An Efficient Event Definition Framework for Retail Sector Surveillance Systems

Fahad Anwar[1,3], Ilias Petrounias[2,3], Sandra Sampaio[4,3]
WMIC[1], Manchester Business School[2]
The University of Manchester[3]
Manchester, UK
fahad.anwar@manchester.ac.uk
ilias.petrounias@manchester.ac.uk

Vassilis Kodogiannis[4,5], Tim Morris[4,3]
School of Computer Science[4]
University of Westminister[5]
London, UK
sandra.sampaio@manchester.ac.uk, kodogiv@wmic.ac.uk,
tim.morris@manchester.ac.uk

***Abstract*— Event representation models provide a framework in which we can reason about events so as to interpret the collective behaviour of objects over time and space domains. Many are context-specific and lack flexibility when faced with unstructured video. In the past many efforts have been made to define a comprehensive event description framework (EDF), which can provide a framework to develop ontologies for semantic annotation of video events. However, it is observed that there are some areas of event modelling that were not fully explored. Hence, we extended and modified the EDF and proposed the extended version of it (EDF^E). Following are some of the major extensions we have proposed in EDF^E. 1) EDF^E extends the entity representation model of EDF by introducing three new entity classes: that of text entity, virtual entity and internal entity. II) EDF^E introduces a new set of predicates for describing more complex event scenarios and facilitating the event detection process. It also introduces granularity as a feature of temporal predicates to capture the temporal association between sub-events. III) It introduces the event evidence feature to capture the full evidence for the detected events. IV) The data structure of EDF is extended and modified to capture the properties of EDF^E and to store the results of the event detection process. V) We model complex events from real world surveillance videos using the proposed EDF^E.**

Keywords-*Multimedia event modelling; intelligent surveillance system; multimedia event annotation and data mining.*

I. INTRODUCTION

Due to the flexibility and expressive power of rich semantic models, they provide a solid ground for any semantic video event model to be used for structured and unstructured multimedia content such as surveillance videos. In [1], Gupta et. al. presented the VIMSYS model, although it mainly focuses on image contents, it provides the basic platform on which many video content management models were later based on. HMM based models presented in [2, 3, 4] have been used for event modelling. However, in this approach representation of an event is not transparent and it is difficult to generalise for new events. The work presented in [5, 6, 7, 8] mainly confronted the problem of indexing video data for efficient data extraction through user queries. In [5], the authors provide a hierarchical structure of video contents by dividing them into video objects, activity and events. The work presented in [8] takes temporal aspects of multimedia content into account in database management of video contents. Although these approaches provide conceptual understanding of how to model multimedia data for event modelling, they do not focus on modelling events in advance for event detection in a real time environment. Moreover, they do not explore the features of video event modelling which can facilitate event detection and discovery of unknown interesting events.

In [9, 10, 11] an important aspect of multimedia data management (the uncertain nature of data and its queries) was addressed. In [12, 13] mapping functions were used to define the relationships among semantic objects and explain how scene layer, object layer and concept layer can be connected by utilising the temporal aspects of multimedia data. In [14, 15] trajectory-projection information along with spatio-temporal aspects were utilised to confront the complex video data retrieval requirements. Object oriented approaches discussed above provide the basis for a video event modelling framework for unstructured multimedia contents (surveillance videos). However, these approaches mainly fall in the video indexing and retrieval category. The work presented in this paper will use an object oriented approach to propose an event modelling framework for modelling complex events and will also facilitate the event detection and event mining process on unstructured videos.

The approaches presented in [16, 17] provide an excellent hierarchical structure for unstructured video content; however, they mainly focus on video content searching and retrieval. While the approach presented in [18], identifies key semantic entities like objects and events and allows users to specify properties and relations between them, it does not make specific commitments regarding the structure of events and also does not provide mechanisms to reason with the annotations. In [19], a CASE based representation of events was extended to strengthen CASE based event representation; however, the proposed model does not focus on spatial aspects of video contents. The main drawback of the approach presented in [20], is that constructing a grammar for a relatively large domain is not feasible, especially taking into consideration the fact that for a human to have complete understanding of a specific

 ISBN: 978-1-61208-320-9

domain and anticipate all possible events is not realistic. The different approaches discussed in [21, 22, 23, 24, 25] provide a flexible hierarchical event modelling structure; however they do not fully explore the aspects of event modelling frameworks which can contain useful information to be used for optimisation of event detection and event mining processes. Moreover, they mainly deal with video contents of multimedia data. Whereas, in our research work we explore the utilisation of other multimedia data streams to model interesting events and explore their importance in event detection and the post event detection mining process.

The remainder of the paper is structured as follows: in Section 2 we will discuss the proposed multimedia event definition framework. In Section 3 we will discuss the limitation of the EDF. In Section 4 we presents Efficient Event definition framework (EDF^E) in detail. In Section 5 we will describe the data structure of EDF^E. Section 6 presents three event modelling examples using EDF^E. Lastly in Section 7 we conclude the work and discuss further research work.

II. PROPOSED MULTIMEDIA EVENT DEFINITION FRAMEWORK

The event modelling framework presented in this paper builds on the Event Description Framework (EDF) proposed in [21]. The advantage of EDF is that it provides a single template predicate for representing all events instead of defining a predicate for each event type. It also provides a set of predicates for describing spatio-temporal relationships between events and entities. Following are some of the reasons why we believe that EDF is the right candidate to provide a framework which can be extended into a promising multimedia event modelling framework for advanced surveillance systems.

- EDF is based upon the object oriented approach where a hierarchy of objects can be defined and their features can be inherited, which means the framework allows a particular event type to be defined as a subclass of another type.
- EDF provides a single template predicate for representing all events instead of defining a predicate for each event type.
- EDF allows a hierarchical decomposition of complex events into simpler events, which can be quite similar to the human approach to describing complex events in the real world.
- Another important element of the EDF framework is that it provides a set of predicates for describing spatio-temporal relationships between events and entities.

Having listed the advantages of EDF, it is also observed that there are some areas of event modelling which are not fully explored in EDF. In our proposed event modelling framework we concentrate on these areas to extend the capacity of EDF, so it can not only represent complex events in surveillance systems but can also provide valuable information for event detection and mining processes. Following are some of the major limitations of EDF which we have addressed in our proposed Efficient Event Description Framework (EDF^E).

III. LIMITATIONS TO OVERCOME

- EDF mainly focuses on the video content of a multimedia stream. However, in certain environments supporting the whole content of a multimedia stream can be very important to model interesting events. For example, in a retail store environment, utilization of ePOS data (text string) in multimedia streams can be used to deter/detect till scanning related frauds. Another example comes from Electronic Toll Collection (ETC) systems where a text stream generated by a scanner-based automatic vehicle classification system (AVC) can be used to initialize/validate the vision-based AVC.
- Although EDF provides a set of predicates for temporal relationships, it does not provide any mechanism to define different granularities of time intervals for those temporal relationships.
- In visual surveillance, there is a need to define virtual scene entities. These entities are not real observable objects but provide contextual scene information as a backdrop against which the events will take place. Two such examples are regions of interest (ROI) and tripwires. EDF does not specifically address the importance of such virtual scene entities.
- EDF provides a set of template predicates for representing video events; however it does not address the utilization of predicates to enhance the functionality of the event detection process.
- EDF does not provide a mechanism to update/store a set of entities in the system. In a surveillance system the number of entities can be large and it is generally not possible to manually store such entities. We address this problem by proposing an event mining framework which explores the relationship between entity feature-sets and associated text strings to generate appearance models of all the entities automatically (see our previous work [26]).

IV. EFFICIENT EVENT DEFINITION FRAMEWORK

In this section, a set of classes for semantic annotation of multimedia data are described along with their properties and relationships. We then present a set of predicates for describing various relationships between events and entities. While explaining each of these concepts we will also discuss how we have extended EDF to confront the limitations described above.

A. Entities

Entities are basically objects such as *car*, *person*, and *chair* observable in a particular domain. For example, while

describing the events occurring at a checkout area of a retail store environment, the various entities could be *till operator*, *barcode scanner*, different shopping items (*milk pack*, *sugar pack*, *butter*, *customer*, etc.) see Fig. 1. Thus for each specific domain we will have a hierarchy of entity classes, where different entity classes can be subclasses of the parent class' entity. For example '*Coke Bottle*' is sub-entity class of parent entity '*Bottle*'.

Let $\mathbf{P_E}$ be a set of parent entities $P_E = \{P_{E1}, P_{E2,}\ldots, P_{En}\}$, where each P_{Ei} can have one or more sub entities class S_E.

$$P_E=\{P_{E1}, P_{E2,}\ldots.., P_{En}\} \quad (1)$$
$$P_{En}=\{S_{E1}, S_{E2,}\ldots.., S_{En}\}$$

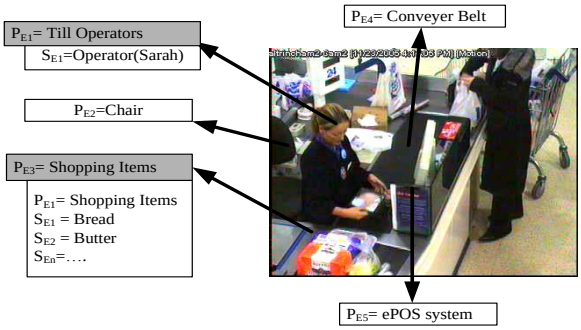


Figure 1. Parent and sub entities classes

These entity classes can have various properties or features associated with them and can also inherit the features of their parent class entity. The set of features for a specific entity class is the union of its own specific features and the features of its super class; for example if 'Coke Bottle' is a subclass of the top level entity class 'Bottle', and if shape and size are features of the super class then the sub class 'Coke Bottle' can have these features plus its own features, as defined below.

$$P_{En}=\{F_1=\text{value}, F_2=\text{value}\ldots\ldots F_n=\text{value}\} \quad (2)$$
$S_{En}=\{F_1=\text{value}, F_2=\text{value}\ldots\ldots F_n=\text{value}\}$ where F_n is feature of an entity.

1) Virtual Entities: In addition to the entities reflecting real life objects, we introduce the user defined virtual entities. These can be manually annotated using a graphical interface. These entities are not real observable objects but provide contextual scene information as a backdrop in which the events will take place. Two such examples are regions of interest (ROI) and tripwires, with their own set of features, for example the features of a tripwire are the spatial locations of start and end points.

Let $\mathbf{UD_{VE}}$ be a set of user defined virtual entity $UD_{VE}=\{UD_{VE1}, UD_{VE2}, UD_{VE3}\ldots.. UD_{VEn}\}$, each UD_{VEi} can have one or more feature values $F_{n,}$ as follows:

$$UD_{VE}=\{ UD_{VE1}, UD_{VE2}, UD_{VE3}\ldots.. UD_{VEn}\} \quad (3)$$
$$UD_{VEn} =\{F_1=\text{value}, F_2=\text{value}\ldots\ldots Fn=\text{value}\}$$

II) Text Entities: The combined knowledge derived from the combination of video and text data is more descriptive than each knowledge source considered in isolation. Based on this fact, it is our conjecture that multi-relational associations should capture more information from the combined metadata (see Fig. 2). In our event modelling framework (EDF^E) we introduce a text entity class to represent the supporting Text multimedia streams. These text entities are used not only to model interesting events but can also provide valuable information to the event mining process, where the association of text and visual entities can be explored to yield valuable information (we have discussed this in [26, 27].

Let $\mathbf{T_E}$ be a set of text entity UD_{VE} $T_E=\{T_{E1}, T_{E2}, T_{E3}\ldots..$ $T_{En}\}$, each T_{Ei} can have one or more labels (L_n) with associated values, as follows.

$$T_E=\{ T_{E1}, T_{E2}, , T_{E3}\ldots.. T_{En}\} \quad (4)$$
$$T_{En}=\{L_1=\text{val}, L_2=\text{value}, L_3=\text{value}\ldots..L_n=\text{value}\}$$
{Id=00384, item=grapes, price=1.68}

Figure 2. Text class entity

B. *Action:*

An 'Action' class refers to actions such as enter, leave, run, walk, that occur in a specific domain. Like entity classes, action classes can be organised in a hierarchy and also have features associated with them (such as speed, angle, etc). Further, they have a patient specification which describes the entity towards which the action is directed, e.g., a 'Coke Bottle' passes over the $Tripwire_1$ where $Tripwire_1$ is the patient. It is important to see that each action will have at least one object entity associated with it. For example, the 'enter' action needs two entities (object which is going to enter and the place, such as $Tripwire_1$ which that specific object is going to enter, as shown in see Fig. 3).

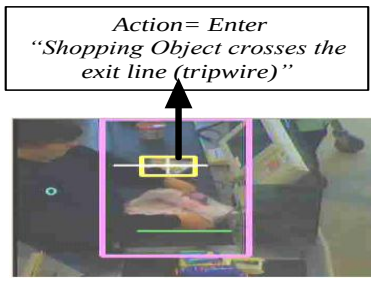


Figure 3. Action class

 ISBN: 978-1-61208-320-9

C. Events

Events are divided into two main categories, that of simple events and composite events.

*I) Simple Events:*These are basically (Actor, Action) tuples, where actor is a set of entities that initiate the event and action is a set of actions performed during the course of the event. For example, in the event of 'Man enters the room', man and room are entities and 'enter' denotes the action (see Fig. 4).

IV)

$$\begin{aligned} \text{Actor}&=\{E_1, E_2 \ldots E_n\} \\ \text{Action}&=\{A_1, A_2 \ldots A_n\} \\ S_{EVENT}&=(\{\text{Actor}_1,\text{Actor}_2 \ldots \text{Actor}_n\}, \{\text{Action}_1, \\ &\quad \text{Action}_2 \ldots \text{Action}_n\}) \end{aligned} \tag{5}$$

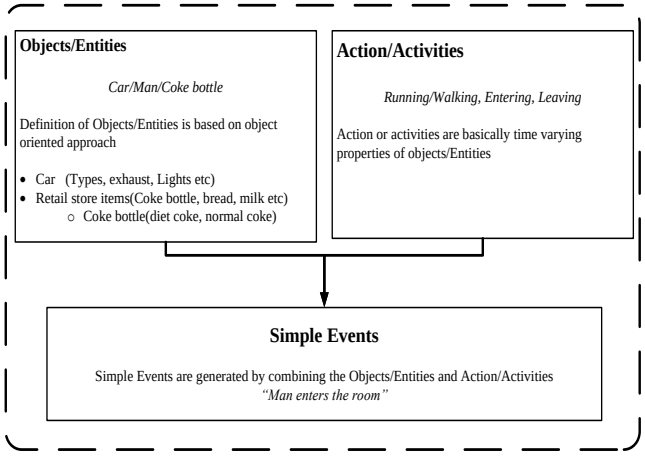


Figure 4. Simple event example

II) Composite Events: Composite events combine two or more simple events and is specified using the predicate PROCESS whose first argument is the event being defined and whose second argument is the composition of other events (see Fig. 5). The compositions of complex events are defined by using the following predicates:

$$\text{Predicate} = \{\text{SEQUENCE, AND, OR}\} \tag{6}$$

- **SEQUENCE** – represents a set of events which happen one after another in a temporal sequence.
- **AND** – represents a set of events with no particular temporal relationship between them.
- **OR** – represents a set of alternate events of which at least one should occur.

III) Additional Predicates: An extensive review of video data from a multitude of sources suggested that these three predicates cannot describe all the events that can occur in surveillance videos. Therefore, we introduce two new predicates (NOT IN, TERMINATE). These predicates not only help in modelling complex events, but they also provide valuable information to the event detection process for effective utilisation of hardware resources.

- **NOT IN:** represents a set of event/events which should not appear within a sequence of other simple events.
- **TERMINATE**: represents when the event detection should be terminated.

$$\text{Predicate} = \{\text{SEQUENCE, AND, OR, NOT IN, TERMINATE}\} \tag{7}$$

IV) Internal Entity Class: In complex events, there can be multiple entities of the same type at different times and locations. Therefore, while modelling complex events the proposed framework should be able to reference an entity which has been already defined in that event. We call such entity an internal entity (IE_{ID}) and refer to it with its ID such as IE_1, IE_4, IE_n.

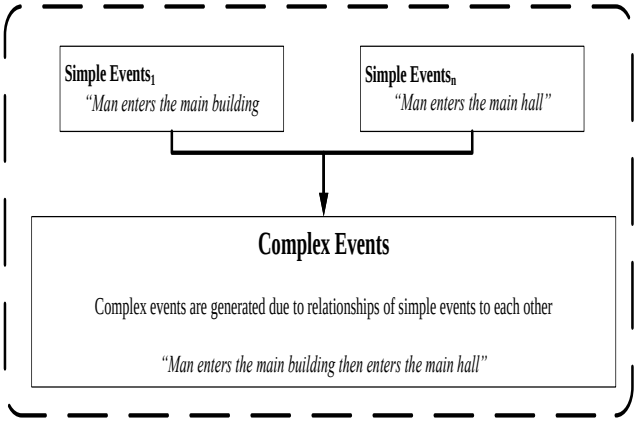


Figure 5. Composite event example

V) *Event Evidence:* Due to the volume of data in surveillance videos, it is important that only relevant information about the detected event should be stored with the constraint that full evidence of detected events is provided as backup. As there is no prior information available as to when a specific event is going to be triggered, the challenge is how to optimise the event storing process. Moreover, due to the nature of each event the requirement of evidence duration to be stored can differ as well. To overcome this challenge we introduce two new features for events: start evidence length (S_{EL}) and end evidence length (E_{EL}). These features' values determine the length of the evidence that needs to be stored for a specific event. For example if the specific event is detected with length of 1 minute and we have S_{EL}=200% and E_{EL}= 100%, then this means that 2 minutes of surveillance video will be stored prior to the detected event and 1 minute of video will be stored after the event being triggered. The main concept is to buffer a specific number of frames and only store them permanently if a

 ISBN: 978-1-61208-320-9

specific event is triggered; this provides vital evidence just prior to and after the event being triggered (see Figure 6).

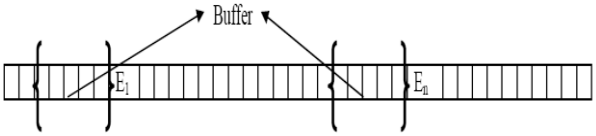


Figure 6. Event evidence buffering

D. Temporal Association Framework

Different predicates were proposed in [21] for defining temporal relationships between two events (e.g., met-by, meets, finishes, finished-by, started-by, starts, during, after, before, overlaps, overlapped-by, contains, simultaneous). Each of these predicates has two arguments and they can be either time intervals or events. These predicates are based on Interval Algebra presented by Allen in [28]; this interval temporal logic is shown in Fig. 7.

AFTER : $T_2^{start} > T_1^{end}$
MEETS : $T_1^{end} = T_2^{start}$
DURING : $(T_1^{start} < T_2^{start}) \wedge (T_1^{end} > T_2^{end})$
FINISHES : $(T_1^{end} = T_2^{end}) \wedge (T_1^{start} < T_2^{start})$
OVERLAPS : $(T_1^{start} < T_2^{start}) \wedge (T_1^{end} > T_2^{start}) \wedge (T_1^{end} < T_2^{end})$
EQUAL : $(T_1^{start} = T_2^{start}) \wedge (T_1^{end} = T_2^{end})$
STARTS : $(T_1^{start} = T_2^{start}) \wedge (T_1^{end} \neq T_2^{end})$

Figure 7. Allen's interval algebra describing temporal logic between $Time_1$ and $Time_2$ [19]

1) Temporal Predicate Granularity: Although these temporal predicates presented in [21] cover most of the temporal relationships to be found in multimedia events, they do not provide a mechanism to define different granularities of temporal associations. We therefore provide granularity (G_n) as a feature of these predicates. Moreover, we introduce two new temporal predicates that can be extensively used in events. These are: the minimum gap (MIN_GAP) and maximum gap (MAX_GAP) with given granularity and its value. These can be use to define the temporal threshold between two events before it can be considered an interesting event.

$$Temporal_{Predicates} = (Predicat_1, Predicate_2, Predicate_3 \ldots\ldots Predicate_n) \quad (8)$$
$$Predicate_n (\{S_{EVENT}/C_{EVENT}\}, \{G_n = value\})$$

E. Spatial Association Framework

We use the spatial association framework defined in EDF [21] to classify spatial relationships into three types: topological, directional and metric.

- Topological relationships are invariant under translation, rotation and scaling and basically describe the properties that characterise the relative position of objects against each other. Studies have shown that all topological relationships can be described using six basic relationships (touch, in, cover, equal, overlap and disjoint) and three operators: "b", which, when applied to an area returns the boundary, and "f" and "t" which return the end points of a line [29].
- Directional relationships are used to describe the relationship involving the relative direction between two objects, such as below, above, right, left, etc. In order to define a set of predicates for describing directional relationships, prepositions in English are used in EDF. The notion behind this is that all directional relationships can be expressed in English using prepositions. The following predicates are defined in EDF – over, upon, opposite, behind, in-front-of, left-of, right-of, above, below.
- Metric relationship involves relationship such as distance, e.g., Distance > 100 that is, applying constraints on spatial metrics. Three predicates are introduced in EDF to represent metric relationships. These are "near", "far" and "at" to specify the location of an entity or event.

V. DATA STRUCTURE

In order to manage the above mentioned ontology of our proposed event modelling framework (EDF^E), we utilise and extend the data schema presented in EDF [21]. The intention is to use relational tables for organising the elements (entities, actions, simple events, predicates and complex events etc) of EDF^E. We describe the different relational tables of the proposed data structure below:

- **viewTbl:** The purpose of this table is to store the information about different camera views available for event modelling /detection. It consists of two fields, *viewID* which is the primary key of the table (primary key is used to store the unique-ID of specific record in table) and *viewDetail* field which stores the descriptive information about the view, such as 'Entrance view', 'Checkout area view', 'Camera view covering clothing section of the shopping mall'.
- **entityTBL** This table stores the information about entities to be used in modelling different events. It consists of *entityID, entityName and parentID,* fields. The *entityID* is a primary key of the table; *entityName* stores the name of the entity such as person, car, object, coke bottle etc. The *parentID* field is used to manage the hierarchy of entities classes. Here *entityID* becomes a foreign key to the *parentID* (the example given in Table I. explains the concept of managing entity hierarchy through *entityID* and *parentID* fields).

TABLE I. EXAMPLE FOR ENTITY HIERARCHY

ID	entityName	parentID	Comments
1	Soft Drink	1	Since the *entityID* and the *parentID* are the same that means it is a top level entity in the hierarchy.
2	*Coke* Bottle	1	The *parentID* points to *entityID* of Soft Drink, that means Coke bottle is sub class of entity Soft Drink

- **entityProperties:** The purpose of this table is to store the different properties of entities defined in entityTbl. The table consists of seven fields: *propertyID, entityID, propertyName, propertyValue, startTime, endTime and viewID*. The *propertyID* is a primary key to the table; *entityID* refers to entityTBL and indicates the entity to which this property belongs. The *propertyName* stores the name of a specific property (for example, size, colour, shape etc). The *propertyValue* is foreign key which refers to the valueTbl table (valueTbl stores the information about the value of the property). The *startTime* and *endTime* fields contain time interval information during which this property is true, the *viewID* field refers to viewTbl and indicates for which specific camera view this property is true.
- **valueTbl:** Since properties of entities can have different types of value, for example the size property of an object can be a number of pixels, whereas the colour property can be a colour histogram stored in a file. Hence, valueTbl is used to manage different types of property values; it consists of *valueID, valueType* and *valueInfo* fields. The *valueID* is the primary key to the table and a foreign key to the entityProperties table; valueType stores the type of value (integer, string, histogram file, text file, etc) and finally *valueInfo* stores the actual value itself, it can be an integer number or file name etc.
- **actionTbl:** The action table stores information about different actions which can take place in simple and complex events. It consists of *actionID, actionName, actionDes* and *patientID* fields. The *actionID* is the primary key to the table and *actionName* specifies the action, such as enter, run, exit, move, etc. The *actionDes* stores the description of action and finally *patientID* field stores a list of *entityID's* (from entityTBL) which are patient to that specific action.
- **simpleEvent_Tbl:** This table stores the information about simple events; it consists of *simpleEventID* (primary key), *simpleEventName* which stores the name of event such as 'Man enters the room', 'object leaving the area'. The *simpleEventString* field contains the event string generated by the EDF^{E}. The *eventActor* field stores a list of *entityIDs* that take part in the event; The *eventActions* field contains a list of *actionIDs* performed during the event; whereas S_{EL} and E_{EL} fields contain the start and end length of evidence to be store for each detected instance of the event.
- **predicateTbl:** The predicateTbl stores the information about different predicates that can be used in complex events. The table predicateTbl consists of *predicateID*, the primary key of the table and used as a foreign key in complexEventTbl. The *predicateType* stores information about the type of the predicate (composite, temporal, spatial). The *predicateName* field stores the name of the predicate, whereas *predicateDes* field contains descriptive information explaining the specific predicate.
- **complexEvent_Tbl:** The complexEvent_tbl table is used to store details of complex events. It consists of *complexEventID* field which is a primary key to the table and the *complexEventName* field which stores the name of the complex event (such as 'Item scanning', 'Tail gating', etc). The *complexEventString* field holds the complex event string generated using the EDF^{E}. The *predicateID* field contains the list of predicates used in the specific complex event (this refers to predicateTbl), whereas *simpleEventsOnly* is a boolean field indicating that this complex event consists of only simple events or a combination of both simple and complex events. The *memberEventIDs* field contains the list of *simpleEventID* and/or *complexEventIDs* used in the specific complex event. Finally S_{EL} and E_{EL} fields contain the start and end length of evidence to be stored for each detected instance of the event (this overrides the S_{EL} and E_{EL} values of included simple events).
- **detectedEventsTbl:** This table stores information about each detected instance of modelled simple or complex events. The table consists of *ID* (primary key), the *eventID* refers to simple or complex event tables against which this instance is detected. The *eventClip* field stores the file name containing video evidence of the detected event, and the *startTime* and *endTime* fields contain the start and end time of each detected instance of the event.

VI. EVENT MODELLING: EXAMPLES

We now provide three examples to explain how simple and composite events can be modelled through the proposed EDF^{E}. The first two examples are of typical surveillance events in a retail store, see Fig. 8 & 9; the third example is based on a view of a secure area (see Fig. 10). In the first example (see Fig. 8), the composite event is defined using virtual entities, text entities, predicates, temporal predicate actor and action elements of EDF^{E}. After defining the main entities, the sequence of two events is defined by using the SEQUENCE predicate along with temporal predicate of MIN_GAP (to represent the minimum gap between the two events), followed by representing the Actor and Action of each events along with their feature sets (such as colour='white', size>=200).

VII. CONCLUSION AND FUTURE WORK

In this paper we presented the efficient event description framework (EDF^{E}), which builds on the event description framework (EDF) presented in [21]. After presenting a general introduction to the event modelling concept, we then discussed the reasons which make EDF the right candidate to be extended for a promising multimedia event modelling framework; this is followed by a discussion of the limitations of EDF which need to be addressed. We then

 ISBN: 978-1-61208-320-9

presented the EDF^E by describing a set of classes for semantic annotation of multimedia data along with their properties and relationships. Next, we presented a set of predicates for describing various relationships between events and entities. While explaining each of these concepts we also discussed how we have extended the framework to confront the current limitations in EDF and the different features of EDF^E which can facilitate event detection and mining aspects of surveillance systems. A modified and extended data structure was also presented to store the ontology of different events. Finally we presented examples to explain how simple and complex events can be modelled through the proposed EDF^E. In future, we will be concentrating our attention on including the event categorisation in event definition framework. That is because due to the nature of certain events in surveillance videos, the number of detected events can be very large. For such events it can be very useful to categorise them into different levels, e.g., using a traffic light system: while detecting an event of miss-scanned items on checkout area; a detected event in which the object's size is relatively large can be categorized as a "Red Event", as the miss-scanned item can be of relatively higher value. Whereas, a detected event in which the object colour is close to the skin colour can be categorized as a "Yellow Event", as this can possibly be a false detected event where a portion of the operator's hand is misclassified as object/item. The categorisation of the detected event can be based on exactness of an event matched to the modelled event as well. For example, if the event is matched with complete certainty then it is categorized as a "Red Event" and if the event is matched with partial certainty to the modelled event then it is categorized as a "Yellow Event" etc. The important question to answer here is: how to measure the exactness of an event matched that is how to know that event is matched 100%, 80% or 60%.

REFERENCES

[1] Gupta, A., T. E. Weymouth, and R. Jain. "Semantic Queries with Pictures: The VIMSYS Model. in Proceedings of the 17th International Conference on Very Large Data Bases (VLDB '91)". 1991. Morgan Kaufmann Publishers Inc.

[2] Andrew, D. W. and F. B. Aaron, "Parametric Hidden Markov Models for Gesture Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence", 1999. **21**(9): pp. 884-900.

[3] Nuria, O., R. Barbara, and P. Alex. "A Bayesian Computer Vision System for Modeling Human Interaction. in Proceedings of the First International Conference on Computer Vision Systems (ICVS '99)". 1999. London, UK: Springer Berlin / Heidelberg.

[4] Matthew, B. and K. Vera, "Discovery and Segmentation of Activities in Video. IEEE Trans. Pattern Anal. Mach. Intell.", 2000. **22**(8): pp. 844-851.

[5] Sibel, A., K. Sel, C. K. Seluck, E. Kutluhan, and V. S. Subrahmanian, "AVIS: The Advanced Video Information System, 1997, Institute for Systems Research Technical Reports.

[6] Sibel, A., K. Sel, C. Su-Shing, E. Kutluhan, and V. S. Subrahmanian, "The advanced video information system: data structures and query processing. Multimedia Syst.", 1996. **4**(4): pp. 172-186.

[7] Hacid, M. S., C. Decleir, and J. Kouloumdjian, "A Database Approach for Modeling and Querying Video Data. IEEE Trans. on Knowl. and Data Eng.", 2000. **12**(5): pp. 729-750.

[8] Carlo, C. "Modeling Temporal Aspects of Visual and Textual Objects in Multimedia Databases. in Proceedings of the Seventh International Workshop on Temporal Representation and Reasoning (TIME'00)". 2000. IEEE Computer Society.

[9] Ramazan, S., Ayg, and Y. Adnan, "Modeling and Management of Fuzzy Information in Multimedia Database Applications. Multimedia Tools Appl.", 2004. **24**(1): pp. 29-56.

[10] Duc, A. T., A. H. Kien, and V. Khanh. "VideoGraph: A Graphical Object-based Model for Representing and Querying Video Data. in ACM International Conference on Conceptual Modeling / the Entity Relationship Approach (ER2000)". 2000. Springer Berlin / Heidelberg.

[11] Duc, A. T., A. H. Kien, and V. Khanh. "Semantic reasoning based video database systems. in 11th International Conference on Databases and Expert Systems Applications". 2000. London, UK.

[12] Yong, C. and X. De. "Hierarchical semantic associative video model. in IEEE International Conference on Neural Networks and Signal Processing". 2003.

[13] Cheng, Y. and D. Xu. "Content-based semantic associative video model. in 6th International Conference on Signal Processing". 2002.

[14] Dönderler, M. E., E. Şaykol, U. Arslan, Ö. Ulusoy, and U. Güdükbay, "BilVideo: Design and Implementation of a Video Database Management System. Multimedia Tools Appl.", 2005. **27**(1): pp. 79-104.

[15] Dönderler, M. E., "Data Modeling and Querying for Video Databases (PhD Thesis), in Computer Engineering2002, Bilkent Unversity: Turkey. pp. 129.

[16] Ekin, A., "Sports video processing for description, summarization and search (PhD Thesis), in Electrical and Computer Engineering2004, The University of Rochester. pp. 166.

[17] Guler, S. and I. Pushee. "Videoviews: A Content Based Video Description Scheme and Video Database Navigator Tool. in Multimedia data mining workshop (MDM/KDD 2002)". 2002. Edmonton , Canada.

[18] Benitez, A. B., et al. "Semantics of multimedia in MPEG-7. in IEEE International Confonference on Image Processing". 2002.

[19] Hakeem, A., Y. Sheikh, and M. Shah, "CASE^E: A Hierarchical Event Representation for the Analysis of Videos, in The Nineteenth National Conference on Artificial Intelligence (AAAI)2004: San Jose, USA. pp. 263-268.

[20] Ivanov, Y. A. and A. F. Bobick, "Recognition of Visual Activities and Interactions by Stochastic Parsing. IEEE Trans. Pattern Anal. Mach. Intell.", 2000. **22**(8): pp. 852-872.

[21] Natarajan, P. and R. Nevatia, "EDF: A framework for Semantic Annotation of Video, in Proceedings of the Tenth IEEE International Conference on Computer Vision Workshops2005, IEEE Computer Society.

[22] Nevatia, R., T. Zhao, and S. Hongeng. "Hierarchical

Language-based Representation of Events in Video Streams. in Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03". 2003. Madison, WI.

[23] Hongeng, S. and R. Nevatia, "Multi-agent event recognition, in Proceedings Eighth IEEE International Conference on Computer Vision (ICCV 2001)2001: Vancouver, Canada. pp. 84-91.

[24] Vu, T., F. Bremond, and M. Thonnat, "Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition, in The Eighteenth International Joint Conference on Artificial Intelligence2003: Acapulco, Mexico. pp. 9-15.

[25] Nevatia, R., J. Hobbs, and B. Bolles. "An Ontology for Video Event Representation. in Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)". 2004. Washington D.C, USA: IEEE Computer Society.

[26] Anwar, F., I. Petrounias, T. Morris, and V. Kodogiannis. "Entity appearance model generation for multimedia events in surveillance videos. in Intelligent Systems (IS), 2010 5th IEEE International Conference".

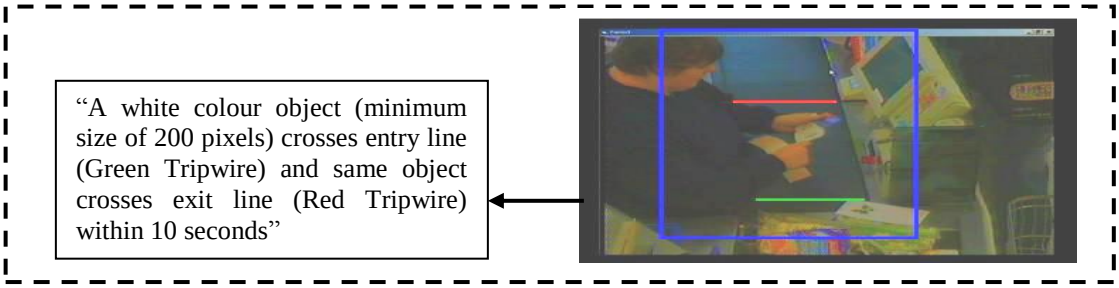
[27] Anwar, F., I. Petrounias, T. Morris, and V. Kodogiannis, "Mining anomalous events against frequent sequences in surveillance videos from commercial environments. Expert Systems with Applications". **39**(4): pp. 4511-4531.

[28] Allen, F. J. and F. George, Actions and Events in Interval Temporal Logic, in Spatial and Temporal Reasoning, O. Stock, Editor. 1997, Kluwer Acadamic Publishers: Dordrecht, Netherlands. p. 205-245.

[29] R., H. G., "An Introduction to Spatial Database Systems. The VLDB Journal", 1994. **3**(4): pp. 357-399.

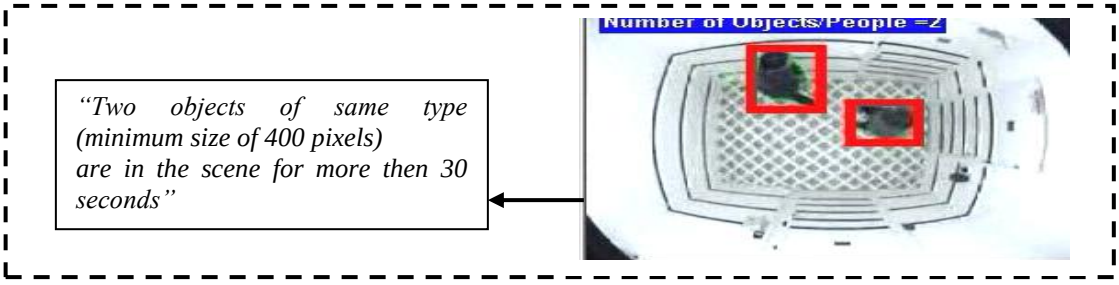
PROCESS(ObjectScanning (Item object, UDVE GreenTripwire, UD_{VE} WhiteTripwire,) SEQUENCE ($Event_1$, $Event_2$, MIN_GAP(G=Second, value<=5)), Actor($Event_1$, class = object, ID=E_1), Action($Event_1$, "Enter", class=GreenTripwire, ID=E_2), Actor($Event_2$, class=E_1, ID=E_3), Action($Event_2$, "Enter", class=WhiteTripwire, ID=E_4)

Figure. 8: Event modelling example (retail store checkout)



PROCESS($Event_{ID}$=3(Item object, UD_{VE}(GreenTripwire,RedTripwire, T_{EN} Text) SEQUENCE ((Event1, Event2), MIN_GAP(G=Second, value<=10)), Actor($Event_1$, class = object, size>=200, Colour='white' ID=E_1), Action($Event_1$, "Enter", class=GreenTripwire, ID=E_2), Actor($Event_2$, class=E_1, ID=E_3), Action($Event_2$, "Enter", class=RedTripwire, ID=E_4)

Figure. 9: Event modelling example (Secure area surveillance)



PROCESS(ObjectCounting(Item object) Event ($Event_1$, G=Second, value=>30) Actor($Event_1$,class=object, size>=400,ID=E1), Actor($Event_1$,class=E_1, ID=E_2), Action($Event_1$,"Remain", class=E_1, class=E_2, ID=E_3)

Figure 10: Event modelling example (Secure area surveillance)