23 SEMANTICS OF A MULTIMEDIA DATABASE FOR SUPPORT WITHIN SYNTHETIC ENVIRONMENTS FOR MULTIPLE SENSOR SYSTEMS

Gerald Sterling
Defence Science and Technology
Organisation

Elizabeth Chang, and
Tharam Dillon
LaTrobe University, Melbourne, Australia

Abstract: This paper describes work investigating the application of synthetic environment technology to sensor systems with particular emphasis on the multimedia aspects of the work. The work is being done within the Air Operations Division of the Defense Science and Technology Organisation and La Trobe University in Australia.

An analysis of the needs of the system users is carried out and a 3D synthetic environment design developed which aims to improve overall man-machine performance of the surveillance sensor systems. It is considered that the improved performance will arise from improved operator situation management system which enables the user to 'see' what his sensors detect and which assists the user in the analysis and interpretation of raw sensor data and higher level information produced by sensor system post-processing. Importantly, the interface permits a user to revisit multimedia archives that record past activity within the sensed world.

An overview of the conceptual analysis and design is presented. A conceptual model of the synthetic environment has been developed covering the knowledge base, the user interface and the sensor system interfaces. An object-oriented model of a synthetic knowledge base is used to record the sensed world and present it to the user as an analogue of the real world. The analysis of requirements of the multimedia facilities that are presented to the operator are discussed in some detail. A method of

assigning meaning to segments of continuous data similar to video is developed. A suitable data model to accommodate multimedia data is proposed. Finally a method of developing user interfaces for presentation of multimedia data based on the concepts of the Abstract User Interface Objects is presented.

23.1 INTRODUCTION

Synthetic Environment Systems cover a broad range of factors from displays systems, auditory and visual, through tactile interface and user interaction issues to improve generation, computing platform and other human factors aspects. This investigation considers some of these issues in the context of the concept of a 3D visual and auditory synthetic environment to portray sensor system data. Mentor is intended to improve the workload disposition of sensor system operators in the formation of some understanding of their tactical situation. To this end Mentor should help the operator assess his situation by aiding in the monitoring of the behavior of each sensor contact.

Mentor has an overall configuration as shown in Figure 23.1. In past work the structural aspects of the objects within the Mentor concept have been described (Sterling and Dillon) [1]. Important amongst these is the hierarchical nature of the Mentor model of the Real World Environment (RWE) that is constructed within the Mentor knowledge base. In this model the Synthetic World Environment (SWE) sensor contacts are represented as synthetic world entities (SWEs). Knowledge regarding contacts – real world entities (RWEs) – and knowledge required to reason about the RWEs is stored in this object-oriented representation of the world.

In this paper we continue to discuss the conceptual structural and behavioral analysis of Mentor. In particular there is a focus upon those aspects of the information management that relate to multimedia issues within Mentor. Multimedia in the Mentor context deals with raw video from radar and flir sensor systems, audio recording of explanatory reasoning from operator or agents and textual record of explanatory reasoning of operator or agents as well as numerical information that indicates location, speed and duration of motion. This information is stored within a Multimedia database to provide the Mentor user the capacity to look again at the way a particular scenario developed and perhaps reinterpret the situation to take account of some previously unobserved facet of an entity's behavior.

The Multimedia information stored in the database is, therefore, primarily focused on storing information collected in real time from the sensors. It also requires putting markers on this information with respect to time, events and locations. There is also a need to retrieve information from the database not just using textual or symbolic ways but using queries that utilize image and

video based information in the query. To this extent it has a somewhat different purpose to multimedia databases that might be used in conjunction with authoring systems or non time oriented information. This requires a somewhat different structure and additional semantics for the multimedia database. A key issue here is the development of the user interface to the multimedia database to assist with appropriate display of multimedia information.

In general, this display has a stronger emphasis on the dynamic aspect as well as the structural aspects normally associated with other databases. In order to facilitate an appropriate design, we introduce and define the notion of Abstract User Interface Objects which define the “external” schema to the multimedia database.

The paper will first briefly discuss the Operator’s Domain and the general behavior of Mentor and its relationship to the operator. Second, the discussion revisits the conceptual model of Mentor addressing primarily the structural aspects of the knowledge base embodied within Mentor and Mentor itself. Third, some attention is paid to the nature of the SWE object representation of an entity including its attributes, methods and rule sets. Next the discussion focuses on the requirements and basis for multimedia management of video, audio and textural records of contact behavior. It examines the semantics underlying the multimedia database and defines Abstract User Interface Object which are used to define the “external schema” to the multimedia database.

Figure 23.1: Configuration of a Sensor Platform and Mentor

23.2 BRIEF OVERVIEW OF THE MENTOR SYSTEM

We present a brief overview of Mentor in this section to create a context for the multimedia system being developed.

23.2.1 *The Mentor Concept*

As sensor systems become more capable many more contacts require management. Each contact requires investigation and each requires some decision making on the part of the operator/interpreter. This situation loads even more work onto already heavily loaded operators. Garner and Assenmacher [2] referred to these matters in their discussion on improving situation awareness. "Operator SA comprises detecting information in the environment, processing the information with relevant knowledge to create a mental picture of the current situation and acting on this picture to make a decision or explore further."

Mentor (see Figure 23.1) can be considered as an interface to a suite of sensors. When a sensor makes a contact, a representation is formed of an unknown entity within the existing tactical terrain environment. The spatial arrangement between the contact, the host sensor platform, the terrain, and other entities is maintained. The operator interacts with entities using this 3D perspective display. The heart of Mentor is the intelligent assistant system used in the management of these entities.

Conceptually this management system is an Intelligent Assistant System that will aid the operator to monitor the behavior of entities. Boy [3] introduces notions of Intelligent Assistants that learn in a manner similar to Rasmussen's [4] Skills Rules Knowledge model of Human Learning. Mentor will employ an object-oriented rule based approach in which intelligent assistant agents respond to rule sets devised by the operator as he deals with entity types for the first time. As these rule sets demonstrate their utility they may be retained for later application with other Synthetic World Entities (SWE's) of that class.

The long term objectives of Mentor is to investigate the use of intelligent assistant systems and multimedia database support systems and assess the utility of 3D perspective visual and auditory interfaces for Sensor management.

23.2.2 *Basic Structure of Mentor*

The fundamental objective of Mentor is to improve a sensor system operators understanding of what is occurring in the sensed world and ease

communication of these understandings to others. To this end analysis has assumed that any final product will be distributed in nature and be of an open architecture that will readily support expansion with the addition of additional nodes to the network. However other than this underlying philosophically based assumption the analysis is not predicated on any other design consideration. The system under investigation has been described in conceptual terms elsewhere (Sterling and Dillon [¹]) but some further attention will be paid to the issues here in particular those aspects that provide the foundation to multimedia database management objects.

The particular focus of this work has been on defence type surveillance systems. These may be airborne, seaborne or land based but for discussion purposes consider say an Airborne Early Warning & Control aircraft (AEW&C). It has many sensor systems, several with sophisticated post processing capability supporting data reduction (range bearing altitude elevation velocity) and tracking. The envisaged Mentor system will have a network interface to sensor, (say Mil Std 1553 bus interface) and it has the capacity to capture lower level raw video or audio output of such systems.

Real World Entities (RWE's) are the things that are detected by sensors and are the substance behind visual and auditory models that appear in the synthetic environment. A full sensor system has several components and these are Video, FLIR, Radar, Sonar and ESM (Electronic Support Measures). The output of all of these is continuous in the sense that video is continuous. This creates a more complex multimedia environment since there are many continuous media. Mentor's role is to provide the interface between the operator on the platform and the sensor systems.

Mentor consists of a number of component subsystems, the most important of these is the Contact Knowledge Base. The Contact Knowledge Base (CKB) is the repository of all information regarding sensed entities (contacts) and each such contact is instantiated in the knowledge base as a Synthetic World Entity (SWE) – (Figure 23.2). Each SWE also acts as an agent, which using rules sets associated with each SWE reasons about the contact and informs the operator as required. Figure 23.2(b) shows one of the component objects namely Video embodiment. The other component objects excluding commentary audio have an analogous structure.

Figure 23.2(a): Synthetic World Entity

Figure 23.2(b): Video Embodiment

The structure of the CKB is essentially hierarchical and the operator forms it as he reasons and forms inferences about a contact. An observation might first be inserted as a contact, next as an aircraft contact, next as a rotorcraft, next a large rotorcraft and concludes perhaps finally as a type. However a particular operator might choose a much flatter structure and Mentor should support both approaches. At each level the context is different and for each context the operator specifies appropriate rules sets which an individual Mentor agent uses to monitor its SWE. These rules provide the action strategy to be used by each agent in its liaison with the operator, other agents, hardware systems and sensors.

An important facility provided to the operator is the capacity to replay old sensor information, that is, to revisit data to confirm that the prevailing interpretation is correct. Clearly higher level data can be stored within the knowledge base but lower level video and audio data from sensors is not so

readily revisited. Also as agents reason about contacts explanatory information is produced both in auditory and textual form and the operator may himself form auditory comments that may be recorded. This is multimedia data with a strong temporal context and Mentor supports the review of such media.

Whilst the Synthetic environment itself can be replayed by looking at the temporal history of each contact and replaying it, replaying the multimedia components seems to require different handling. The requirements are analysed this way for four reasons. First, much of the multimedia data is continuous in nature. Second, it has to be viewed in the same temporal context as other media. Third, accessing the data by content is difficult and finally because of the nature of the likely multimedia archival mechanisms can be compute intensive.

With the facilities described the investigation can probe more deeply into the problems and mechanisms necessary to deliver effective sensor control and improved situation awareness using 3D perspective visual and auditory displays.

23.3 MULTIMEDIA MANAGEMENT METHODS FOR DEALING WITH MEDIA RETRIEVAL

Mentor receives sensed data in higher level form across a data bus and in lower level form as direct video or audio. It is important that the auditory comments of the operator be archived where such may be of benefit in the interpretation of a tactical situation. Further textual data reflecting intelligent assistant explanatory output should be accessible. The issue is how best to deal with this information, how best to tie it back to the synthetic world that is primary user interface to the sensor systems, and how to maintain continuity between the two.

Previous work on video management in multimedia framework was reviewed to help decide on an approach. Lieinhart, Effelsberg, Jain [5] in "Visual GREP" propose a systematic method to compare and retrieve video sequences which strives to determine similarity in video sequences. They describe techniques to compare video at the frame, shot, scene and whole video levels in an attempt to develop a query system for video files analogous to the UNIX (grep) for text files. They describe a colour coherence vector (CCV) to quantify color atmosphere, an edge change ratio (ECR) to measure motion intensity, face detectors and static and mobile framing to make similarity judgements and then use an algorithmic approach to comparing video at differing video levels. It does not seem to provide techniques that would serve especially well as an index mechanism to allow an operator a rapid search technique to review a lengthy recording that is effective for a real time system.

Wactlar, Kanade, Smith, and Stevens [6] work involves accurate connected speech recognition to automatically transcribe video sound track for later analysis. They assert the importance of image and language understanding in finding a video paragraph using camera cuts, object tracking, speaker changes and changes in audio content. They identified numerous sources of error in video transcription that must be addressed. They asserted no pressing need for real time performance. They discuss the image and language processing issues and describe the effectiveness of their combined image and language skimming techniques. However, whilst some of the image processing techniques may have some applicability in the context of our work (object presence and scene transition effects) the Imformedia Project work as a whole is not of great relevance.

Leinhart, Pfeiffer and Effelsberg [7] in "Video Abstracting" describe their work in the formation of video abstracts. They analyze a movie as a progressive aggregation of frames into shots, of shots into scenes and of scenes into the whole video. They define a "clip" as "a frame sequence selected to become an element of the abstract". Their three-step process includes video segmentation and analysis, clip selection and clip assembly. In the segmentation process shots are determined by video and audio similarity (frequency and intensity spectrum) and dialog detection between consecutive shots. After segmentation the important features are extracted: e.g. actors, gunfire explosions and text. The Rowley, Buluja, Kanade face detection is used for identifying actors, audio analysis for action and OCR type text recognition. The clips are selected then on the basis of title text, action and genre. Comparison of automatically generated abstracts with man made ones did not detect a "better" product one way or the other. That being said for the purposes of our work, the interest must lie in the early segmentation work with cut detection and the use of Edge Change Ratio for that purpose. The notion using frames and clips as abstractions, is in many respects similar to our approach.

Christel, Smith, Taylor, and Winkler [8] consider that approaches of forming brief titles with individual "thumbnail" images (similar to our Stills) does not capture the temporal nature of video. Their proposal for a "video skim" presents a short video compacted by the ratio 10:1 and viewed at normal playing speed. The skim preserves important audio and video component.

Our work aims to record raw material in multiple media formats and in essence the only techniques that seem applicable are either some form of visual fast forward, use of some form of cut detection and formation of a "Stills" collection or the formation of a "Rushes" collection. The other aspect of our work is the development of an appropriate user interface to support the fast review technique selected. The work at University of Mannheim and at

Carnegie Mellon University has broader applicability to video archival and retrieval particularly in a non real time framework whilst our interest is more on cut detection and image segmentation. In our technique, we abstract the meaning of the video into either stills, video rushes, temporal or textual markers.

First some preliminary discussion of the synthetic environment. As a scenario is played out a record is kept for each contact of position versus time. A trail of 'flagstones' can be switched on which trails the contact (SWE) showing where it and other entities have been in time and it also shows important event markers. By selecting a suitable flagstone or marker the synthetic environment may be regenerated and provides access to all contact data available at that point. This would include time and location and accordingly these data should be useful as an index to the multimedia recordings. Thus it should be possible to initiate replay by selection of such markers.

In recording media information the global philosophy has been to record material and maintain the context with SWE object in the process. In other words if a radar or flir maintains an ongoing focus on an entity then that video is recorded with reference to that entity. If a sensor is maintaining a general 360° sweep then a media client at the Contact Knowledge Base level manages that recording accordingly, then an operator may expect that by selecting a SWE his multimedia replay and record accesses will all pertain to that entity.

In this analysis consideration has been given to the work of Tonomura, Akutsu, Taniguchi and Suzuki [9] paper regarding structure and meaning. The approach they took in respect of browsing is of particular value with the notions they introduce of flash and rush browsers and paper video. Within Mentor after a video segment has been recorded the sequence is reviewed and searched for logical cuts and event markers. A small portion of the sequence after the cut or the marker is appended to a 'rush' file and a single frame appended to a 'still' file. The operator may then review either the 'rush' or 'still' file and commence detailed replay from that point. Checking the record time reference of the frame of interest and then replaying all media using index-time effects this. Where a media has no recorded information at the initial index-time, replay commences when data becomes available as the index-time increments during replay.

Whilst an entity is in the synthetic world there will be periods where a media record is either not necessary or not available and such breaks lead to logical formations of sequences of numbered sessions. Sessions have a start time, a duration and a start location and provide a ready access to the data in the companion media.

Video, audio and text (Media) may be received continuously or intermittently. The material will depend on the use to which the sensor is being put (it may be temporarily focused on other contacts), what the operator

or agent may wish to say and what explanatory text is in fact necessary. This intermittency necessitates the provision of logical session and composite sequences.

Ultimately an operator or an agent controls a media client and recording and checking the field of view of the sensor for consistency with the bearing and elevation of the SWE and the current range of the sensor can control its termination. A facility is required such that of an SWE object is the subject of video or other media, a message is sent from the SWE object and temporary recording is started. Each recorded frame is numbered, a time recorded, a location and any event marker number and name noted.

Each reference is then be available as an index to a desired frame, audio segment or textual passage. An operator or agent may request that recording be made permanent starting at a particular time and recording continues until a stop is called or the contact moves out of focus.

Figure 23.3 depicts the logical media data format that the media units are to create that will allow the browsing of data and the replay of material from a browser selected position or from a location defined by 'flagstones' from within the synthetic environment.

Figure 23.3: Logical Media Data Assembly

A still synopsis is to be formed by recording the location on disk of periodic frames (say every 500th) interspersed with frames collected after a cut and frames that coincide with an operator or agent initiated event marker. A rush file synopsis is formed by selecting a series of say 5-second sequences of

frames recording the location on disk of the start frames. Every 500th second say another 5-second interval is indexed. Interspersed within these are further 5 second frame sequences identified after a cut and 5 second frame sequences that coincide with an operator or agent initiated event marker. The formation of 'Stills' and 'Rushes' collections is less vital than real time recording and replay and is a background task that provides an important tool for the operator to find a required sequence in the archive.

23.4 MULTIMEDIA DATA MODEL

The multimedia database system is modeled using an Object-oriented Data Model, which is extended to allow for the different types of media.

In object-oriented systems, each instance object that is the value of an attribute belongs to an associated class. The associated class has a system-defined identifier, a value domain and allowed operations.

In the case of atomic instance objects, the value domain is a value set, and in the case of complex objects it is the domain construction, such as tuple type, set type, etc. If an instance object that is the value of an attribute is considered to be an element and the associated class an element type, then the value construction for instance objects and the value domain for the classes are described using a BNF-like language below.

Furthermore, the values could be complex objects. Note this forms a nested structure as the attributes associated with a class could themselves be complex objects and so on. In addition to the name and value of an attribute, it is also useful to associate semantic integrity constraints with it. These could specify the domain of the attribute (type), the range of permissible values, and admissibility of certain values.

<element>	::=	<ATOMIC element>
		<COMPOSITE element>
<ATOMIC element>	::=	<INTEGER element> <REAL element> <BYTESTRING element> <Image Element> <Audio Segment element> <Radar Segment element> <Flir Segment element> <Video Segment element> <Sonar Segment element>
<COMPOSITE element>	::=	<TUPLE element> <SET element>
<TUPLE element>	::=	[a ₁ :E ₁ , ..., a _n :E _n] a _j : attribute name E _j : element
<SET element>	::=	{E ₁ , ..., E _n }

Figure 23.4: Value construction

<element type>	::=	<ATOMIC element type> <COMPOSITE element type>
<ATOMIC element type>	::=	INTEGER REAL BYTESTRING IMAGE AUDIO SEGMENT VIDEO SEGMENT SONAR SEGMENT Radar Segment Flir segment
<COMPOSITE element type>	::=	<TUPLE element type> <SET element type>
<TUPLE element type>	::=	TUPLE of (a ₁ :EType ₁ , ..., a _n :EType _n) a _i : attribute name EType _i : element type
<SET element type>	::=	SET of (EType ₁ , ..., EType _n)

Figure 23.5: Value domain construction

As for other object-oriented databases, an attribute in a domain object is a structure field in a tuple that gives semantics to the tuple object. It serves as a field of a tuple object that maps to an atomic object (an object which has atomic element as a value) or a set of atomic objects. Note this value which is an atomic element could belong to any of the media categories as shown in Figures 23.4 and 23.5.

As for other object-oriented databases, a relationship, like an attribute, is a field of a tuple object. Unlike an attribute which links a tuple object to an atomic object, a relationship maps a tuple object to another tuple object or a set containing at least one tuple object. That is, a relationship eventually gives an atomic object after more than one level of reference. Note here that the relationship always maps to another tuple object at the first level of mapping. It is only after this that it would map to an atomic element, drawn from different media.

As with other object-oriented databases it is important to allow hierarchical relationships and non-hierarchical relationships.

A hierarchical relationship is a relationship that links two parties in a hierarchical fashion. There are three kinds of hierarchical relationships, normally associated with object-oriented systems and these are also pertinent for multimedia databases and they are: generalisation (ISA relationship),

classification (instance-of relationship) and aggregation (components-of relationship).

A non-hierarchical relationship is a relationship that links two parties in an arbitrary manner. This relationship cannot be represented in a hierarchy. These are sometimes referred to as association relationships.

The operations specified for a class and inherited by its instances are distinguished in three ways (Dittrich [10]/Dillon and Tan [11]): type specific vs generic; predefined vs user-defined; and one-level vs multi-level (composite elements). Type-specific operations are operations that are applied to the atomic values, whereas generic operations are for the composite values. Predefined operations are installed by the database developer. They are applied to some atomic value types. Generic operations are also usually predefined operations. User-defined operations, as the name suggests, are defined by the user. The existing operations can be called within the user-defined operations.

For composite elements, such as tuples and sets, operations can affect either the value only (one-level) or all levels of the whole composite element (multi-level).

Note in the case of video, to provide a handle for retrieval we often define a <labelled_video segment> as Tuple of (label: EType₁, duration:Etype₂ video_segment: VIDEOSEGMENT).

Note that the label, is of type text, image, video segment (in the case of a rush), time stamp. Etype₂ would be of type time duration. In a similar fashion, one could also define <labelled_audiosegment>. Note these atomic elements, labelled elements, and composite elements would be used as instance objects which form the value of attributes of domain objects.

Note that if we use a more general label to identify a video segment such as an Event label, or a label of type text, image or video segment (in the case of rush), then these would map directly to a time marker which would be used for retrieval.

With the structuring of video explained in the last section, a video segment, a radar segment or flir segment is essentially retrieved using a time marker. Event markers are mapped onto a time marker which is then used to retrieve the video segment, etc. To provide a handle for retrieval, we define a <time_labelled:video_segment> as

Tuple of (time

label:Etype₁,duration:Etype₂,video_segment:VIDEOSEGMENT)

Note that this time label and duration would be of type INTEGER.

23.5 ABSTRACT INTERFACE OBJECTS AS THE BASIS FOR ACCESSING MULTIMEDIA DATABASES

The user interface to a multimedia database is considerably more complex than that for a traditional database. The mechanisms used to define a user interface for a traditional database may not be adequate.

In databases, views are often used as the basis of generating particular displays for the user. A view in an object-oriented system can be thought of as a virtual class, ie. it is a class that does not contain any instances of its own. Rather, when it is invoked, a set of instances pertinent to it are generated. Since some views can be considered as specialisations of existing domain classes, the question that arises is whether these views should be incorporated into the domain ISA inheritance hierarchies. Early work on views by Scholl et al [12] and Abiteboul and Bonner [13] argued that they should be incorporated into the ISA or inheritance hierarchy of the domain. However, others such as Kim and Kelly [14] and Bertino [15] have argued that this is an inappropriate way to proceed.

It is inappropriate to include the view virtual classes in the domain inheritance hierarchies because:

- (a) a view can be derived from an existing class by having fewer attributes and more methods. It would be inappropriate to treat it as a subclass unless one allowed for the notion of selective inheritance of attributes;
- (b) two views could be derived from the same subclass with different groups of instances. However the instances from one view definition could be overlapping with the other and non-disjoint. An example of this would be a view A of employees that are salaried and casual employees. Consider another view B which separates employees into four different areas of Melbourne such as the Northern, Eastern, South Eastern and Western Suburbs. It is clear that both view A and view B partition the instances of class employee. However, note that one employee living in the Northern Suburbs could also be a Salaried employee. Hence, these view definitions A and B do not disjointedly partition the instances of the class employee.
- (c) view definitions, while useful for examining data, might give rise to classes that may not be semantically meaningful to users Bertino [15].
- (d) effects of schema changes on classes are automatically propagated to all subclasses. If a view is considered as a subclass, this could create problems Kim and Kelly [14] in requiring the changes to be propagated to the view as it might be appropriate or inappropriate.
- (e) an inappropriate placement of the view in the inheritance hierarchy, could lead to violation of the semantics because of the extent of overlapping with an existing class Kim and Kelly[14].

For the above reasons, it is better that views be part of a separate inheritance hierarchy known as a view inheritance hierarchy. This view inheritance hierarchy is useful if we wish to define a view as a specialisation of another view. For example, if we wish to define the view `Overtime_Permitted_Salaried_Employee` as a specialisation of the view `Salaried_employee`. This new view is considered to be a subclass of the virtual class (view) `Salaried_Employee`.

Bertino [15] has essentially defined a view definition language. It is claimed that this view definition language allows one to represent both structure and behavioral features. However, it does the second only to a limited extent in the sense that it specifies the methods in the definition of a view but it does not model inter-object dynamics, such as the sequencing of messages between objects which define their dynamic interactions with one another. This feature, it must be conceded, is not of paramount importance in traditional database systems even though one still has to take cognisance of pre-conditions and post-conditions before a method in a particular object can be invoked. Kim and Kelly [14] also gave a very careful definition of views. Barclay and Kennedy [16] define three types of views: selection, projection and join. The above comments about deficiencies in modelling inter-object dynamics also apply to the work of Kim and Kelly [14] and Barclay and Kennedy [16]. While the modelling of inter-object dynamics does not loom so large in databases, it is of considerable significance when modelling multimedia systems. It is with text or numeric values of vital importance in real time systems, event driven software, simulation software and multimedia systems or in defining the navigational aspects of user interface design. In this section, we will introduce the notion of Abstract user Interface Objects, to overcome some of these deficiencies.

The primary purpose of views is three fold (Bertino [15], Kim and Kelly [14]).

- (1) to allow the user to directly interact with data from a particular orientation and hence making them useful in defining queries.
- (2) for content-based authorisation schemes
- (3) as a basis for schema evolution

The Abstract User Interface Objects will also permit this and in addition allow the user to deal with the navigational aspects of the user interface.

Postulate 1 An Abstract User Interface Class (AUI) is a virtual class and it has

- (i) attributes
- (ii) methods

Since it is a virtual class, there are no stored instances; they are generated only when it is invoked. We refer to these as generated instances

Postulate 2 An Abstract User Interface Class (AUI) is specified by its class specification that is a signature which consists of attributes, methods and their domains.

Postulate 3 The Abstract UI objects can generally be distinguished into two broad categories:

- (i) AUI Information Objects
- (ii) AUI Command Objects

Postulate 4 AUI Information Objects provide a display of information for the user, to explain and expose the system state to the user, provide help media between the user and the system. They are essentially a document which describes the knowledge of the system state to the user and is a service which the user can use for further decision making. They are more static than dynamic, because they do not have “pre-defined user operations or functions” associated with these AUI objects, which means they do not need the user to operate on them to get further processing

(Here, do not confuse predefined user operations with pre-defined object functions). All objects have their own properties such as attributes and methods. For example, ‘Are you a student?’ is an AUI object, which potentially will be used on screen; the ‘display’ of this message is a method built in with Message objects. But this is done by the system, not by the user.

Postulate 5 AUI Command Objects provide the following facilities:

- (i) navigation from one window or portion of the UI to another window or another portion of UI;
- (ii) moving data (ie: moving data from screen to the application program or database or vice versa);
- (iii) initiating an action within the system or stopping the action being carried out by the system.

They have predefined user events or methods or functions associated with them. They need user interaction to get further processing for completing tasks. They are more dynamic in the sense they require the user to take an action during a task performance. When the user takes an action, it is seen by the system as an Event. Hence we call them Command or Event or Control Objects.

Postulate 6: An Abstract User Interface class (AUI) is a selection or a projection or contains other user-defined operations on a single domain class.

A projection consists of only some of the properties of the base domain class.

For a AUI class C’ that is a projection of the base domain class C.

If $p=[p_1, p_2, \dots, p_n]$ is a list of properties of the base domain class C , and

$p'=[p'_1, p'_2, \dots, p'_3]$ is a list of properties of C' , then

$p' = p'' \cup p$ where

$p'' \subseteq p'''$ and

p''' are the list of properties that consist of the user defined operations that are defined in the AUI class. Further p'' , p' and p are properties that include attributes, relationships and user defined operations.

If AUI class C' is a selection on a base domain class C , which satisfies some specified condition then $C' \subseteq C$. The generated instances of C' are a subset of the instances of C .

Postulate 7: An Abstract User Interface class (AUI) is a selection or a projection or other user-defined operation on two or more classes.

Postulate 8: All Abstract User Interface classes (AUI) are organised into an ISA hierarchy built by the binary ISA relationship which has a particular order. The ISA hierarchy for AUI classes is separate from the ISA hierarchy for domain classes or persistent classes (in a database). This permits an AUI class to be a specialisation of another AUI class. It also permits one to define an AUI class based on another AUI class. The reasons for separating the ISA hierarchy of the AUI classes and the domain classes or persistent classes are similar to those discussed for views in the last section.

Postulate 9: An attribute of an AUI class may be identical to that of a domain class or it could have a different definition or domain. It may also be computed from one or more attributes of domain classes.

Postulate 10: The specification of a method in an AUI class may be the same as that of the corresponding base domain classes or is redefined in the case of a method with the same name as the base class, or is newly defined in the case of an additional method name. Thus, if O is an operation or method in the base domain class C and the AUI class C' is derived from C , then an operation O' of C' where $O' = O'' \cup o$, where ' o ' are the user defined methods that define the AUI class C' , can correspond to one of the three cases below:

(a) $O'' \subseteq O$

(b) $O'' \supseteq O$

(c) $O'' = O$

Postulate 11: An AUI class C' can be a composite virtual class consisting of component AUI classes (E' , F' , G') where C' is derived from the base composite domain classes C and E' , F' and G' are

derived from domain classes E, F and G respectively where C is a composite class consisting of component classes E, F and G.

Postulate 12: If AUI classes C_1' and C_2' are derived from base domain classes C_1 and C_2 respectively.

The object messages diagram has messages between one or more of the following:

- C_1' and C_1
- C_2' and C_2
- C_1' and C_2'
- C_1 and C_2 .

These Abstract Interface Objects are generalisations of the notion of views and are suitable for basing queries on Multimedia Database systems of the sort found in the Mentor system.

23.6 ILLUSTRATIVE EXAMPLE OF USE OF ABSTRACT USER INTERFACE OBJECTS

We carry out a sample design of a screen that is utilised with the multimedia database based on the notion of abstract user interface objects. For the synthetic world entity and the video embodiment given in Figure 23.2 we can derive the following abstract user interface objects shown in Figure 23.6 for use with the stills version of accessing video.

Note that these Abstract User Interface Objects have a subset of the attributes in the video embodiment object and some additional attributes and methods. A screen representation corresponding to these Abstract User Interface Objects is shown in Figure 23.7.

Figure 23.6: Sample Abstract User Interface Object

Figure 23.7 : Sample Window Layout

23.7 CONCLUSION

Mentor aims to create a 3D visual and auditory synthetic environment for sensor system operators. The synthetic environment will provide the operator with an interface to a knowledge base representing the contacts made in the real world environment and this knowledge base will be supported by an intermediate intelligent assistant system. The IAS in Mentor will be taught by the operator to provide him the kind of support that the particular operator values most. The operator will teach the IAS the rules sets and behaviour he finds most valuable. A data model is proposed which is the basis for video and audio segments that pertain to entities. In this paper we have described the data model and abstract interface objects that will support these multimedia facets of Mentor.

References

- [1] Sterling G, Dillon T; *Analysis and Design of Synthetic Environments for Multiple Sensor Systems*; Proc. of the Second International Simtect Conference; Canberra Australia
- [2] Garner K T; Assenmacher T J; *Improving Airborne Tactical Situational Awareness*; Journal of Electronic Defense; November 96 p42-45
- [3] G Boy; *Intelligent Assistant Systems*; Academic Press 1991
- [4] Jens Rasmussen; *Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human performance Models*; IEEE Transactions on Systems and Cybernetics vol.SMC-13, No.3 pp.257-266
- [5] Leinhart, R., Effelsberg, W., Jain, R., *Visual GREP: A systematic method to compare and retrieve video sequences*, SPIE, Vol. 3312, January, 1998
- [6] Wactlar, H., Kanade, T., Smith, M., Stevens, S., *Intelligent Access to Digital Video: The Informedia Project*, IEEE Computer, Vol. 29, No. 5, May, 1996
- [7] Leinhart, R., Pfeiffer, S., and Effelsbert, W., *Video Abstracting*, Communications of ACM, Vol 40, No. 12, pp55-62, December, 1997.
- [8] Christel, M.G., Smith, M.A, Taylor, C.R., & Winkler, D.B., *Evolving Video Skims into Useful Multimedia Abstraction*", Proceeds of the CHI'98 Conference on Human Factors in Computing Systems. (Los Angeles, CA., April, 1998)
- [9] Tonomura, Akihito Akutsu, Yukinobu Taniguchi, Gen Suzuki; *Structured Yoshinobu Video Computing*; IEEE Multimedia Fall 1994; p34-43

- [10] Dittrich, K.R. *Object-oriented Database Systems for information systems of the future*, Seminar Notes, Melbourne
- [11] Dillon and Tan *Object-Oriented Conceptual Modelling*, Prentice-Hall (1993)
- [12] Scholl, M., et al *Object Algebra and Views for Multi-Objectbases*, Proceedings of Workshop on Distributed Object Management. Edmonton, Alberta, Aug. 1992
- [13] Abiteboul, S. and Bonner, A. *Objects and Views*, Proceeds ACM SIGMOD Conference, pp 238-247 (1991)
- [14] Kim, Wong and Kelly, W., *On View Support in Object-Oriented database Systems*, Marden Database Systems: The Object Model, Interoperability, and Beyond, ACM Press pp 109-129 (1995)
- [15] Bertino, Elisa *A view Mechanism for Object-Oriented Databases*, Proceedings of the 3rd International Conference on Extending Database Technology (EDBT92), Vienna, Austria, March 1992 pp 136-151 (1992)