Event-Based Modeling and Processing of Digital Media

Rahul Singh, [1] Zhao Li, Pilho Kim, [2] Derik Pack, Ramesh Jain
School of Electrical and Computer Engineering [1] College of Computing
Georgia Institute of Technology
{rsingh, phkim, jain}@ece.gatech.edu, {[1]gtg981p, [2]gtg009}@mail.gatech.edu

ABSTRACT

Capture, processing, and assimilation of digital media-based information such as video, images, or audio requires a unified framework within which signal processing techniques and data modeling and retrieval approaches can act and interact. In this paper we present the rudiments of such a framework based on the notion of "events". This framework serves the dual roles of a conceptual data model as well as a prescriptive model that defines the requirements for appropriate signal processing. Amongst the key advantages of this framework, lies the fact that it fundamentally brings together the traditionally diverse disciplines of databases and (various areas of) digital signal processing. In addition to the conceptual event-based framework, we present a physical implementation of the event model. Our implementation specifically targets the problem of processing, storage, and querying of multimedia information related to indoor group-oriented activities such as meetings. Such multimedia information may comprise of video, image, audio, and text-based data. We use this application context to illustrate many of the practical challenges that are encountered in this area, our solutions to them, and the open problems that require research across databases, computer vision, audio processing, and multimedia.

1. INTRODUCTION

Starting as a discipline that dealt with pragmatic problems related to storage and retrieval of data, Database research has evolved to deal with structure, organization, and effective use of data and the information they represent [20]. Its results are manifested through a multitude of database management systems that have become essential to business and scientific information [14]. At the basis of various database implementations lies the notion of a *data model* that can be thought of as an abstraction device through which a reasonable interpretation of the data can be obtained [20]. In this context, another area of research that has allied goals is Machine Vision where the ultimate objective is to create a model of the real world using image-based information [13]. In spite of this similarity, Database and Machine Vision research has traditionally evolved independent of each other. The urgency of exploring the intersection of these disciplines is motivated by an upsurge in sensing, processing, and storage capabilities that permeate applications as varied as homeland security and disaster response, (multi-sensor surveillance, biometrics, situation monitoring), life sciences (genomics, proteomics, molecular imaging), medicine (biomedical imaging, patient records), personal information (home videos, digital photographs), and enterprise (business activity monitoring, data warehouses, and data mining). Due to the different data modalities owing to the use of multiple types of sensors, prevalent in each of the aforementioned areas, it may be observed that the true challenges lie at in the intersection of database research and research in various types of digital signal processing, including but not limited to static (image) and non-static (video) processing, audio processing, text processing, as well as processing data from various other types of sensors. Towards this, under the rubric of multimedia research, a number of data models have been developed to address the structure and semantics of media data like images and video [4, 5, 8, 10] or sound [2]. Much of this research has typically focused on the development of powerful features to describe the corresponding media and the use of similarity functions to answer queries based on these features [16]. Such an approach simplifies the general multimedia database problem, because a database is assumed to contain only a specific type of media data [7]. However, emphasizing the semantic coherence between different media is essential to support the premise that multimedia or multimodal information capture provides a more complete picture than any of its constituent modalities considered in isolation.

We approach the problems associated with multimedia modeling from a fundamental perspective by seeking a *general, semantically unifying concept* around which multimedia data can be brought together. Our approach is motivated by the notion of *an event* as the underlying physical reality that generates the information captured using various media. This notion can therefore be used to unify multimedia data in a semantically relevant manner. Furthermore, our definition emphasizes the spatial and temporal characteristics of events. These characteristics are cardinal to information captured from the physical world and are essential for the functional and behavioral understanding of the data and underlying processes.

The rest of the paper is organized as follows: In Section 2 we describe the conceptual data model and present its implementation in the context of modeling multimedia data describing indoor group-oriented activities like meetings. Our implementation takes into account factors related to expressiveness and efficiency of the model. Examples illustrating query and retrieval of media data are also presented here. The signal processing challenges associated with event-based modeling of multimedia data in the contexts considered by us are reviewed in Section 3. Our solution to some of these problems is based on developing a semi-automated methodology that facilitates event detection and identification. Salient to this framework is the use of algorithmic approaches from computer

CVDB'04, June 13, 2004, Paris, France.

vision and audio processing to facilitate human annotation of media data. In Section 4 we present a system architecture for event-based processing of multiple media streams. The paper is concluded in Section 5 with a synopsis of our current work and a discussion on the various open research problems in this area.

2. EVENT-BASED MODELING

2.1 The Conceptual Model

The fundamental idea underlying the data model being considered by us is the notion of an *event* which may be defined as under:

Definition 1: An event is an observed physical reality parameterized by space and time. The observations describing the event are defined by the nature or physics of the observable, the observation model, and the observer.

Certain key issues, in this definition, need to be highlighted; events are treated as a fundamental physical reality and the observations that describe them are defined to depend on the observation model and the observer. For the purposes of our research, the observation model includes among others, the observation method (e.g. audio, video, images, or other media/modalities), sampling model (e.g. video-rate), and sampling period. The role of the observer is fulfilled by users involved in creation or consumption of the media and is especially important because the semantics associated with complex media (like video or images) is *emergent*, i.e. media is endowed with meaning by placing it in context of other similar media and through user interactions [17].

This definition provides us with the central semantic notion, based on which a conceptual model can be developed. As part of the conceptual model, the specification of an event covers three primary aspects:

- *Event information:* The information component of the event may consist of specific attributes. Since events are spatio-temporal constructs, the event information component necessarily contains the time period of the activity and its spatial characteristics, e.g. its location. Additionally, information required to uniquely identify an event are also stored here. Further, entities like people or objects that participate in an event may be described here along with other types of domain specific information.
- *Event relations:* Events (and the activities underlying them) may be related to other events (activities) that occur in the system. Examples of such relations can be temporal and spatial co-occurrences, temporal sequencing, cause-effect relations, and aggregations of events. This information is modeled and described in the event relations component.
- *Media Support:* Each event is a unifying point for the various observations that describe it. These observations are available to us through different types of media data. Specific media data is said to support an event, if it captures (describes) that event. We note that the exact form of the description depends on the characteristics of the media. For example, a basketball game may be described by video, photographs, and mixed text-image new article. Each of these descriptions exemplifies specific media that have different characteristics, while supporting the same event. Such media data may reside as multimedia files in a file system or in media specific databases. In the media support component, information such as media types, resource locators, or indexes corresponding to specific media that support the given event are stored. It should be noted that the conceptual model imposes no restrictions on the same media simultaneously supporting multiple events.

Time and Space are two of the fundamental attributes of the event model and how they are represented significantly impacts our ability to reason with the proposed model. It may be noted that modeling of time and space has received significant attention in both database and knowledge representation communities (see [20] and references therein) and our approach draws significantly from prior results in the area. In the context of temporal representation, a simple approach is to tag each attribute (or tuple) with a discrete timestamp. Its deficiency lies in that common algebraic operations like addition, multiplication, and division are not applicable to timestamps. Further, information that is not explicitly represented becomes difficult to query on. Research in temporal databases has also explored interval-based models of time. Such representations are ideally suited to describe events (such as a game or a meeting) that occur over a period of time. However, the modeling problem we are considering is significantly more complex and can not be sufficiently addressed through interval-based models only. As an illustration, consider the example of parents taking a digital photograph of the "first smile" of their child. Taking the photograph is in itself an event, that has an infinitesimal character (manifested using a single timestamp). Further, based on that single photograph, an interval can not be defined for the event "first smile". In such cases either the fundamental nature of the event or lack of domain semantics precludes the use of interval representations. We therefore propose two temporal datatypes, infinitesimal time points and time intervals to describe events. In the following, we denote time points with a lowercase letter, potentially with subscripts (e.g. t_1, t_2) and time intervals with upper case letters T = $[t_1, t_2]$. Algebraic operators can be used to convert information among these types. For example, time intervals can be added or subtracted from time points to yield new time points. Further, time points can be subtracted to determine time intervals. Three classes of relationships can then be defined to reason about temporal data. These include:

- *Point-Point Relations*: Assuming a complete temporal ordering, two arbitrary time points t_1 and t_2 can be related as: $t_1 < t_2$ (***before***); $t_1 = t_2$ (***simultaneous***); and $t_1 > t_2$ (***after***).
- *Point-Interval Relations:* The relations between an arbitrary time point t_1 and an arbitrary time interval $T=[t_a, t_b]$, are: $t_1 < T \Rightarrow t_1 < t_a$ (***before***); $t_1 \in T \Rightarrow t_a < t_1 < t_b$ (***during***); and $t_1 > T \Rightarrow t_1 > t_b$ (***after***)
- *Interval-Interval Relations:* Given two intervals $T=[t_a, t_b]$ and $U = [u_a, u_b]$, the possible relations between them are [1]:

  $t_b < u_a$ (***before***); $t_a = u_a$ & $t_b = u_b$ (***equal***); $t_b = u_a$ (***meet***); $t_a < u_a$ & $t_b < u_b$ & $u_a < t_b$ (***overlap***); $t_a > u_a$ & $t_b < u_b$ (***during***); $t_a = u_a$ & $t_b < u_b$ (***start***); $t_a > u_a$ & $t_b = u_b$ (***finish***); and the corresponding symmetric relationships (excluding the case for ***equal***).

These relations allow us to deal with relative position of intervals and are necessary to reason about effects that may influence the occurrence of each other (causality) or manifest themselves with delay.

Multimedia data, like photographs and videos have obvious spatial (geographic location) characterization associated with them. Therefore, the ability to reason with space, analogous to reasoning with time is a key component in our model. While we do not address modeling of space in detail here, we point out to the reader that an abstraction such as [9], that supports modeling single objects like points and lines along with collections of objects like partitions or networks can form the basis for modeling space. Such models allow us to deal with spatial characteristics such as topological relationships (*containment, intersection, adjacency, and enclosure*) or numeric spatial attributes like *distance* and *area*.

2.2 The Application Domain

Meetings are an important part of our organizational life and range from informational meetings to brainstorming meetings [15]. Generally, two barriers exist between an attendee and a meeting – time and location. In the last few years, there has been significant attention given to designing smart meeting rooms using pervasive multimedia sensing technologies, such as video, audio and text-image display & presentation system. One major motivation of such systems is to facilitate remote participation. The idea of meeting rooms that provide facilities far more sophisticated than the current video conferencing systems is very appealing to reduce or eliminate barriers of space.

In such settings, different types of multimedia data can be used for different purpose, for instance, video clips can be used to record visual information about an attendee's actions or activities, such as raising hands or asking question. Audio clips can record what the attendees talked about and meeting related documents such as PowerPoint slides or meeting agenda illustrate the key point that the attendees are discussing. In summary, by using the multimedia meeting data, users can re-experience a meeting that happened in the past rather than just learn the result or the decision made in it.

It is necessary to manage the multimedia meeting data using database approaches. This is because: (1) Data volume: on one hand, today, in a well-equipped meeting room, we can easily acquire the multimedia meeting data. While on the other, thousands of meetings happen in our daily life. Moreover, video and audio, the chief components of multimedia meeting data, take a large amount of disk storage. (2) The complex nature of multimedia data types such as video and audio usually necessitate efficient indexing for rapid retrieval. This is especially important since people are typically interested in media related with some specific events rather than the entire multimedia recording.

Our research in this part is directed towards designing a meeting information system to store and index multimedia data consisting of video, audio, PowerPoint files and other media-based information. Towards this, we use the aforementioned event model as a basis to capture and describe multimedia data from a unified perspective. Our approach for designing the meeting information system consists of two steps: the first step involves specifying aspects of event model based on user requirements and domain semantics. The second step involves implementing the model. To do so in this paper, we make use of the techniques from both relational databases and XML databases.

2.3 A Closer Look at Multimedia Meeting Data

We can record many types (modalities) of data from meetings, including video, audio, PowerPoint slides and documents such as agenda or information on attendees.

The salient characteristics of such multimedia meeting data are:

- Semantic relationships exist among the different types of data. For example, there is a semantic relationship between the video clip in which somebody is speaking, the audio clip which records what the attendee is speaking, and potentially the PowerPoint slides – which are related to the discourse
- For purpose of implementation, we classify the meeting data into two categories: static data and dynamic data. Static data catalogue non-time dependent information such as names of attendees while dynamic data focus on describing the process of the meeting, such as how a topic was discussed.

As is well known, most static data are structured. Therefore it is easy to design schemata for static data using relational approaches. On the other hand, most dynamic data are unstructured. Consequently, it is hard to design schemata and manage such data by traditional database approaches. We propose to use a semi-structure data representation (XML) to handle the dynamic data because of the flexibility it provides. However, the problem with XML is its native hierarchical structure that usually causes low efficiency in data retrieval [22], [23]. To address this issue, we implement the data model as a novel combination of XML and relational database technologies.

Commonly in database design, a single type of data model (typically relational or semi-structured) is used to describe the data. Such an approach, however, is incapable of satisfying the problem of modeling both the structured (static) as well as the semi-structured (dynamic) multimedia information from the meeting domain. A nature approach would be to model both the static and dynamic aspects of the information simultaneously within one model. The reason is that the static data and the dynamic data are tightly connected, and play different roles; moreover, user queries usually start with the static information, and then the dynamic information. For instance, the typical start point for querying multimedia meeting data is usually the meeting's title or location. Further queries may then be issued on the meeting process. Additionally, using static information to cluster the dynamic information can accelerate the process of information retrieval because most static data can be stored in tables within a (relational) database, thereby taking advantage of the retrieval speed inherent to the relational paradigm.

2.4 Implementation Technology

Oracle 9i database platform gives us a chance to implement our approach to model the static and dynamic data together. In Oracle 9i, XML data can be stored in specialized column of a table. This allows storage of static and dynamic data within a single table. Such an implementation approach takes care of issues related to both speed and flexibility.

Additional reasons that directed our choice of Oracle as the implementation platform include: (1). The Oracle database is a comprehensive database system, through which we can manage

the multimedia data within a single database. For example, we can manage video, audio and images by using Oracle interMedia [11], handle XML data via Oracle XML database [21], and deal with static data through Oracle relational Database. (2) As described earlier, Oracle can integrate both structured and unstructured data within a single table; we extensively use this capability to combine relational DB design and XML DB design.

In the following sub-sections, we describe our implementation in greater detail.

2.4.1 Modeling Meeting Data

As a real-world example, we captured the multimedia meeting data from regular research group meetings at GATech. The information underlying the meeting structure is roughly composed of three phases: the first phase is "introduction", in which, every group member reviews work done in the previous week and answers questions. The second phase is "discussion" on topic of general interest. This phase may or may not occur in every meeting. The third phase is "presentation", every week a group member presents his or her research. In most cases, there is a PowerPoint file associated with the presentation.

At this stage, we manually parse the multimedia meeting data to generate the events (see section 3 for details on design of a semi-automatic event tagging system). In our initial version of the meeting information system, we used 1 hour of video and audio data, which was parsed into roughly 80 video and audio clips by events. Information from the event-based parsing of the media is stored in an XML file which describes the process of the meeting. The amount of multimedia meeting data used by us is roughly 1,000 MB.

To make the meeting infrastructure clear, we classify the meeting events into two categories: compound events and simple events. Generally, the compound event consists of several simple events. Based on the conceptual model, the relationship between compound events and simple events is an aggregation relationship. In the case of the group meeting data, the compound events come from the meeting agenda and include "introduction", "presentation", and "discussion". Examples of simple events include "ask_question", "answer_question", and "express_idea".

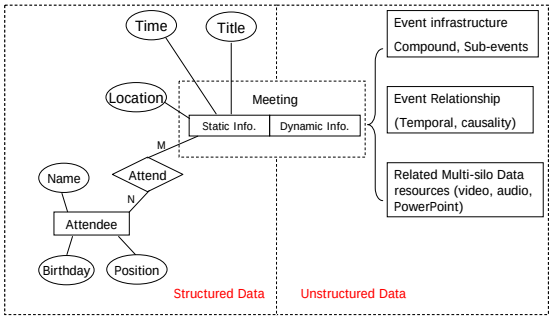

Figure 1. Mixed ER-Semi-Structured Implementation of the data model

The implementation of the event data model is shown in Figure 1. The left part of Figure 1 is a traditional ER model, in which we use the relational database approach to manage static data; we first define entities such as the attendee and static attributes of the meeting. We also consider relationships such as an attendee participating in the meeting. The left part of Figure 1 looks like an ER diagram. The right part of Figure 1 manages the dynamic data by using XML. The corresponding XML schemata are shown in Figure 4 and Figure 5.

Figure 2 shows the tables in the database. As a major distinction from traditional database design, we embed the XML based information directly into a relational table. For example, we save the XML data that describe the events that occurred in the meeting into the Meeting Processing attribute of the meeting table. This design combines both relational Database design and XML database design thereby leading to performance and flexibility.

Meeting

MT#	Time	Location	MeetingProcessing (XMLType)

Attendee

E_mail	FName	LName	Position	Image

Attend Relationship

E_mail	MT#

Figure 2. Tables of the meeting information system

Figure 3 illustrates how XML is used for describing the meeting process. The XML file unifies the multimedia meeting data along with other necessary information such as time and content of events; for example, in Figure 3, from the XML description, we can get the information on the event "Pilho asks_question" (where Pilho is the name of an attendee): To whom the question was addressed, the content of the question, and related video and audio files.

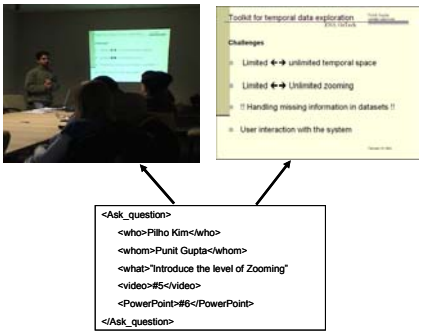

Figure 3. The XML data of the meeting

Figure 4 is the XML schema (generated using the software XMLSPY©) for compound events. In this Figure, three compound events are shown: "Introduction", "Discussion" and "Presentation". Each compound event is associated with some attributes: "who" – the attendee taking part in the event, "what" – the title or basic content of the event, "when"—the time of the event, and "how" – the process of the event. As part of "when" attribute, we record the start time and duration.

In Figure 5, we present details on the "how" attribute of "Introduction" from Figure 4, which illustrates the process of this compound event. The attribute "how" is composed of three simple events: "ask_question", "answer_question", and "express_idea". Each simple event also has attributes such as "who" (who ask the question), "whom" (to whom the question is addressed), "time" (start time and duration), and "what" (the content of the question). The "Multimedia_data" attribute associates video, audio and PowerPoint file with the meeting event.

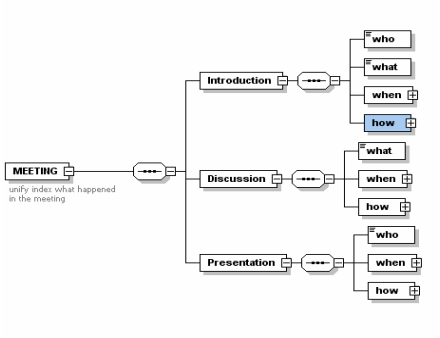

Figure 4. Meeting XML Schema for compound events

In the following we enumerate some example queries that can be issued with the systems. We also present data from preliminary experiments that examine the scalability of our approach.

2.5 Experimental Results

We present two sets of results to illustrate the functionality of our implementation. The first set of results presents examples of event-based query and retrieval of multimedia meeting data. The second set of results pertains to the performance of the system.

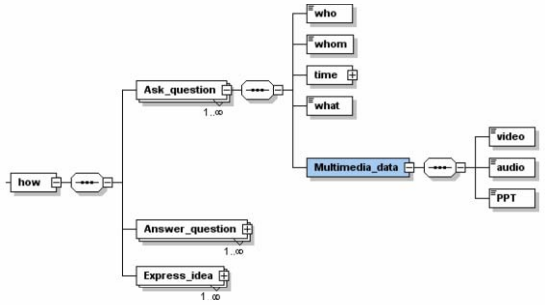

Figure 5. Meeting XML Schema for simple events

2.5.1 Query Events

The structure of the language used by us to query events is similar to SQL and supports operations such as SELECT, FROM and WHERE. To query the XML data, the XPath Query mechanism is embedded in the traditional SQL structure. We present example queries (see Figure 6) on the dynamic data.

2.5.2 Performance Evaluation

The experiments presented in this section were conducted to test the scalability and efficiency of the proposed approach. The parameters of the testing environment used by us were: (1) *Hardware*: We use Dell Precision 450 computer to run our program, the machine had two 2.4G Intel XEON CPU, 2G Memory and 300G hard-disk. (2) *Software*: We used Oracle 9.2 Database to store the multimedia meeting data and used Java (J2SE1.4), JDBC to design the program that uploads the data to Oracle DB, and measures the running time of the program.

Two different types of data model implementations were used in the experiment. The first was the proposed combinations of relational and XML approach. The second was a pure semi-structured (XML) implementation. Data capture and processing was done as follows: We captured one hour video of a group meeting. Associated information from the meeting such as PowerPoint files was also stored. The video was parsed into roughly 80 clips, where each clip was associated with an event (Figure 3). To test the scalability of the proposed approach, we physically replicated the meeting data 20 times, with each replication corresponding to a new meeting. The tables (Figure 2) were populated by serially increasing the meeting number and time (by 7 days) for each new meeting tuple. The structure of the XML-based model is illustrated in Figure 7. In it, we distinguish two different meeting nodes by time and location. We store the pure XML data into an Oracle XMLType table [21].

The video data for each of the 20 meetings, with 80 clips per meeting, was stored in a simple oracle table (media table) along with the clip numbers. The total size of the experimental database is roughly 12 Gigabytes.

The reader may note that in our experiments both the mixed relational-semi-structured implementation and the XML implementation retrieve media from the media table.

Query1
SELECT existsNode(meeting.meetingprocessing, '//Introduction')
FROM meeting;
The '//Introduction' is XPath query. This query tell us whether the Event 'Introduction' happened during the meeting.

Query2
SELECT meetingid, extract((e.meetingprocessing),
'//ask_question[who="Bin Liu" and whom='Punit']/video')
FROM meeting e;
This query returns all the video clips related with the event Bin Liu ask Punit.

Query3
SELECT meetingid, extract(meetingprocessing, '/MEETING')
"Process of meeting"
FROM meeting
WHERE meeting.Location="TSRB309 GATech" and Date = "11/09/04";
This query returns the process of the meeting that held in 11/09/04 in TSRB309, GATech. The result is in terms of XML file.

Figure 6. Examples of event queries

```
<?xml version="1.0" standalone="yes" ?>
- <MeetingSets>
  - <MEETING>
      <DateTime>
        <STime>2004-05-03 16:00:00</STime>
        <ETime>2004-05-03 17:00:00</ETime>
      </DateTime>
      <Location>TSRB0</Location>
      +<Introduction>
      +<Introduction>
      +<Presetation>
    </MEETING>
  - <MEETING>
      <DateTime>
        <STime>2004-05-10 16:00:00</STime>
        <ETime>2004-05-10 17:00:00</ETime>
      </DateTime>
      <Location>TSRB1</Location>
      +<Introduction>
      +<Introduction>
      +<Presetation>
    </MEETING>
  + <MEETING>
 </MeetingSets>
```

Figure 7. The Instance of XML Meeting Data

The queries used to measure the performance are:

Query 4: Issued against the data under the combination of relational and XML implementation:

```
select extract((e.meetingprocessing),
'/meeting/Presentation/how/ask_question/video')
from meeting e
where meetinglocation='TSRB1';
```

Query 5 Issued against the pure XML implementation:

```
select extract(value(e),
'//MEETING[Location="TSRB1"]/Presentation/how/ask_question
/video')
from xmldemotable e";
```

We measure time for queries 4 and 5 as the number of meetings increases from 1 to 20. The running time is defined as the sum of the time to obtain a clip number and the time to retrieve the corresponding multimedia (video, image and PowerPoint data).

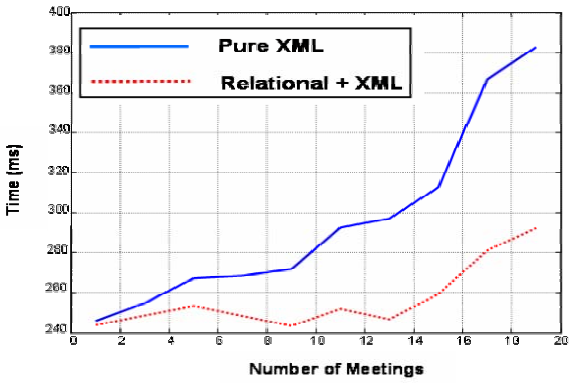

Figure 8. The Performance comparison between the combination of Relation and XML implementation and pure XML implementation

For each data set, we measure the running time for retrieving the records corresponding to the first, middle and last locations. The plot of the average times is shown in Figure 8. The dotted line shows the scalability of the proposed implementation. For purposes of comparison, the scalability of the pure XML implementation is also shown with bold lines. Note that for the XML implementation the entire data had to be searched to ensure that no location information is missed, leading to the observed performance.

3. EVENT BASED TAGGING SYSTEM

A key challenge to the successful application of the aforementioned data modeling, storage and retrieval is the processing of relevant media to determine the events that are represented by them. Towards this goal, we are developing a media event tagging system that combines human expertise with algorithmic processing capabilities. In doing so, we seek to leverage the fact that computers are good at fast low level processing and humans are good at high level analysis and understanding. Our approach to bridging the "signal-to-symbol" barrier is based on a 3-layered event-processing architecture composed of a lower *data event layer*, higher *domain event layer,* and an *elemental event layer* in the middle, which links lower data and higher domains with symbol indexes. A domain event detector provides data-event detectors with domain specific configuration information. Data event detectors, in turn, return enhanced results using the domain configuration information. The role of elemental event detectors is to index to the data events and let the domain event detectors perform appropriate filtering (see [15] for details). For domain modeling, we need to handle complex events in many applications. Complex event analysis can begin with a simple model as described in [6]. Two important challenges in this context are providing humans with an interactive interface to do a domain level event tagging, and providing computers with methods to construct computational causality models based on tagged events. Towards this, in the following, we first explain the necessity of human-computer collaboration in the context of a real-world meeting application. This is followed by a description of the major technical problems faced during processing or storage of such multimedia data. We then demonstrate how we ameliorate these problems by employing database technologies in the processing loop. Finally our approach to computer supported event tagging (annotation) is described.

3.1 Group Meeting Application

A meeting may include informal verbal communication, specific topic including technical words, personal activity, and group activity. Figure 9 shows samples of events related with one group meeting. In it, different colors signify different types of data and duration of each event is marked as width of the box on time line. For meetings our goal is to create the history and well-edited minutes so that its actual progress can be identified clearly. Many of the existing challenges towards this goal are described in the ISL reports on meeting room systems [19]. In addition, many types of static data (e.g. member information, meeting topics, presentation materials, group working reports, etc.) are also involved with group meeting as shown in Figure 9. ICSI meeting Corpus has been constructing broad collections of actual meeting samples with records of their various experiences [12]. Figure 9 represents another fact: that tagging system should be not only be designed for processing media from meetings but also for event recording of personnel group activities.

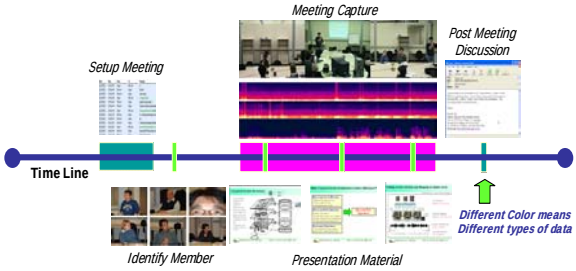

Figure 9. Group Meeting Event Flow

Based on these observations, the requirements of a meeting event tagging system are following: (1) It should help editors by reducing the complexity of multimedia editing (e.g. moving back and forth, zooming in and out, playing repeatedly to understand what is going on, etc.) by using advanced signal processing. (2) It also should provide an easy way for users to create relations between existing data and newly-tagged events both in the context of dynamic multimedia data or static data. (3) All processed data should be stored at the database to be shareable and reusable.

3.2 Media Processing and Role of Database

The flow of speaker identification as an example of media processing is illustrated in Figure 10. For more information on speaker identification, we refer the reader to [3]. In Figure 10, signal processing subroutines for each step are on the left side and information required for processing is on right side. This arrangement can be applied to video processing algorithms also. Traditionally, multimedia research has encountered difficulties in categorizing and managing the data shown on the right side of Figure 10. Usually, this has been solved through definition of proprietary data formats and development of embedded codec to read/write configurations. Another fact, which we observe is that the general goal of developing media processing algorithms that

are self-adjusting to changing environment or robust to arbitrary situations, is often practically infeasible. Because of this programmers usually provide user interfaces to adjust parameters of such algorithms. However, tuning such parameters is complicated because user interfaces (when at all available) do not typically convey the semantics behind the processing but just show a list of numeric values across which the tuning has to be performed (e.g. Sensitivity from 0 to 100). The problem we are indicating here is that there exist many useful media processing applications but they rarely provide proper facilities to store configurations so as to be reusable in a broad range of settings.

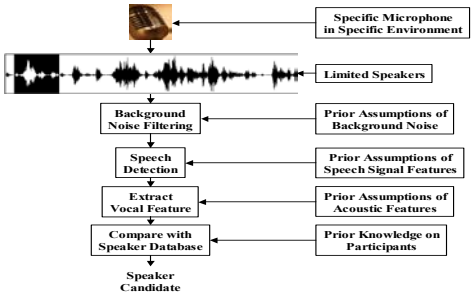

Figure 10. Speaker Identification Flow

Databases can play a significant role in addressing these problems: A prior knowledge of a specific domain can be accumulated with proper domain property tags into databases to be available with any others over the network

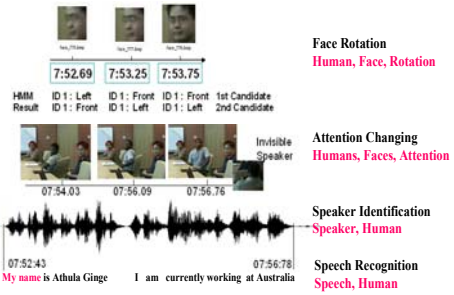

Figure 11. Multimodal Analysis of Meeting Media

3.3 Computational Event Tagging Model

We are developing various meeting media processing technologies to detect semantically meaningful events for meeting summarization as shown in Figure 11. Each processing step in this example requires meeting dependent meta-data including participant information (e.g. Face images and voice features) and models of behaviors in meeting (e.g. attention change caused by speaker change). Since signal processing cannot be perfect, there should be an interactive process to modify errors and a training process to enhance further outputs.

Figure 12 represents the flow of media processing and user tagging in our system. The data network on the left represents distributed data sources. The symbol network represents database including unified indexes of each event. Data types include text, web page or media file when a filter for each data type is available. In case of streaming data, users can do manual annotation part by part with different sorts of filters. Each output of a data event detector is tagged using unified event-based indexes. They are grouped by sources where it comes from, domain information including time and location, and tags attached by user or detected by predefined domain specific knowledge. We provide views of the current media event tagging system, which is under development at the following web address:

http:// esgdevel.oit. gatech.edu:8084 /member/phkim/mets/

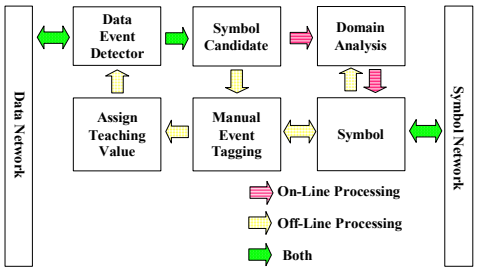

Figure 12: Event Tagging Procedure

4. System Architecture for Event-based Processing

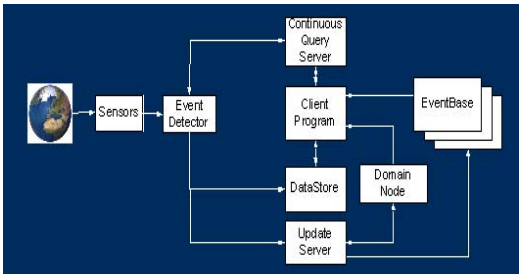

Figure 13. System Architecture

The Event Based Tagging System and the database implementation that are described earlier in this paper can be used as part of a larger system design (Figure 13). This design allows events to be correlated from multimodal sensors and distributed to multiple event repositories.

The flow of information moves from left to right in Figure 13. Sensors capture information about the outside world. The information from the sensors is then sent to event detectors. The Event Tagging System is an example of a complex event detector which provides human knowledge with low level event detection to determine higher level events. It can be used in conjunction with other event detectors in order to detect information from the media. Through the use of multiple instances of the tagging system, various heterogeneous events can be found using the same sensor data. When events are detected, the media they inhabit is saved, and the event is transited to an update server. This update server stores the event in multiple repositories dependent upon the event type and the repository. These repositories are represented in Figure 13 as the "Domain" and "EventBase". The "Domain" provides a hierarchy of events and processes that interact with them. It may be noted that the "Domain" can contain multiple events and "sub-domains". This allows multiple applications to be designed using the same domain. The "EventBase" is used to store events. The database implementation of this paper is one method to implement an "EventBase". As long as the update server is supplied with an interface to a given implementation of an "EventBase", then

multiple implementations can be accessed and updated at a given time.

Client programs can interact with the system by querying the "Domain", the media storage, and the "EventBase" for events. In cases where an event has not been found, the client can send a request to a query server that will monitor the event detectors to find the event that is of interest to the client.

5. CONCLUSIONS

In this paper, we have presented our current research on developing a framework for unified modeling and processing of multimedia data. Our approach for bringing together distinct media in a semantically coherent manner is motivated by the notion of an event as the physical reality that underlies the information that is captured. Based on this, we develop a conceptual data model for multimedia information. Etudes of the model presented here, demonstrate its ability to support temporal and spatial reasoning on complex, dynamic information. The paper presents a physical implementation of this model directed towards storage and querying of multimedia data from indoor group settings such as meetings. Using this implementation we present various types of queries such a system can support as well as preliminary results on its scalability in real-world situation. We also describe a framework we have developed for semi-automatic event identification, processing, and annotation. This framework exploits the integration of various algorithmic techniques from computer vision, speech processing, and image processing by incorporating the human in the computational decision making loop.

Our results indicate a variety of problems that exist in this area and merit further research. For instance, in the broad context of signal processing, is it possible, in non ad-hoc manners, to bring together media dependent processing techniques (e.g. image processing, audio processing) to work in conjunction by aiding each other? Further study is also needed to identify the impact of "top-down" approaches such as the one proposed, on the traditional "bottom-up" processing common to disciplines like computer vision and audio processing. Research problems from a database perspective include, among others, researching system implementations for unified multimedia models, further investigations of the scalability of event-based modeling, and development of methodologies to quantitatively and qualitatively evaluate such unified indexing approaches.

6. REFERENCES

[1] J. Allen, "Maintaining Knowledge About Temporal Intervals", CACM, Vol. 26, No. 11, 1983

[2] T. Blum, D. Keislar, J. Wheaton, and E. Wold, "Audio Databases with Content-based Retrieval", Proc. IJCAI Workshop on Intelligent Multimedia Information Retrieval, 1995

[3] J.P. Campbell, Jr. Speaker recognition: A tutorial In *Proceedings of the IEEE*, 85, 1997, 1437-1462

[4] Lei Chen II, M. Tamer Özsu, Vincent Oria: Modeling Video Data for Content Based Queries: Extending the DISIMA Image Data Model. MMM 2003: 169-189

[5] C. Carson, M. Thomas, S. Belongie, J. M. Hallerstein, and J. Malik, "Blobworld: A System for Region-Based Image Indexing and Retrieval", Proc. Intr. Conf. on Visual Information Systems, 1999

[6] David Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, Reading, MA, 2002.

[7] J. D. N. Dionisio and A. Cardenas, "A Unified Data Model for Representing Multimedia, Timeline, and Simulation Data", IEEE Trans. On Knowledge and Data Engineering, Vol. 10, No. 5, 1998

[8] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System", IEEE Computer, 1995

[9] R. H. Guting and M. Schneider, "Realms: A Foundation for Spatial Data Types in Database Systems", Proc. 3rd Intr. Symp. On Large Spatial Databases, pp. 14-35, 1993

[10] A. Gupta, T. Weymouth, and R. Jain, "Semantic Queries with Pictures: The VIMSYS Model", Proc. 17th Intr. Conf. on Very Large Darabases, 1991

[11] interMedia User's Guide and Reference, Oracle 9.2 document, 2003

[12] A. Janin et al. The ICSI Meeting Corpus. In *ICASSP*, 2003.

[13] R. Jain, R. Kasturi, and B. Schunck, "Machine Vision", McGraw-Hill, 1995

[14] H. Garcia-Molina, J. Ullman, and J. Widom, "Database Systems: The Complete Book", Prentice Hall, New Jersey, 2002

[15] Ramesh Jain, Pilho Kim and Zhao Li. Experiential Meeting System. *ACM ETP*, 2003.

[16] S. Santini and A. Gupta, "Principles of Schema Design for Multimedia Databases", IEEE Trans. On Multimedia, Vol. 4, No. 2, 2002

[17] S. Santini, A. Gupta, and R. Jain, "Emergent Semantics Through Interaction in Image Databases", IEEE Trans. On Knowledge and Data Engineering, Vol. 13, No. 3, 2001

[18] Tanja et al. The ISL Meeting Room System. In *Proceedings of the Workshop on Hands-Free Speech Communication* (HSC-2001). Kyoto Japan, April 2001.

[19] D. C. Tsichritzis and F. H. Lochovsky, "Data Models", Prentics-Hall, New Jersey, 1982

[20] G. Widerhold, S. Jajodia, and W. Litwin, "Dealing with Granularity of Time in Temporal Databases", CAiSE91, pp. 124-140, 1991

[21] XML Database Developer's Guide-Oracle XML DB, Oracle 9.2 document, 2003

[22] H. Wang, S. Park, W. Fan, P. Yu, "ViST: a dynamic index method for querying XML data by tree structures", SIGMOD 2003.

[23] G. Gottlob, C. Koch, R. Pichler, "Efficient algorithms for processing XPath queries", VLDB 2002.