

An Event Model and its Implementation for Multimedia Information Representation and Retrieval

Derik Pack Rahul Singh Sean Brennan* Ramesh Jain

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332
{dpack,rsingh,rjain}@ece.gatech.edu, *sean@gatech.edu

Abstract

This paper presents a novel framework, built around the notion of an event, for modeling, storage, analysis, and querying of multimedia data. We present an event model using which multimedia data, its spatio-temporal characteristics, and a variety of other, potentially flexibly defined attributes can be represented. We also describe the design of a distributed event-based information management system that allows interpretation of multimedia data based on event definitions, storage of such event-based media data, and definition of various types of queries on such information. Examples involving event-based modeling and querying of multimedia data from different settings are presented to illustrate the approach.

1. Introduction

With advances in processing, storage, and sensor technologies over the last few decades, electronic media of different types is increasingly being used to capture and store disparate activities in different situations. Examples vary from news videos of political, social, or sports events, to proceedings of a conference, to personal information such as photographs. Such multimedia information can be thought to electronically chronicle the activities and events that are captured using them [1]. As opposed to the (increasing) ease with which such data can be collected, the problems related to its management, presentation, and understanding the information it contains are becoming increasingly complex. In addition to well studied issues like data volume and complexity of media querying [7], [8], [9], other important factors that contribute to this include fragmentation of semantics across multiple media and the inability of most multimedia data models to account for the fact that people typically conceptualize information not in terms of the media it is represented in, but in terms of the events it is related to. Furthermore, information especially when it is related to the physical world is fundamentally anchored to space and time. People too use space and time as important cues for organizing information. For example, it has been observed that having photographs ordered chronologically is helpful in locating them [2]. Similar conclusions can also be made regarding spatial

information [3]. This implies that both spatial and temporal aspects of information need to be given prominence. Based on these observations, we present in this paper an event-based approach for conceptualizing, storing, and querying multimedia information. We acknowledge the necessity of effective user interfaces in order to achieve this goal, and while this paper does not focus on this area, we point to the following to illustrate work on such interfaces [10, 11]. Our central focus is to use events as a unifying concept to model various types of multimedia data. By building on the intuitive notion of events as occurrences of significance in space-time, we are able to highlight and use the spatio-temporal characteristics of information. The event model we propose is outlined in Section 2. Based on this formal conceptual model, we identify the architectural requirements of a system that can be used to support event-based interpretation, analysis, and querying of multimedia data. The design of this system is also presented in Section 2. In Section 3, we present two examples of how the event-based approach can be used to model and query complex multimedia information and highlight the advantages this offers. Finally we outline the conclusions from our research in Section 4.

2. Event Model and System Description

The fundamental challenge in designing a conceptual model for events lies in balancing the expressible nature of an event with enough uniformity to ensure that related information can be identified and queried efficiently. Our approach to this problem has been to define events through a combination of mandatory and optional attributes. The mandatory attributes of an event include *name*, *time*, *location*, *category*, *participants*, and *data*. These are correspondingly defined as:

- Name is a universal human readable identifier.
- Time denotes the temporal range in which the event occurs.
- Location is the geographic or cyber position of the event.
- Category is the classification of an event which is derived from other information about the event. This attribute is used to group events into collections.
- Participants are the objects that are captured in the media for an event. This may include humans or

objects in the environment that take part in the event environment.

- Data is the heterogeneous media sources that together capture and describe a particular event.

Depending on the category and the media present for an event, the model for that event may have certain optional attributes. These optional attributes vary between categories and are defined by users or domain experts creating the definition for that category. The event model can be used to define relationships that encapsulate events. Although we have not defined all possible relationships between events, we present the following salient ones:

- *Parent-Child Relationship:* This relationship is used to show how events can encapsulate each other. This relationship is defined through the time and location of events. An encapsulated event's time and location are a subset of the time and location of the event that encapsulates it.
- *Category Relationship:* This relationship is found between events in the same category. Since the optional attributes for an event are defined by its category, events in the same category can be compared to one another.
- *Domain relationship:* This relationship is a hierarchical structure that relates event categories to an organization.

An architecture designed using the event model can be broken into several functional groups. These groups are as follows: *event detection, event storage and updates, and event querying and communication.* Event detection is the ability to query media for information related to a particular occurrence. The specific event of interest is identified and the various media analyzed. Various methods have already been created for event detection in audio and video, e.g. [4], [5]. After an event is detected from its associated media a summary of an event is stored. By creating event summaries, it is possible to search for values using certain attributes of an event including time, location, participants, and category of that event. By allowing searches on events, it becomes possible to return all the media relative to a certain value.

Based on the concepts introduced above, we identify the following criteria that need to be supported in a system, for event based processing of multimedia data:

- *Extensibility:* Because of the dynamic nature of information and the many possible ways to detect events, the architecture needs to facilitate run-time assimilation of new components. The communication protocols between system components also need to be extensible to support changes in system modules.
- *Distributed Information Management:* With the heterogeneous nature of data, event descriptions can

involve distinct types of media including images, video, or audio. This media needs to be stored and processed in an appropriate manner, thus requiring distributed information management.

- *Selective Persistence:* Certain domains may require media associated with events to be available all the time. Update agents must be able to effectively cache the media for these domains, without the necessity to make all information persistent.
- *Search Efficiency:* Efficient search and processing of event-based data is essential. This is especially challenging because the event's semi-structured nature complicates search performance.
- *Update Efficiency and Consistency:* The distributed nature of event-based information management requires updates to be supported efficiently and consistently as simultaneous queries and updates may occur. System updates must not become visible until they have been propagated throughout the entire system, to ensure consistency.

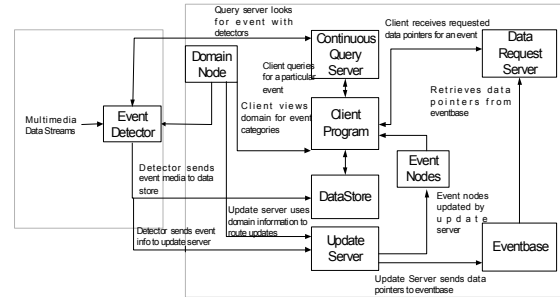


Figure 1. System Description

The system architecture is shown in Figure 1. Event detection can be handled in many ways, including, but not limited to, manual detection, web crawling, or video and audio processing. Event storage is broken up into a domain and event data repository, and a database for data pointers. Event updates are handled by an update server. Events can be queried by client programs and/or a continuous query server. All these components interact using communication protocols.

The information flow in the system can be examined in two ways. The first perspective deals with data flow from the event detectors. The event detectors query media to detect a given set of events that are defined by the domain. When a matching event is detected, the media is stored and an event summary is created. This is sent to the update server. The update server uses domain knowledge to update event categories, and an eventbase which hold the data pointers and a reference to the associated event. The second perspective is from the client. The client is any program or device that can access the domain or the event data for information. Typically the client accesses the domain to find out about the event categories that are present.

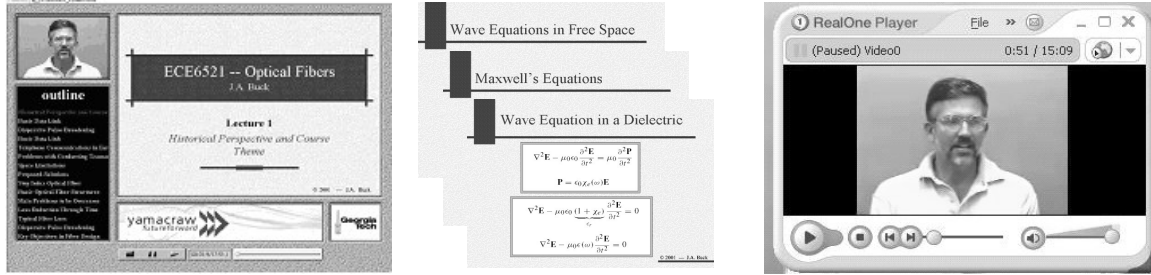


Figure 2. (a) Original Multimedia (structured) data. (b) Results from Query 1 (c) Results from query 2

The client picks the category they wish to view, and accesses the event category to retrieve the associated events. If particular events are of interest, then a request is sent to the data request server to get the data pointers for that event. The client then connects to the data store to access the media for the events. If the particular event is not available in the system but may be available at a future point in time, the client can initiate a continuous query. The continuous query server takes the specified time, location, category, and participants from the client request and translates any requests into the form necessary for any given event detector to understand. The event detectors query their individual media sources until they find an event matching the request sent to them or the job is terminated. Once an event is found using the event detectors, the continuous query server returns the information to the client.

3. Experiments

We present two examples to illustrate the event based interpretation of multimedia data. The first deals with event-based modeling of multimedia information from an instructional/educational setting. At Georgia Institute of Technology, the teleconferencing research group produces multimedia presentations of lectures and places these presentations on CD. Each CD contains a set of presentations that summaries the contents of a class for a given semester. Typically the information includes audio, video, and PowerPoint slides. This media is placed in a set of SMIL [6] presentations where each presentation corresponds with a particular lecture. Each SMIL presentation is used to advance the PowerPoint slides in parallel with the audio and the video. The user is also given the option to move to certain positions within the presentation by clicking on any of the topics of the presentation.

After examining the SMIL presentations (Figure 2a) for the lectures several advantages can be found if an event based model is used. Although each SMIL presentation encapsulates a lecture (a conceptual notion that embodies some of the characteristics of an event), it places the media in a fixed interpretation with limited querying ability. Our event model provides more flexibility by providing different granularity layers. Searches can be issued for information about particular lectures or topics within a lecture. Another

advantage of the event model is its explicit use of time. The SMIL presentations are only related by the order in which they took place. Since the event model uses an exact time, searches could be done using time as an attribute.

This multimedia data is modeled using our event-based approach. Two event categories are defined. The first category is lectures and the second is topics. Topics are defined as sub-events of lectures. Each multimedia presentation coincides with a lecture event. The time, location, and name of the presentation are added to the lecture event, and the topics within the presentation are parsed out from the SMIL presentation for the topic events. This provides a name, time and location for each topic event also. A pointer to the presentation is considered data for the lecture. The data for the topic events are the individual pieces of media that make up the presentation.

Using an event based model, it becomes possible to pose many complex queries that were difficult or impossible to formulate earlier, for example:

- Search for particular media within a time range on a particular topic, e.g.: *Search for all slides on the topic of Maxwell's or Wave Equations in the first two weeks of lectures.*
- Search for particular media on a given date about a particular topic, e.g.: *Search for the first video clip in a presentation that is about Maxwell's equation.*

The example Query 1 returned 11 slides that referenced wave or Maxwell's equations. A few of the slides are shown in Figure 2b. The results for the example in Query 2 are shown in Figure 2c. For the latter query, the SMIL file was parsed by times and the video segment referring to the event was isolated.

The second experiment involved collecting media sources from the internet about Iraq for four weeks. Although there were several ways to structure the information, we decided to create a domain about news, and a sub-domain about Iraq (Figure 3). This sub-domain contained categories of information related to Iraq. The decision for domains was made to ensure that other categories or sub-domains of important news topics could be added without having to change the information being collected on Iraq.

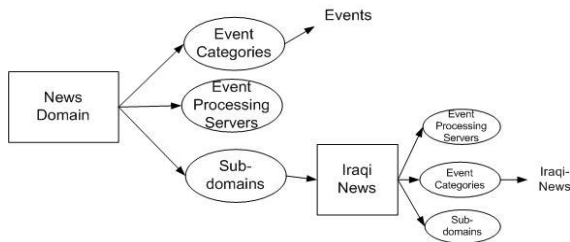


Figure 3. News Domain

Information came into the system by using a crawler that monitored Google News and would take a snapshot of the headlines once an hour. The headlines were parsed in order to gather multimedia sources and to create additions to an event node and the data pointers.

The final results from the crawler provided over 2000 pieces of information on world news and around 80 items on Iraq. We found that placing the information in an event model reduced the complexity of many types of queries. One example would be a query about Iraq during a certain time period, e.g.: *Return all references to Iraq between November 5th and November 10th* (Query 3). Figure 4 shows the distribution of events when Query 3 is applied to various countries for the given time period. This event based approach to news could be used to show trends in news coverage at given periods of time.

Using Google News Query 3 cannot be directly posed. One can sort by time relevance but this requires time consuming manual searching of the data for the information one wants. The event model returns all results that take place during the specified time period and requires no manual searching. Besides time based searches, the event model allowed searches on location, web source, and other optional attributes. The additional ability to search by these attributes made it far easier to search on subsets of events that exist within the larger body of events. The system implementation also provides a necessary separation between the meta-data and the media. This gives the user the ability to decide whether the particular topic is important enough to continue viewing before increasing the system load by viewing the media itself.

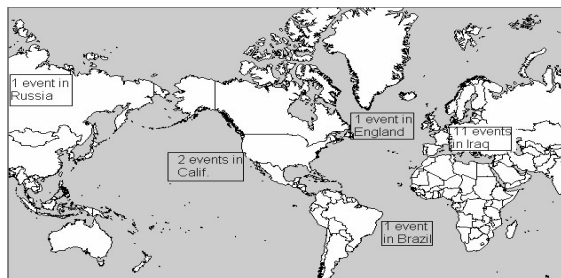


Figure 4. Results of Query 3 applied to several locations

It may be noted, that in the first example, the detectors could use some of the structuring information available in the SMIL file. In the second example however, the

detectors processed a significantly less structured source of information (the web).

4. Conclusions

This paper presents an approach to model multimedia data using the concept of an event. Using an event-based conceptual model, we identify a set of design requirements that guide the development of a storage-processing system architecture. Based on this a system is designed and used for illustrating event-based modeling and querying in two multimedia settings. The system allows for multiple methods of event detection to be used in order to create an event summary. This summary is used to relate multiple types of heterogeneous media to a particular event. This provides any easy method to search for heterogeneous media. System modularity and extensibility are some of the other advantages of this architecture. When better methods for event or object detection are found, they can be added to the system without system shutdown or recompilation. This modularity and extensibility extends beyond the ability to detect events. It also applies to the storage and retrieval of event summaries. Modules can be added to the system that will change where updates are made or how an update is processed. The retrieval format for the system is designed for a set of complex queries and to allow the user to define more complex queries for a given set of events.

5. References

- [1] Jain, R. "Multimedia Electronic Chronicles", IEEE MultiMedia, pp. 102-103, Volume 10, Issue 03, July 2003.
- [2] Rodden, K. and Wood, K. R. "How Do People Manage Their Digital Photographs", Proc. CHI 2003
- [3] Shekhar S., Chawla S., "Spatial Databases: A Tour", Prentice Hall, 2003
- [4] Pavlovic, P. I., Sharma, R., and Huang, T. S. (1997), "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on PAMI*, **19/7**, pp. 677-695.
- [5] Dang, Y., Mangunath, B.S., "Spatio-Temporal relationships and Video Object Extraction". Proceedings of the 32nd Asilomar Conference on SSC, 1998.
- [6] "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification". June 15, 1998 <http://www.w3.org/TR/REC-smil/>
- [7] Srihari, R., Rao, A., Han, B., Muniranthnam, S., Wu, X., "A Model for Multimodal Information Retrieval". International Conference on Multimedia and Expo, 2000.
- [8] Subrahmanian, V.S., "Principles of Multimedia Database Systems", Morgan Kaufmann, 1998.
- [9] Ozden, B., Rastogi, R., Silberschatz, A., "Multimedia Support for Databases", Proc. PODS 1997.
- [10] Ramesh Jain, "Experiential Computing" in Comm. Of ACM, July 2003.
- [11] Bin Wu, Rahul Singh, Punit Gupta, Ramesh Jain. "eVitae: An Event-Based Electronic Chronicle", EDBT 2004