Semantic Characterization of Real World Events

Aparna Nagargadde[1], Sridhar V[1] and Krithi Ramamritham[2]

[1] Applied Research Group, Satyam Computer Services Limited,
14, Langford Avenue, Lalbagh Road, Bangalore 560025, INDIA
{aparna_nagargadde, sridhar}@satyam.com
[2] Indian Institute of Technology-Bombay Powai, Mumbai 400 076, INDIA
krithi@cse.iitb.ac.in

Abstract. Reducing the latency of information delivery in an event driven world has always been a challenge. It is often necessary to completely capture the attributes of events and relationships between them, so that the process of retrieval of event related information is efficient. In this paper, we discuss a formal system for representing and analyzing real world events to address these issues. The event representation discussed in this paper accounts for the important event attributes, namely, time, space, and label. We introduce the notion of *sequence templates* that not only provides event related semantics but also help in semantically analyzing user queries. Finally, we discuss the design for our Query-Event Analysis System, which is an integrated system to (a) identify a best sequence template given a user query; (b) select events based on the best sequence template; and (c) determine content related to the selected events for delivering to users.

1 Introduction

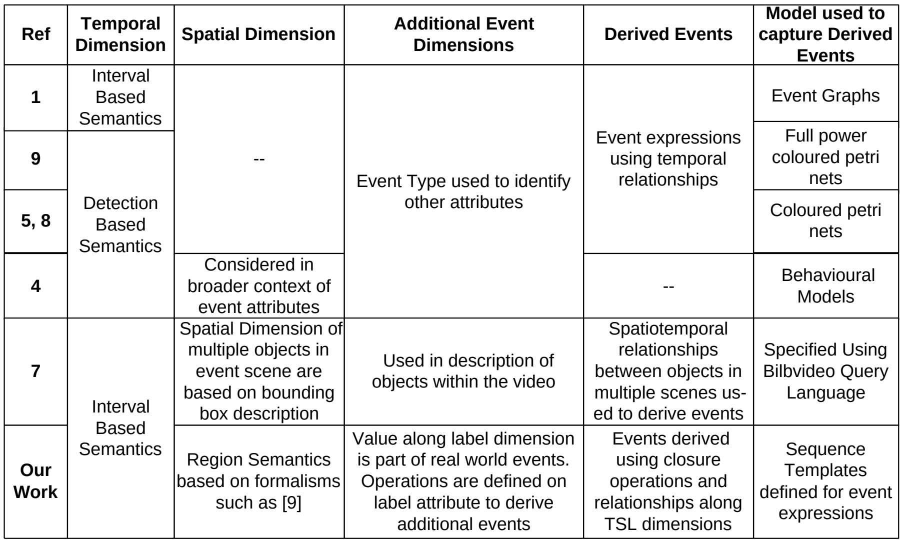
Real world events are of interest to people with diverse interests. For example, when the event 'Cricket match' is in progress, the queries from users could span a wide-range such as "What's the current run-rate?," "How many wickets are down?," "What was the highest total chased successfully on this ground?," "What is the history of matches played here?," and so on. The ability to semantically characterize the events enhances the scope and flexibility of the event management system in answering these complex queries.

There is a need to formally address the issues related to representation and analysis of real-world events. Some of these issues include (a) Characterization of event attributes (b) Identification of event relationships (c) Identification of composite and derived events (d) Derivation of additional information from a set of events (e) Closures on real world events, and (f) Processing of event related information in order to answer user queries, based on a set of events. In this paper, we describe a formal framework to clearly address these issues. In building our formal framework, we draw upon the various advances in the fields of temporal semantics, spatial semantics and event composition in distributed systems. However, though these advances address the various facets of the issue of semantics of real world events, there is no comprehensive formalization of representation of real world events. Real world events are characterized by temporal, spatial and label attributes; the lack of even one attribute would result in an incomplete characterization of the event. For example, the event "Kaif hits half-century at Lords" is incomplete without the time stamp *13th July 2002*, location *Lords* and an event tag (label) *half-century by Kaif.* The formalism presented in this paper represents a holistic approach to the analysis of the temporal, spatial and label attributes of real world events.

The main contributions of this work include a) Event representation in terms of temporal, spatial and label attributes b) The use of domain-specific hierarchies along temporal, spatial and label dimensions for enhanced semantic analysis, c) The definition of event closures in conjunction with these domain specific hierarchies in order to recognize the similarities between otherwise unrelated events d) The use of comprehensive sequence templates for semantic analysis of events and e) A system that aggregates, creates events with a view to answer diverse user queries. We also discuss the problem of event analysis in a real world scenario and present a methodology of identifying *meta events* from an event history by means of sequence templates. Most of the scenarios in this paper are drawn from the realm of international cricket, in particular from the NATWEST series in 2002 [14] A table of the various terms used in the realm of cricket is found in the technical report [3].

1.1 Related Work

Event representation and analysis has been an area of active research. In particular, the temporal nature and properties of events have been widely studied. The representation of time and temporal relationships has been addressed in several papers, notably [2]. Though the temporal and spatial attributes of events have been widely studied, there are very few event specification languages that support a unified view of both these attributes in the case of real world events. Composite event detection by means of using event templates has also been proposed in several papers [4], [5], [12]. But the proposed event templates do not consider the temporal, spatial and label event attributes holistically. Table 1 depicts a comparison of the related work in this sphere. *Derived events* represent all those events that can be generated using the various event operators and closures. In the real world, event input is received through several loosely coupled event sensors/detectors. We employ the temporal frameworks suggested in [11], [12] to order the event input for processing. We subscribe to the use of interval-based semantics [1] for composite event detection. Two aspects are important in the case of real world events: event attributes and event content. Event attributes must be able to provide adequate cues for the automatic generation and dissemination of event contents. For example, in the case of an event *Six* by a batsman in an ODI match, it is required to automatically generate a video clip depicting the batsman hitting the ball directly outside the boundary line. We suggest that the sequence templates, introduced in this paper, can play an important role in this content generation activity. In our related papers [6],[13], we have described issues related to content generation and dissemination.



Ref	Temporal Dimension	Spatial Dimension	Additional Event Dimensions	Derived Events	Model used to capture Derived Events
1	Interval Based Semantics	--	Event Type used to identify other attributes	Event expressions using temporal relationships	Event Graphs
9	Detection Based Semantics				Full power coloured petri nets
5, 8					Coloured petri nets
4		Considered in broader context of event attributes		--	Behavioural Models
7	Interval Based Semantics	Spatial Dimension of multiple objects in event scene are based on bounding box description	Used in description of objects within the video	Spatiotemporal relationships between objects in multiple scenes us-ed to derive events	Specified Using Bilbvideo Query Language
Our Work		Region Semantics based on formalisms such as [9]	Value along label dimension is part of real world events. Operations are defined on label attribute to derive additional events	Events derived using closure operations and relationships along TSL dimensions	Sequence Templates defined for event expressions

Table 1. Comparison of related work

2 Events and Their Representations

We describe an algebra for real world events, and in this respect, every real world happening is an event. Event related information can be categorized into two kinds: formal attributes and informational attributes. Formal attributes form the basis for formally analyzing events. On the other hand, informational attributes provide more information regarding events. An example is a video clip associated with an event. Informational attributes can also be viewed as a bag of attributes. In this paper, we consider time, space and label as part of the formal attribute set of events. Accordingly, we define an event to be characterized along three event dimensions, namely, time, space, and label dimensions. It is apparent that the event specification is complete when an event possesses attribute values along these three dimensions. Hereafter, we shall refer to the time, space and label dimensions as T, S, and L dimensions, and the respective attributes as TSL attributes of an event. We now provide an analysis of the event dimensions and event compositions, and further describe closures related to a set of events.

2.1 Event dimensions

The three event dimensions of time, space and label are each unique and distinct in their characteristics.

The time dimension is continuous and dynamic. The temporal attribute of an event can either specify a time point of occurrence or a time-interval over which the event was observed. The granularity of a time point depends on the *event space*[3] within which the event is defined. For example, the time attributes

[3] An event space ψ is defined as the space encompassing a time interval $\mathrm{T} = < T_s, T_t >$, a region $\Re$ and a set of label hierarchies $\mathcal{L}$[3].

'2004:08:03:05:xx:xx' and '2004:08:03:xx:xx:xx' can both be considered as time points, depending on whether the time granularity is in hours or days. Since events are detected by a distributed network of sensors/detectors; it would be simplistic to presuppose the existence of a global clock. In this paper, we resort to temporal modelling assuming a global reference time as proposed in [11].

We define the spatial attribute as a region that exhibits a physical contiguity and can be well defined using 2D/3D bounds. Using the lower-level representations of spatial attributes, in terms of bounding boxes as a basis [10], we can define the semantics of region bounding operators to describe 'regions'. However, we attach names to these regions for the sake of simplicity. For example, the region 'Trent-bridge' can be described using a set of 2D/3D points that satisfy its region-attributes. Similar to the temporal dimension, the granularity of space attribute is also determined by the event space. Depending on the granularity, a region could comprise of other smaller regions, with well defined spatial relations defining the orientation of the component regions with each other. A few relational operations (`touch, inside`) are described in [7]. The various spatial relationships can be automatically derived using the procedural semantics associated with these regions [10].

Event labels are used to categorize events that occur. Every domain is associated with a set of generic event labels that categorize the various events in that domain. For example, the domain of soccer is associated with event labels such as *Goal*, *Penalty*, *Match*, etc. Event labels can be represented as a hierarchical set, with the root being a generic label, and each child node being a specialization of the corresponding parent. Label hierarchies form an important tool in determining relationships between events, as well as in analyzing composite events. They can also be used to provide some additional information about events such as their frequency, location, sensitivity, and criticality. If label l_1 is a specialization of a label l_2, then $l_1 \rightarrow l_2$. $l_1 \leftrightarrow l_2$ indicates the existence of an alias. Logical operations and, or and not($\neg$) are defined on label attributes.

An event that can be detected by an event sensor is called a *basic event*. Basic events could be sensed or detected automatically, or could be provided as input by an event originator.

2.2 Event composition and event sequences

A *composite event* is an event that is derived using one or more events. The two fundamental event composition operations are *disjunction* and *conjunction*. Event composition along the temporal dimension is a well researched topic. We follow the same temporal semantics as presented in [1]. These semantics follow from the 7 well known relational operators [2] along the temporal dimension namely, before, during, starts, finishes, overlaps, meets and equals. We define the spatial and label semantics for conjunction and disjunction of events. A detailed description of the semantics as well as the relationships on which these semantics are based can be found in the technical report [3]. A more generic event composition operation is the *sequence* operation.

Event sequence : $e^3_{t_3,s_3,l_3} = e^1_{t_1,s_1,l_1} \odot_k e^2_{t_2,s_2,l_2}$
An event e^3 composed of two events e^1 and e^2 with a well defined temporal

ordering forms an event sequence. By the natural definition of a sequence, $t_1 \leq t_2$ is a constraint that needs to be satisfied. The other constraints to be specified could include constraints such as $t_2 - t_1 \leq \mathit{\Delta}$; locations $s_1, s_2 \;\epsilon\; \mathcal{R}$, etc. The sequence operator is $\odot$. The subscript k represents the set of constraints along the TSL dimension that must be satisfied by the two adjacent events in an event sequence. The time of occurrence of the sequence is given by the time interval $t_3 =< min(t_1, t_2), max(t_1, t_2) >$[4]. The region of occurrence of the sequence is given by $s_3 = s_1 \cup s_2)$, where s_3 represents the total region encompassed by the individual regions s_1 and s_2.
An event sequence π can be generalized as an ordering of events $\{e^1_{t_1,s_1,l_1} \odot_{k_1} e^2_{t_2,s_2,l_2} \odot_{k_2} e^3_{t_3,s_3,l_3}, \ldots, e^n_{t_n,s_n,l_n}\}$. The set of all events belonging to a sequence π is represented by E^π. $\kappa^\pi = \{k_1, k_2, \ldots k_n\}$ represents the set of all constraints satisfied by the events in the event sequence π. An example of a generic event-sequence is the 'run-out' event (see Fig. 1 on page 8). A generic event sequence is referred to as a *meta-event*, and the label of a meta-event is called *meta-label.* Further explanations on meta-events follow in section 3.3.

2.3 Event closure

Various types of *event closures* can be defined on real world events so as to enable a quick retrieval of the related events based on a query. We discuss below two types of event closures, namely, *logical closure* and *sequence closure.* The logical closures help in retrieving the events that are logically related along the *TSL* dimensions. Logical closures can be used in analyzing various aspects of basic, conjunct, disjunct, and negation events. The sequence closure describes the closure rules for event sequences. Sequence closures help in determining alternate sequences that can be constructed from logical closures of events in a given sequence π. *Event closures are defined only for basic events.*

Logical closure: We categorize logical closure into two categories: generic closure and semantic closure.

Generic closure, $C_G(e)$, of an event e is used to identify those events that are contained within an occurred basic event. The generic closure on an event $e_{t,s,l}$ is given by
$C_G(e) = \{e_{t_1,s_1,l_1} \mid e_{t,s,l} \;\wedge\; (t_1 \;\epsilon\; t) \wedge\; (\; s_1 \;\subseteq s) \wedge\; (l \;\rightarrow\; l_1)\}$
Consider the event $\mathrm{e}^a_{<2004:03:24:16:30:xx,2004:03:24:18:30:xx>,cricket-field,rain}$. Let event e^a represent rains over the region *cricket-field* during the time interval 16:30 to 18:30 on the 24th of March 2004. The pitch is a rectangular area of the ground between two bowling creases and is at the center of the cricket-field. The batsmen hit the balls bowled to them and run between the wickets on the pitch to score the runs. Since the location pitch is contained in the region *cricket-field*, by generic closure, we have: $e^1_{2004:03:24:16:45:xx,pitch,rain} \;\epsilon\; C_G(e^a)$.

Semantic closures, $C_S(e)$, are closures based on logical implication. Semantic closures have been defined in order to allow the closures related to description of time, space attributes such as *today, yesterday, this city,* and *this block.* Semantic closure also addresses the issue of alias along the label dimension. For example,

[4] $< t_1, t_2 >$ represents a continuous interval of time beginning at t_1 and ending at t_2

on the 25th of March 2004, the semantic label *yesterday* refers to any point of time on the 24th of March 2004. Therefore, $\mathrm{e}_{yesterday,cricket-field,rain} \,\epsilon\, C_S(e^a)$.
The semantic closure of an event e is defined as:
$C_S(e) = \{e_{t_1,s_1,l_1} \mid e_{t,s,l} \wedge t \rightarrow t_1 \wedge s \rightarrow {s_1}^{5} \wedge l \leftrightarrow l_1\}$.

Sequence closure: A sequence closure C_Q is used to determine the closure of a sequence of events in terms of individual events of the sequence. In other words, a sequence closure of an event sequence π is the set of all possible sequences that can be constructed using the events present in the closures of individual events in the sequence such that the sequence constraints are not violated.
Sequence closure of an event sequence π is given by
$\mathrm{C}_Q(\pi) = \{\pi^1 \mid \forall\ e^1 \epsilon E^{\pi^1}\ \exists\ e \epsilon E^{\pi}\ \wedge\ (e^1 \epsilon C_G(e)\ \vee\ e^1 \epsilon C_S(e))\ \wedge\ \kappa^{\pi} = \kappa^{\pi^1}\}$

Let e be an event in the sequence. Then, the sequence closure is used to determine whether there exists any other event e$'$ which can be substituted for e, while still satisfying all the sequence constraints. . It is apparent that if such an event e$'$ exists, it must belong to the closure of event e. π, π^1 are the two event sequences. The events belonging to these sequences are represented by E^{π} and E^{π^1} respectively. Every element in the event sequence π^1, belongs to the generic or semantic closures of the events in the event sequence π. $\kappa^{\pi} = \kappa^{\pi^1}$ indicates that both π and π^1 satisfy the same set of constraints. (Refer sections 2.2 and 2.3 for the notations used.)

Note that the generic and semantic closures are defined only for basic events. As a result, every event that is generated using the closure operation is a valid event and can be derived from the event history H (see 3.2). Sequence closures represent the valid sequences that can be generated by permutations of events generated by logical closures on events of an elementary sequence.

3 Event Analysis

In the previous section, we discussed, in general, about events and event relationships from the point of view of basic events. However, while dealing with real world events and trying to answer queries based on such real world events, there is a need for detailed analysis of the observed events. For example, consider the event set: $\{e^1_{2004:07:07:12:46:30,Manchester,Bowl:Vaughan}, e^2_{2004:07:07:12:46:55,Manchester,Miss:Sangakara}, e^3_{2004:07:07:12:47:15,Manchester,BallHitsPad}, e^4_{2004:07:07:12:47:50,Manchester,OutCalled:Umpire}\}$. In order to deduce that the above set of events depicts an *lbw*[3] event, a proper semantic analysis of the observed events needs to be carried out. The event analysis is done based on an event history H that is compiled from event sets received from one or more event detectors in different locations.

3.1 Event history

Event history H is a set of observed, basic events. Event history H is also associated with a corresponding event space ψ (Recall that event space ψ defines a bound on temporal, spatial, and label dimensions). Typically an event needs to

[5] Let ς denote a verbose translation of a region with respect to an observer; examples include *here, this city* etc. We have $s \rightarrow \varsigma$ if s belongs to the region $\mathcal{R}$ that is referred to by ς. A more detailed explanation of semantic closure is found in [3]

be analyzed in the context of those events that occurred prior to the event under consideration and those that could occur after the event consideration. Such an analysis is required as observed or basic events are quite primitive and are not sufficient to answer the complex user queries. The objective of event analysis is to analyze the events contained in H to derive the interesting meta-events. Note that some of these meta-events could arise due to closure operations, some due to composition operations, and some more due to sequence operations.

3.2 Derived events

Event history H is a set of only the observed, basic events. In order to be able to process queries, it is necessary to augment H and in this subsection, we briefly discuss this augmentation process. In the previous subsections, we discussed several operations related to a set of events and repeated application of one or more of these operations on H is one of the ways to augment H. Specifically, closure and composition operations are helpful in expanding H. The derivation rules for deriving events from H are given below:

1. $e \,\epsilon\, H \rightarrow H \vdash e$
2. $e \,\epsilon\, H \;\wedge\; e^1 \,\epsilon\, C_G(e) \rightarrow H \vdash e_1$
3. $e \,\epsilon\, H \;\wedge\; e^1 \,\epsilon\, C_S(e) \rightarrow H \vdash e_1$
4. $H \vdash e_1 \;\wedge\; H \vdash e_2 \rightarrow H \vdash e_1 \; op \; e_2$, where op represents the conjunction, disjunction or sequence operator.

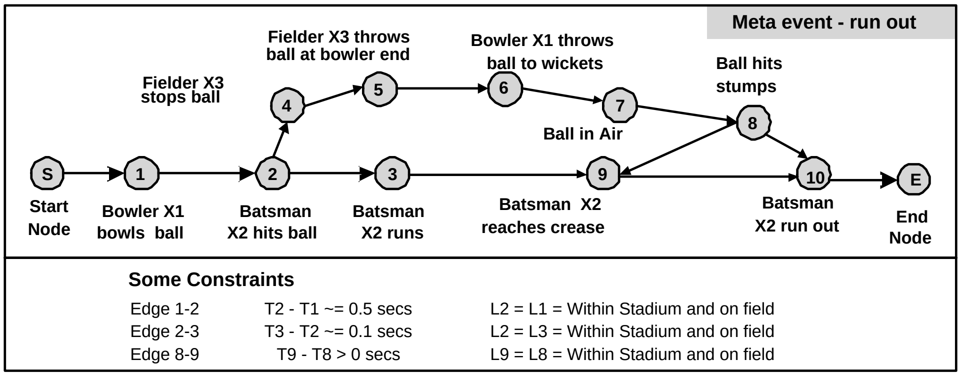
Note that, '$a \vdash b$' is used to denote that given a, the derivation of b is possible by using a set of inference rules.

3.3 Identification of meta-events in H

In order to semantically characterize H, we need some additional information about event space ψ. In this section, we propose the notion of capturing event semantics in the form of sequence templates. A sequence template is semantic characterization of a meta event that addresses the temporal and spatial relationship of a set of events from a semantic point of view. An illustrative sequence template is depicted in Fig. 1. Note that, as Fig. 1 depicts a sequence template, the actual event attributes are left unspecified. Furthermore, a sequence template defines certain important constraints on the event attributes such as temporal and spatial constraints.

Based on domain and related queries of interest, multiple sequence templates are defined and are made part of the *Sequence Template Database* (ST Database). The objective is to analyze H with respect to *ST Database* to generate the G_S, which is a semantic characterization of H. A meta event in G_S is depicted by using only the initial event(e_i) and final event(e_f) of the sequence template and a directed edge from e_i to e_f. The label of this directed edge holds the information of the instantiated sequence template corresponding to the actual meta-event that has transpired.

The event history H, the ST Database and the corresponding semantic representation G$_S$ are used to develop a query-event analysis system.



Some Constraints

Edge 1-2	T2 - T1 ~= 0.5 secs	L2 = L1 = Within Stadium and on field
Edge 2-3	T3 - T2 ~= 0.1 secs	L2 = L3 = Within Stadium and on field
Edge 8-9	T9 - T8 > 0 secs	L9 = L8 = Within Stadium and on field

Fig. 1. Example Sequence Template

4 Query-Event Analysis System (QEAS)

In this section we discuss a query and event processing system that is based on events contained in H. Query Event Analysis System is a formal system that generates responses to user queries using either H or G$_S$ as input. Fig. 2 depicts the functional description of QEAS. The QEAS has two basic functions namely a) Analysis of input events and b) Analysis of user queries. Every new observed event must be made a part of the event history H. The event aggregator adds the new event e to H. It also dispatches e to ST State Machine, in order to verify whether the event e is a part of a meta-event. The ST State Machine matches the event e to the available sequence templates in ST, and suitably updates G_S. A user query Q is first analyzed with respect to the available sequence templates. If a matching template is found, the query is analyzed using G_S as input. Else, the query is analyzed by the Query Processing System using H as input. The result of a query is the set of one or more events that match the query. The appropriate content associated with these events is sent to the user.

4.1 Event analysis

Events are analyzed by using state machines associated with the sequence templates, in order to identify the instantiated meta-events. A new state machine is instantiated when the start of a new meta event is detected. The occurrence of an event could (a) Cause one or more state machines to terminate successfully (b) Cause one or more state machines to make a legal transition (c) Invalidate one or more state machines. (d) Instantiate a new state machine m, which corresponds to a sequence template of a meta-event. Every time a state machine terminates, G_S is updated to reflect the meta events that have taken place. Lapse of time/space constraints could also cause state machine invalidation. The algorithm to generate G_S is shown in fig 3(a).
Theorem 1: With the assumption that no two events occur simultaneously, an occurred event alone is adequate to derive all consequential meta-events in G_S.

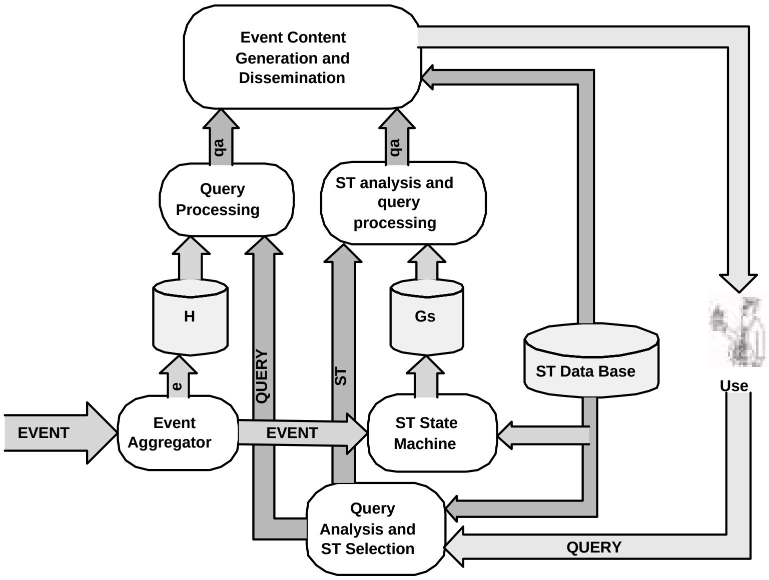
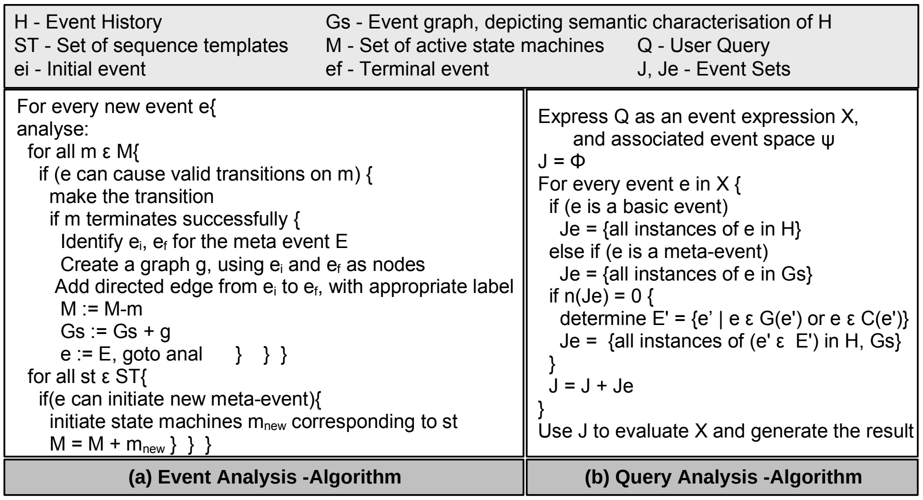

Event Content Generation and Dissemination
qa
qa
Query Processing
ST analysis and query processing
H
Gs
ST Data Base
Use
e
QUERY
ST
EVENT
Event Aggregator
EVENT
ST State Machine
Query Analysis and ST Selection
QUERY

Fig. 2. Functional description of QEAS

A proof of the same is to be found in the technical report [3]. Note that, when two or more events can occur concurrently, the set of concurrent events has to be processed together; and all possible permutations of the concurrent events must be taken into account while deducing the possible transitions.
Corollary 1: If a sequence of events E that cause valid transitions on any state machine M appears in the event history H, then the events e ϵ E are consumed by M to recognize the corresponding meta-event.

4.2 Query analysis

Every event query can be represented as an ordered pair (E, ψ), where E is the event expression and ψ is the event space that corresponds to the user query. As depicted in Fig. 2, Query analysis begins by comparing the input query with sequence templates contained in ST database. As sequence templates capture semantics, using them in query processing enables semantic analysis of a query. The objective of comparison is to select a sequence template *st* that best matches with the input query. This *st* is used in generating the query answer (*qa*) using H and G_S. Finally, the content database is analyzed to extract the relevant content using *st* and *qa* to generate the most appropriate event related content that is delivered to user. The algorithm for query analysis is shown in Figure 3(b). The QEAS analyzes every user query as being equivalent to an event expression. An event expression consists of one or more events combined using the conjunction, disjunction and sequence operations along with the related constraints. The safety and liveness properties of QEAS have been proved in the technical report [3].



H - Event History	Gs - Event graph, depicting semantic characterisation of H	
ST - Set of sequence templates	M - Set of active state machines	Q - User Query
ei - Initial event	ef - Terminal event	J, Je - Event Sets

```
For every new event e{
analyse:
  for all m ε M{
    if (e can cause valid transitions on m) {
      make the transition
      if m terminates successfully {
        Identify $e_i$, $e_f$ for the meta event E
        Create a graph g, using $e_i$ and $e_f$ as nodes
       Add directed edge from $e_i$ to $e_f$, with appropriate label
        M := M-m
        Gs := Gs + g
        e := E, goto anal      }   }  }
  for all st ε ST{
    if(e can initiate new meta-event){
      initiate state machines $m_{new}$ corresponding to st
      M = M + $m_{new}$ }  }  }
```

(a) Event Analysis -Algorithm

```
Express Q as an event expression X,
      and associated event space ψ
J = Φ
For every event e in X {
  if (e is a basic event)
    Je = {all instances of e in H}
  else if (e is a meta-event)
    Je = {all instances of e in Gs}
  if n(Je) = 0 {
    determine E' = {e' | e ε G(e') or e ε C(e')}
    Je =  {all instances of (e' ε  E') in H, Gs}
  }
  J = J + Je
}
Use J to evaluate X and generate the result
```

(b) Query Analysis -Algorithm

Fig. 3. Algorithms for event and query analysis

In general,query processing involves three distinct steps namely (a) query pre-processing (b) retrieval of event information using SQL queries and (c) post processing. the pre processing step involves identification of (a) meta event labels (b) temporal characteristics and (c) spatial characteristics of the input query, and the TSL hierarchies associated with each domain play a significant role in this process. Observe that the input events are analysed in real time to update H, G_S (refer Fig. 2, Fig. 3) which are stored in the database as the basic and meta event tables respectively. The retrieval of event information using SQL queries mainly uses these two tables. Post processing involves filtering and rearrangement of events to suit the user requirements. Such a three step query processing system would help in answering complex queries such as “Generate a 30 minute highlight of the first innings of the cricket match between England and Sri Lanka on the 27th June, 2002 at Trent-Bridge”.

As an example, consider the following query that was posed after the first innings of the match between England and SriLanka on the 27^{th} June, 2002: “Generate the highlights of all boundaries hit by each SriLankan player today?” The query can be mapped to the event-expression:
$\mathrm{E} = (\mathrm{e}_{today,\ trent-bridge,\ boundary\ by\ SriLankan\ player\ X})^{+}$ and the corresponding event space ψ = {today, trent-bridge, {Set of labels in the domain ‘cricket’} }
The simplification of the Event expression, which gets evaluated for every SriLankan player X is shown below:
$\mathrm{e} = \mathrm{e}_{(today,trent-bridge,boundary:X)}$
$= \mathrm{e}_{(today,\ trent-bridge,four:X)} \mid \mathrm{e}_{(today,trent-bridge,six:X)}$ (by generic closure)
$= \mathrm{e}_{(<2002:06:27:10:30:xx,2002:06:27:13:30:xx>,\ trent-bridge,four:X)} \mid$
$\mathrm{e}_{(<2002:06:27:10:30:xx,2002:06:27:13:30:xx>,trent-bridge,six:X)}$ (by semantic closure)

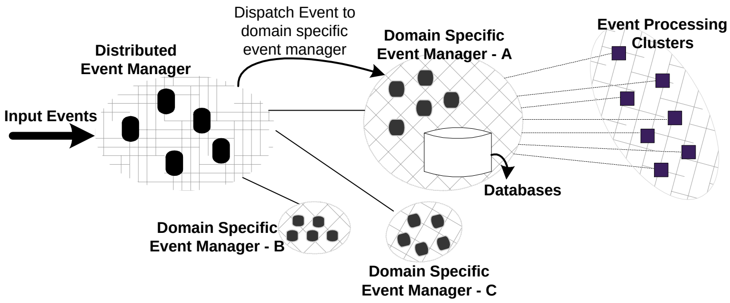

Dispatch Event to domain specific event manager
Distributed Event Manager
Domain Specific Event Manager - A
Event Processing Clusters
Input Events
Databases
Domain Specific Event Manager - B
Domain Specific Event Manager - C

Fig. 4. Illustration of distributed event processing

The events four:X and six:X are meta-events and are a part of G_S. The QEAS looks for the events in G_S in order to evaluate the event expression.

4.3 Distributed Event Processing

It is easy to see that the event analysis and query processing have to meet real time constraints. The processing of any event should preferably be completed before its real time successor occurs. In a scenario wherein there are a considerable number of incoming events belonging to different domains, a distributed event processing network can provide a substantial increase in processing speed. Fig. 4 shows an illustration of a hierarchical event processing system. There exist several clusters of Domain Specific Event Managers that process the events occurring within a particular domain. The Distributed Event Manager routes the incoming events to the domain specific event managers for further processing. The domain specific event Manager manages several event processing clusters and it identifies a suitable cluster to process an input event based on the attributes of the event. For example, location attribute could be used to cluster events that occur in a particular locale.

5 Conclusions and Future Work

In this paper, we have described an approach for representing and analyzing real world events. Events are characterized along three dimensions, namely, time, space, and label dimensions. Such a multidimensional characterization helps in better syntactic and semantic analysis of events. We have defined the notion of event closures and have categorized them into logical closures and semantic closures. We have also introduced the notion of sequence templates that are useful in (a) providing a semantic structure (G_S) to an otherwise a collection of events H; and (b) characterizing content related to meta events. We have described the main steps involved in the generation of G_S given H and have illustrated the use of G_S in answering queries related to real world events. The broad outline of Query Analysis System also describes an important step of event related content identification and dissemination. We have illustrated the proposed formal and query analysis system with the help of a set of real world events.

From the point of view of event related content dissemination, it is required to be able to derive the meta events that are of interest to users as soon a basic event occurs. We are focusing our efforts on how to (a) identify all such meta events; (b) identify related content; (c) identify users who are interested in the occurred and meta events; and (d) efficiently and effectively deliver content. Observe that some of the users could be mobile demanding effective caching and transcoding techniques. Our objective in event representation and analysis is to ultimately deliver content to mobile users with minimum delay. We are in the process of implementing a system for the dissemination of event related information based on the formalism presented in this paper.

References

1. R. Adaikkalavan, *Snoop Event Specification: Formalization Algorithms, And Implementation Using Interval-Based Semantics.* MS thesis, The University of Texas At Arlington, 2002.
2. J. F. Allen, “Time and time again: the many ways to represent time,” *International Journal of Intelligent Systems*, vol. 6, pp. 341–355, 1991.
3. Aparna Nagargadde, Sridhar. V, Krithi Ramamritham.,“Syntactic and Semantic characterisation of real world events,” *Technical Report, Satyam Computer Services Limited*, SCSL/ARG/2004/1, Sept. 2004
4. P. Bates., “Debugging heterogeneous distributed systems using event-based models of behavior,” *ACM Transactions on Computer Systems*, pp. 1–31, 1995
5. Branding. H, Buchmann. A. P, Kudrass. T, Zimmermann. J., “Rules in an Open System: The REACH Rule System,” in *Procs of 1st Intl. Workshop on Rules in Databases*, (Edinburgh), Sept. 1993.
6. Darshan Gujjar, Amit Thawani, Srividya Gopalan, Sridhar V., “An Efficient Web based Event Management System for Distributed Multimedia Services”, *The IASTED International Conference on Internet And Multimedia Systems and Applications (EuroIMSA 2005)*, (Grindelwald, Switzerland), 2005 (to appear)
7. Dönderler. M. E, Ulusoy. Ö, Güdükbay. U., “Rule based spatiotemporal query processing for video databases,” *The VLDB Journal*, vol. 12, pp. 86–103, 2004.
8. Gatziu. S, K. R. Dittrich, “SAMOS: an Active Object-Oriented Database System,” *IEEE Quarterly Bulletein*, Jan 1993.
9. M. Z. Hasan, *The Management of Data, Events, and Information Presentation for Network Management.* PhD thesis, University of Waterloo, 1996
10. Kuijpers. B, Paredaens. J, Vandeurzen. L., “Semantics in Spatial Databases,” *Semantics in Databases, LNCS 1358*, pp. 114–135, 1998.
11. C. Liebig, M. Cilia, A. Buchmann., “Event Composition in Time-dependent Distributed Systems,” in *Procs of the 4th IFCIS (CoopIS 99)*, 1999.
12. Pietzuch. P. R, Shand. B, Bacon. J., “Composite Event Detection as a Generic Middleware Extension,” *IEEE Network Magazine, Special Issue on Middleware Technologies for Future Communication Networks*, pp. 44–55, 2004.
13. A. Thawani, S. Gopalan, Sridhar. V, “Event driven semantics based ad selection,” in *Procs of IEEE International Conference on Multimedia and Expo (ICME’2004)*, (Taipei, Taiwan), June 2004.
14. http://plus.cricinfo.com/link_to_database/Archive/2002/OD_Tourneys /NWS/Scorecards/ENG_SL_NWS_ODI7_07JUL2002_BBB-COMMS.html.