEA matrices indicate the events and actions for which video and audio footage is required, and provide the objects that must be included in the filming of each event and action. Once the footage is collected, it is captured digitally and edited into clips. A front-end software tool, such as the Clip Manager, may be used to assist with the management of clips.

Stage 4: Construct annotated spatial network diagrams for video clips. The developer is now in a position to model the spatial relationships between the objects depicted in the captured video clips. Annotated spatial network diagrams are used to conceptually represent the spatial relationships. These diagrams represent each object as a node, and link one object to another through the use of arcs, which are annotated with the spatial relationships that exist between the objects. To construct the diagrams, the video clips created in Stage 3 are played, and one diagram is created for each video frame in which the spatial relationships differ from that of the previous video frame.

Stage 5: Implement SYMs. The annotated spatial network diagrams are now used to create the SYMs, together with the addition of on-screen co-ordinates for objects modelled within the diagrams. The SYMs may be implemented directly by the developer, or a front-end software tool, such as the SYMulator, may be used.

Stage 6: Implement SEMs. Implementation of the SEMs is based on the Description matrix (Stage 1) and the TOEA matrices (Stage 2). The Description matrix is used as the basis for the Description SEMs, and the TOEA matrices are used as the basis for the Events and Actions SEMs. A front-end software tool, such as the SEMulator, may be used to assist with the creation of the SEMs.

Stage 7: Implement multimedia support environment. The final stage of the method is to implement the multimedia support environment that will use the model. Tasks within this stage are determined by the type of MMIS being developed. In the case of ARISTOTLE, this stage was concerned with the development of the domain, tutor, and student modules typical of an interactive instructional MMIS.

6.3 FUTURE RESEARCH AND DEVELOPMENT

This section discusses further work that arises from this thesis, and which will be the subject of future research and development. At the moment, ARISTOTLE is a prototype, which contains enough knowledge to demonstrate sufficiently the full-scale semantic content-based model. The development of a fully functional interactive instructional MMIS was beyond the scope of the model, since a larger knowledge base would not contribute further to the demonstration of the model's implementation. Thus, one area of further development will be to increase the size of ARISTOTLE's domain knowledge in order for the system to teach about a more substantial number of animals and animal classes. In addition, because only some of the video footage within

ARISTOTLE contains commentary, this development would also include the addition of commentary to all video footage.

Although ARISTOTLE does permit the student-user to click on objects during the presentation of shots from Description SEMs, during student-led teaching, it does not permit the student-user to click on objects as events and actions are being displayed. Further development of the system will rectify this. ARISTOTLE also does not permit the student-user to 'jump' around the animal hierarchy. Further development will enhance ARISTOTLE's multimedia-based teaching strategies so that, for example, the student-user may click on an animal in a video as it is being presented, and then ask ARISTOTLE to switch to teaching about that animal.

Because many teaching strategies were not required to adequately demonstrate the full-scale semantic content-based model, ARISTOTLE currently has two multimedia-based and two non-multimedia-based teaching strategies. Further work on the system will increase the number of teaching strategies within ARISTOTLE's tutor module.

At the moment, ARISTOTLE communicates with the student-user through the use of text. Speech communication was not implemented as it lay beyond the scope of the thesis. However, further development will link ARISTOTLE to First Byte's Monologue program so that ARISTOTLE may also communicate to the student-user through speech.

Similarly, further development will link ARISTOTLE to Dragon Systems' Talk→To Plus program so that the student-user may communicate with ARISTOTLE through the use of speech, as well as the keyboard, and the mouse.

An object of further work will also be to research into existing techniques for spatial and temporal reasoning in order to enhance the use of spatial and temporal relationships within ARISTOTLE. An investigation of existing research in spatial and temporal reasoning was beyond the scope of this thesis and the full-scale semantic content-based model, since the objective was to develop a full-scale model that encompassed all seven semantic aspects.

Another area of further research and development will be to the SYMulator. Currently, the SYMulator ties the storage of object co-ordinates within the SYMs to the particular screen resolution that the program is run under. Further development of the SYMulator will seek to store the co-ordinates in a format that is independent of screen resolution.

Development will also be undertaken on help systems and user manuals for the Clip Manager, the SYMulator, and the SEMulator, so that they may be used by other multimedia researchers and professionals.

Further research will also seek to establish how well the full-scale semantic content-based model scales up. This will be carried out in two ways: (1) by using the model in large systems, both instructional and non-instructional types, and (2) by using the model in the indexing of large bodies of raw video and audio, such as entire television programmes and movies, when what will eventually be regarded as 'entities of interest' is not known in advance.

Finally, the method will be used for systems other than interactive instructional MMISs, such as digital encyclopaedias and Web applications, in order to further demonstrate its practicality.

REFERENCES

Adah, S., Candan, K. S., Chen, S.-S., Erol, K. and Subrahmanian, V. S. (1996). The Advanced Video Information System: data structures and query processing. *Multimedia Systems*, Vol. 4, No. 4, August, pp. 172-186.

Agius, H. W. (1996). Synthesising technology and context for instructional multimedia information systems within the primary classroom. In: Y. J. Katz, D. Millin and B. Offir (eds.), *The Impact of Information Technology: From Practice to Curriculum*. Chapman & Hall, London, pp. 95-100.

Agius, H. W. and Angelides, M. C. (1997a). Desktop video conferencing in the organisation. *Information & Management*, Vol. 31, No. 6, pp. 291-302.

Agius, H. W. and Angelides, M. C. (1997b). Integrating logical video and audio segments with content-related information in instructional multimedia systems. *Information and Software Technology*, forthcoming.

Aguierre Smith, T. G. and Davenport, G. (1992). The Stratification System: a design environment for random access video. In: *Proceedings of the Third International Workshop on Network and Operating Systems Support for Digital Audio and Video*. Springer-Verlag, New York, NY, pp. 250-261.

Aigrain, P., Zhang, H. J. and Petkovic, D. (1996). Content-based representation and retrieval of visual media: a state-of-the-art review. *Multimedia Tools and Applications*, Vol. 3, No. 3, November, pp. 179-202.

Alonso, F., de Antonio, A., Fuertes, J. L. and Montes, C. (1995). Teaching communication skills to hearing-impaired children. *IEEE MultiMedia*, Vol. 2, No. 4, Winter, pp. 55-67.

Angelides, M. C. (1995). Developing hybrid intelligent tutoring and hypertext systems. *The New Review of Hypermedia and Multimedia*, Vol. 1, pp. 67-106.

Angelides, M. C. and Demosthenous, A. (1996). Towards multimedia based training systems. In: Y. J. Katz, D. Millin and B. Offir (eds.), *The Impact of Information Technology: From Practice to Curriculum*. Chapman & Hall, London, pp. 123-128.

Angelides, M. C. and Dustdar, S. (1997). *Multimedia Information Systems*. Kluwer Academic Publishers, Boston, MA.

Bach, J. R., Paul, S. and Jain, R. (1993). A visual information management system for the interactive retrieval of faces. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, August, pp. 619-628.

Barber, R., Equitz, W., Faloutsos, C., Flickner, M., Niblack, W., Petkovic, D. and Yanker, P. (1995). Query by content for large on-line image collections. In: B. Furht and M. Milenkovic (eds.), *A Guided Tour of Multimedia Systems and Applications*. IEEE Computer Society Press, Los Alamitos, CA, pp. 357-378.

Berra, P. B., Golshani, F., Mehrotra, R. and Liu Sheng, O. R. (1993). Guest editors' introduction: Multimedia information systems. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, August, pp. 545-550.

Blattner, M. M. (1994). In our image: interface design in the 1990s. *IEEE MultiMedia*, Vol. 1, No. 1, Spring, pp. 25-36.

Bonarini, A. (1993). Modeling issues in multimedia car-driver interaction. In: M. T. Maybury (ed.), *Intelligent Multimedia Interfaces*. The AAAI Press / The MIT Press, Menlo Park, CA, pp. 353-371.

Bove, V. M., Jr., Granger, B. D. and Watlington, J. A. (1994). Real-time decoding and display of structured video. In: *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Boston, MA, May. IEEE Computer Society Press, Los Alamitos, CA, pp. 456-462. (Reprinted in: B. Furht and M. Milenkovic (eds.), *A Guided Tour of Multimedia Systems and Applications*. IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 123-129.)

Brand, S. (1988). *The Media Lab: Inventing the Future at MIT*. Penguin Books, New York, NY.

Brink, A., Marcus, S. and Subrahmanian, V. S. (1995). Heterogeneous multimedia reasoning. *Computer*, Vol. 28, No. 9, September, pp. 33-39.

Burrill, V., Kirste, T. and Weiss, J. (1994). Time-varying Sensitive Regions in dynamic multimedia objects: a pragmatic approach to content-based retrieval from video. *Information and Software Technology*, Vol. 36, No. 4, April, pp. 213-223.

Chang, S.-K. and Hsu, A. (1992). Image information systems: where do we go from here? *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, No. 5, October, pp. 431-442.

Christel, M., Kanade, T., Mauldin, M., Reddy, R., Sirbu, M., Stevens, S. and Wactlar, H. (1995). Informedia Digital Video Library. *Communications of the ACM*, Vol. 38, No. 4, April, pp. 57-58.

Csinger, A., Booth, K. S. and Poole, D. (1995). AI meets authoring: user models for intelligent multimedia. *Artificial Intelligence Review*, Vol. 8, No. 5/6, pp. 447-468.

Davis, M. (1993). Media Streams: an iconic visual language for video annotation. In: *Proceedings of the IEEE Symposium on Visual Languages*, Bergen. IEEE Computer Society Press, Los Alamitos, CA, pp. 196-202.

Department for Education (1995). *Science in the National Curriculum*. HMSO, London.

Dustdar, S. and Angelides, M. C. (1997). Organisational impacts of multimedia information systems. *Journal of Information Technology*, Vol. 12, No. 1, pp. 33-43.

Edwards, A. D. N. and Blore, A. (1995). Speech input for persons with speech impairments. *Journal of Microcomputer Applications*, Vol. 18, No. 4, October, pp. 327-333.

Feigenbaum, E. A. (1996). How the 'What' becomes the 'How'. *Communications of the ACM*, Vol. 39, No. 5, May, pp. 97-104.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. and Yanker, P. (1995). Query by image and video content: the QBIC system. *Computer*, Vol. 28, No. 9, September, pp. 23-32.

Fox, E. A. (1991). Advances in interactive digital multimedia systems. *Computer*, Vol. 24, No. 10, October, pp. 9-21.

Furht, B. (1994). Multimedia systems: an overview. *IEEE MultiMedia*, Vol. 1, No. 1, Spring, pp. 47-59. (Reprinted in: B. Furht and M. Milenkovic (eds.), *A Guided Tour of Multimedia Systems and Applications*. IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 4-16.)

Furht, B. and Milenkovic, M. (1995). Introduction. In: B. Furht and M. Milenkovic (eds.), *A Guided Tour of Multimedia Systems and Applications*. IEEE Computer Society Press, Los Alamitos, CA, pp. 1-3.

Golshani, F. and Dimitrova, N. (1994). Retrieval and delivery of information in multimedia database systems. *Information and Software Technology*, Vol. 36, No. 4, April, pp. 235-242.

Gong, Y., Chua, H. C. and Guo, X. (1996). Image indexing and retrieval based on color histograms. *Multimedia Tools and Applications*, Vol. 2, No. 2, March, pp. 133-156.

Goodman, B. A. (1993). Multimedia explanations for intelligent training systems. In: M. T. Maybury (ed.), *Intelligent Multimedia Interfaces*. The AAAI Press / The MIT Press, Menlo Park, CA, pp. 148-171.

Grosky, W. I. (1994). Multimedia information systems. *IEEE MultiMedia*, Vol. 1, No. 1, Spring, pp. 12-24.

Gudivada, V. N. and Raghavan, V. V. (1995). Guest editors' introduction: Content-based image retrieval systems. *Computer*, Vol. 28, No. 9, September, pp. 18-22.

Hemphill, C. T., Thrift, P. R. and Linn, J. C. (1996). Speech-Aware Multimedia. *IEEE MultiMedia*, Vol. 3, No. 1, Spring, pp. 74-78.

Hirzalla, N., Falchuk, B. and Karmouch, A. (1995). A temporal model for interactive multimedia scenarios. *IEEE MultiMedia*, Vol. 2, No. 3, Fall, pp. 24-31.

Jain, R. (1994a). Multimedia computing. *IEEE MultiMedia*, Vol. 1, No. 1, Spring, pp. 3-4.

Jain, R. (1994b). What is multimedia, anyway? *IEEE MultiMedia*, Vol. 1, No. 3, Fall, p. 3.

Jain, R. (1996). CAD for creative people. *IEEE MultiMedia*, Vol. 3, No. 1, Spring, pp. 3-4.

Kanade, T. (1996). Immersion into visual media: new applications of image understanding. *IEEE Expert*, Vol. 11, No. 1, February, pp. 73-80.

Kazman, R., Al-Halimi, R., Hunt, W. and Mantei, M. (1996). Four paradigms for indexing video conferences. *IEEE MultiMedia*, Vol. 3, No. 1, Spring, pp. 63-73.

Koons, D. B., Sparrell, C. J. and Thorisson, K. R. (1993). Integrating simultaneous input from speech, gaze, and hand gestures. In: M. T. Maybury (ed.), *Intelligent Multimedia Interfaces*. The AAAI Press / The MIT Press, Menlo Park, CA, pp. 257-276.

Le Gall, D. (1991). MPEG: a video compression standard for multimedia applications. *Communications of the* ACM, Vol. 34, No. 4, April, pp. 46-58. (Reprinted in: B. Furht and M. Milenkovic (eds.), *A Guided Tour of Multimedia Systems and Applications*. IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 95-107.)

Lee, J. C.-M., Li, Q. and Xiong, W. (1997). VIMS: a video information management system. *Multimedia Tools and Applications*, Vol. 4, No. 1, January, pp. 7-28.

Liou, M. (1991). Overview of the p×64 kbit/s video coding standard. *Communications of the* ACM, Vol. 34, No. 4, April, pp. 59-63.

Little, T. D. C. (1994). Time-based media representation and delivery. In: J. F. K. Buford (ed.), *Multimedia Systems*. ACM Press / Addison-Wesley, New York, NY, pp. 175-200.

Little, T. D. C., Ahanger, G., Chen, H.-J., Folz, R. J., Gibbon, J. F., Krishnamurthy, A., Lumba, P., Ramanathan, M. and Venkatesh, D. (1995). Selection and dissemination of digital video via the Virtual Video Browser. *Multimedia Tools and Applications*, Vol. 1, No. 2, June, pp. 149-172.

Little, T. D. C., Ahanger, G., Folz, R. J., Gibbon, J. F., Reeve, F. W., Shelleng, D. H. and Venkatesh, D. (1993). A digital video-on-demand service supporting content-based queries. In: *Proceedings of the First ACM International Conference on Multimedia*, Anaheim, CA, August. ACM Press, New York, NY, pp. 427-436.

Little, T. D. C. and Ghafoor, A. (1990). Synchronization and storage models for multimedia objects. *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, April, pp. 413-427.

Little, T. D. C. and Ghafoor, A. (1991). Spatio-temporal composition of distributed multimedia objects for value-added networks. *Computer*, Vol. 24, No. 10, October, pp. 42-50.

Little, T. D. C. and Ghafoor, A. (1993). Interval-based conceptual models for time-dependent multimedia data. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, August, pp. 551-563. (Reprinted in: B. Furht and M. Milenkovic (eds.), *A Guided Tour of Multimedia Systems and Applications*. IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 257-269.)

Lowe, D. (1995). HyVIS: The Hypermedia and Visual Information Systems Group. *IEEE MultiMedia*, Vol. 2, No. 3, Fall, pp. 70-72.

Mackay, W. E. (1989). EVA: an experimental video annotator for symbolic analysis of video data. *SIGCHI Bulletin*, Vol. 21, No. 1, October, pp. 68-71.

Mackay, W. E. and Davenport, G. (1989). Virtual video editing in interactive multimedia applications. *Communications of the ACM*, Vol. 32, No. 7, July, pp. 802-810.

Manaris, B. Z. and Slator, B. M. (1996). Guest editors' introduction: Interactive natural language processing: building on success. *Computer*, Vol. 29, No. 7, July, pp. 28-32.

Martin, P., Crabbe, F., Adams, S., Baatz, E. and Yankelovich, N. (1996). SpeechActs: a spoken-language framework. *Computer*, Vol. 29, No. 7, July, pp. 33-40.

Meyer-Boudnik, T. and Effelsberg, W. (1995). MHEG explained. *IEEE MultiMedia*, Vol. 2, No. 1, Spring, pp. 26-38.

Moore, J. D. and Mittal, V. O. (1996). Dynamically generated follow-up questions. *Computer*, Vol. 29, No. 7, July, pp. 75-86.

Mundy, J. L. (1995). The Image Understanding Environment program. *IEEE Expert*, Vol. 10, No. 6, December, pp. 64-73.

Narasimhalu, A. D. and Christodoulakis, S. (1991). Multimedia information systems: the unfolding of a reality. *Computer*, Vol. 24, No. 10, October, pp. 6-8.

O'Docherty, M. H. and Daskalakis, C. N. (1991). Multimedia information systems – the management and semantic retrieval of all electronic data types. *The Computer Journal*, Vol. 34, No. 3, pp. 225-238.

Oomoto, E. and Tanaka, K. (1993). OVID: design and implementation of a video-object database system. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, August, pp. 629-643. (Reprinted in: B. Furht and M. Milenkovic (eds.), *A Guided Tour of Multimedia Systems and Applications*. IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 323-337.)

Paris, C. and Linden, K. V. (1996). An interactive support tool for writing multilingual manuals. *Computer*, Vol. 29, No. 7, July, pp. 49-56.

Parkes, A. P. (1988). CLORIS: a prototype video-based intelligent computer-assisted instruction system. In: *Proceedings of* RIAO '88, pp. 24-50.

Pentland, A., Picard, R., Davenport, G. and Haase, K. (1994). Video and image semantics: advanced tools for telecommunications. *IEEE MultiMedia*, Vol. 1, No. 2, Summer, pp. 73-75.

Price, R. J. (1991). Multimedia information systems. In: J. A. Waterworth (ed.), *Multimedia: Technology and Applications*. Ellis Horwood, New York, NY, pp. 114-150.

Rader, G. M. (1996). Creating printed music automatically. *Computer*, Vol. 29, No. 6, June, pp. 61-68.

Rich, C., Waters, R. C., Schabes, Y., Freeman, W. T., Torrance, M. C., Golding, A. R. and Roth, M. (1994). An animated on-line community with artificial agents. *IEEE MultiMedia*, Vol. 1, No. 4, Winter, pp. 32-42.

Riley, M. D. (1989). *Speech Time-Frequency Representation*. Kluwer Academic Publishers, Boston, MA.

Rodriguez, A. A. and Rowe, L. A. (1995). Multimedia systems and applications. *Computer*, Vol. 28, No. 5, May, pp. 20-22.

Sakauchi, M. (1994). Database vision and image retrieval. *IEEE MultiMedia*, Vol. 1, No. 1, Spring, pp. 79-81.

Sheremetyeva, S. and Nirenburg, S. (1996). Knowledge elicitation for authoring patent claims. *Computer*, Vol. 29, No. 7, July, pp. 57-63.

Siemer, J. and Angelides, M. C. (1996). Remedial tutoring with intelligent tutoring systems - the case of INTUITION. *Journal of Intelligent Systems*, Vol. 6, No. 3/4, pp. 279-308.

Siemer, J. and Angelides, M. C. (1997a). A comprehensive method for the evaluation of full-scale intelligent tutoring systems. *Decision Support Systems*, forthcoming.

Siemer, J. and Angelides, M. C. (1997b). Integrating an intelligent tutoring facility into a gaming-simulation environment. *Journal of Information Technology*, Vol. 12, No. 4.

Siemer, J. and Angelides, M. C. (1997c). Towards an intelligent tutoring system architecture that supports remedial tutoring. *Artificial Intelligence Review*, forthcoming.

Smoliar, S. W. and Zhang, H. J. (1994). Content-based video indexing and retrieval. *IEEE MultiMedia*, Vol. 1, No. 2, Summer, pp. 62-72.

Steinmetz, R. and Nahrstedt, K. (1995). *Multimedia: Computing, Communications and Applications*. Prentice Hall PTR, Upper Saddle River, NJ.

Strawn, J. (1994). Digital audio representation and processing. In: J. F. K. Buford (ed.), *Multimedia Systems*. ACM Press / Addison-Wesley, New York, NY, pp. 65-107.

Swanberg, D., Shu, C.-F. and Jain, R. (1992). Architecture of a multimedia information system for content-based retrieval. In: *Proceedings of the Third International Workshop on Network and Operating Systems Support for Digital Audio and Video*. Springer-Verlag, Berlin, pp. 387-392.

Tanaka, Y. (1996). IntelligentPad as meme media and its application to multimedia databases. *Information and Software Technology*, Vol. 38, No. 3, March, pp. 201-211.

Tonomura, Y., Akutsu, A., Taniguchi, Y. and Suzuki, G. (1994). Structured video computing. *IEEE MultiMedia*, Vol. 1, No. 3, Fall, pp. 34-43.

Triebwasser, M. A. (1994). Multimedia: digital video in the social sciences - a brief tutorial on an emerging technology. *Social Science Computer Review*, Vol. 12, No. 4, Winter, pp. 543-559.

Vicsi, K. (1995). A product-oriented teaching and training system for speech handicapped children. *Journal of Microcomputer Applications*, Vol. 18, No. 4, October, pp. 287-297.

Wactlar, H. D., Kanade, T., Smith, M. A. and Stevens, S. M. (1996). Intelligent access to digital video: Informedia project. *Computer*, Vol. 29, No. 5, May, pp. 46-52.

Waibel, A. (1988). *Prosody and Speech Recognition*. Pitman, London.

Waibel, A. (1996). Interactive translation of conversational speech. *Computer*, Vol. 29, No. 7, July, pp. 41-48.

Weiss, R., Duda, A. and Gifford, D. K. (1995). Composition and search with a video algebra. *IEEE MultiMedia*, Vol. 2, No. 1, Spring, pp. 12-25.

Wermter, S. and Weber, V. (1996). Interactive spoken-language processing in a hybrid connectionist system. *Computer*, Vol. 29, No. 7, July, pp. 65-74.

Woolf, B. P. (1996). Intelligent multimedia tutoring systems. *Communications of the* ACM, Vol. 39, No. 4, April, pp. 30-31.

Woolf, B. P. and Hall, W. (1995). Multimedia pedagogues: interactive systems for teaching and learning. *Computer*, Vol. 28, No. 5, May, pp. 74-80.

Wu, J. K. and Narasimhalu, A. D. (1994). Identifying faces using multiple retrievals. *IEEE MultiMedia*, Vol. 1, No. 2, Summer, pp. 27-38.

Yoshitaka, A., Kishida, S., Hirakawa, M. and Ichikawa, T. (1994). Knowledge-assisted content-based retrieval for multimedia databases. *IEEE MultiMedia*, Vol. 1, No. 4, Winter, pp. 12-21.

Zhang, H. J., Kankanhalli, A. and Smoliar, S. W. (1993). Automatic partitioning of full-motion video. *Multimedia Systems*, Vol. 1, No. 1, June, pp. 10-28. (Reprinted in: B. Furht and M. Milenkovic (eds.), *A Guided Tour of Multimedia Systems and Applications*. IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 338-356.)