Toward a Common Event Model for Multimedia Applications

Utz Westermann and Ramesh Jain
University of California, Irvine

Although events are ubiquitous in multimedia, no common notion of events has emerged. Events appear in multimedia presentation formats, programming frameworks, and databases, as well as in next-generation multimedia applications such as eChronicles, life logs, or the Event Web. A common event model for multimedia could serve as a unifying foundation for all of these applications.

Events are an elementary concept for humans. Our brains continuously monitor and analyze sensory inputs, recognizing events of importance and initiating actions appropriately. In addition, we episodically organize our memories of the events we've experienced. Events are also an elementary concept in multimedia. They're currently receiving attention in the experiential computing context, which aims to develop multimedia applications that explicitly address users' event-centricity, but they've also long been used in classic areas of multimedia, including presentation formats, content analysis, and databases (see the "Events in Multimedia" sidebar, next page).

Despite the ubiquity of events in multimedia, a common event model for capturing events from applications has not yet emerged. A common conceptual foundation would offer considerable opportunities for reusable components, tools, and techniques for event management and processing—for example, common event databases, event visualization and exploration tools, and event query languages. A common model would also provide a glue for integrating and syndicating events and media from largely isolated multimedia applications and data sources, making possible novel cross-application and cross-data source multimedia services.

This article attempts to create awareness in the multimedia community of the importance of a common event model and incite interest to join our efforts in developing and establishing such a model. In addition to elaborating the prospects and benefits of a common multimedia event model, we derive an extensive set of requirements for a common event model that also establishes elementary taxonomic dimensions for the comparison of event models and their capabilities.

Why a common event model?

As the "Events in Multimedia" sidebar describes, events play important roles in classic areas of multimedia (such as programming frameworks and databases) and also form the backbone of new experiential applications such as eChronicles, life logs, and event-centric media managers. So far, none of these applications share a commonly established notion and model of events; instead, they use proprietary event models defined to suit their immediate application needs.

A common event model would offer the following features:

- *Common base representation*. A common base representation for events could be applied to a broad range of multimedia applications. A common base representation would reduce development efforts and realize gains through reuse. It would no longer be necessary to invent specialized event models every time one develops a new application with interest in events, and the resulting event models would likely be more coherent.

- *Event-centric multimedia*. When users access media, they often require information about events. This is obvious, for example, in news, sports, and surveillance applications. It's also frequently the case when users access personal media. A common event model would promote event-centric content organization throughout multimedia.

- *Unified media indexing*. Most current multimedia databases focus on managing one type of media for one application domain, thereby creating heterogeneous, isolated silos of type- and application-specific content.[1] Events offer a suitable paradigm for integrating heterogeneous media collections and their metadata.[2] A common multimedia event model provides

1070-986X/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

Events in Multimedia

Events are a common concept used in many traditional areas of multimedia computing. In addition, the emerging field of experiential computing has spawned new threads of multimedia research that have a dedicated interest in events. Here, we describe some examples of traditional and emerging uses.

Traditional uses

Multimedia presentation formats such as Synchronized Multimedia Integration Language (SMIL) or Multimedia and Hypermedia information coding Expert Group (MHEG) use events to specify the dynamics of multimedia presentations. They use events to denote state changes in a presentation that influence the presentation's further course—for example, the beginning and end of a video presentation, the video presentation's arrival at a certain time point, or a user interaction such as a mouse click on an image.

Similar to multimedia presentation formats, *multimedia programming frameworks*—such as the Java Media Framework or QuickTime—let users integrate multimedia functionality into applications. These frameworks use events to notify application code about state changes in the media presentation and user interaction.

A common theme in *multimedia content analysis* is detecting events of user interest from content—mostly from video streams and related modalities such as audio and text. The level of events captured by those systems varies greatly. Some systems are concerned with detecting low-level technical incidents within media, such as tracked objects appearing in certain spatial relationships[1] or longer blocks of coherent video frames.[2] Others are concerned with higher-level domain incidents—for example, in news[3] or surveillance videos.[4] In addition, formalisms for inferring domain-level events from lower-level video events have been receiving much attention recently in video analysis. The prominent Video Event Representation Language (VERL)[5] is an outcome of this development.

Some (albeit not many) *multimedia database systems*—for example, the IBM S3 system,[4] the Smooth system,[6] or Snoek and Worring's work[7]—are built on data models that offer dedicated support for the indexing and retrieval of managed media along events: applying content analysis for automatic detection or relying on manual media annotation by the user.

Multimedia frameworks and metadata formats represent another traditional use of events. Given the relevance of events for multimedia content analysis, it isn't surprising that standards and formats for the interchange of multimedia metadata—most prominently MPEG-7—include schemes for representing events occurring in or documented by media. The MPEG-21 multimedia framework and its digital rights management components use events for the standardized specification of access rules to content, denoting use cases for content, such as content played, content copied, and so on.

Emerging areas

EChronicles are information systems for documenting real-world events via multiple media or other suitable sensor data.[8] EChronicles offer tools that let users interactively explore, visualize, and experience the course of events at different levels of granularity. Real-world events and their characteristics form the central units of interest in eChronicles—and not media and their metadata as in traditional multimedia content analysis, databases, and metadata formats. Experimental eChronicles have been implemented for a diverse range of domains, including the chronicling of tennis matches,[9] or meetings.[10]

Life logs, another emerging application, aim to permanently record and document people's activities with media and other sensor data, so that users can go back, review, and explore important events in their lives. Life logs can be considered a subcategory of eChronicles. As with eChronicles,

a unified means for indexing different media types from different applications.

- *Application integration*. Basing the representation of events in multimedia applications on a common model makes it easier to create homogeneous event views based on the same model that syndicate events from different applications. Thus, a common multimedia event model promotes the integration of applications. It also facilitates homogeneous access to and interlinking of events from different applications, thereby potentially giving users insights that they couldn't obtain from one application alone.

- *Common event management infrastructure*. A common multimedia event model creates an underpinning for event management infrastructures that form reusable implementation platforms for many applications. These can include event databases for event storage and retrieval,[3] event-detection systems generating events from media or data streams they observe,[4] event-inference systems deducing events from the occurrence of others,[5] and systems for event propagation between applications.[6]

- *Common event-processing languages*. A common multimedia event model would let users develop languages for processing events in different application contexts. Such event-processing languages might include event-query languages for retrieving and correlating events, event-rule languages for specifying the notion of events is central to life logs. Some life log applications record the daily life events of people with wearable sensors;[11] others record all that media users produce and activities they perform on their computers and relate both media and activities to real-world events, such as that obtained from user calendars.[12]

Humans organize their memory around the events they've experienced. Studies show that this is also reflected in how they organize their personal media.[13] As a result, personal *event-centric media managers* that organize media, especially photos, along the documented events have been appearing.[14]

A vision for the future is to interconnect the events from all of these applications to form an *Event Web*.[8] An Event Web would let users explore events and media documenting them from distributed applications and information sources from all over the world in a uniform manner and discover the links between them. This gives users a much richer experience, providing insights that they couldn't obtain from an isolated application or media collection. The commercial world is slowly risking a glance at this grand vision, as in SEraja's EventWeb service (see http://www.seraja.com).

References

1. D. DeMenthon and D. Doermann, "Video Retrieval Using Spatio-Temporal Descriptors," *Proc. 11th ACM Conf. Multimedia* (Multimedia), ACM Press, 2003, pp. 508-517.
2. M. Fayzullin et al., "The Priority Curve Algorithm for Video Summarization," *Proc. 2nd ACM Int'l Workshop Multimedia Databases* (MMDB), ACM Press, 2004, pp. 28-35.
3. P. Duygulu, J. Pan, and D.A. Forsyth, "Towards Auto-Documentary: Tracking the Evolution of News Stories," *Proc. 12th ACM Int'l Conf. Multimedia* (Multimedia), ACM Press, 2004, pp. 820-827.
4. A. Hampapur et al., "S3-R1: The IBM Smart Surveillance System-Release 1," *Proc. 2004 ACM SIGMM Workshop on Effective Telepresence*, ACM Press, 2005, pp. 59-62.
5. A.R.J. Francois et al., "VERL: An Ontology Framework for Representing and Annotating Video Events," *IEEE MultiMedia,* vol. 12, no. 4, 2005, pp. 76-86.
6. H. Kosch et al., "Smooth—A Distributed Multimedia Database System," *Proc. 27th Int'l Conf. Very Large Databases* (VLDB), Morgan Kaufmann, 2001, pp. 713-714.
7. C.G.M. Snoek and M. Worring, "Multimedia Event-Based Video Indexing Using Time Intervals," *IEEE Trans. Multimedia*, vol. 7, no. 4, Aug. 2005, pp. 638-647.
8. R. Jain, "Experiential Computing," *Comm. ACM*, vol. 46, no. 7, 2003, pp. 48-55.
9. G.S. Pingali et al., "Instantly Indexed Multimedia Databases of Real-World Events," *IEEE Trans. Multimedia*, vol. 4, no. 2, 2002, pp. 269-282.
10. R. Jain, P. Kim, and Z. Li, "Experiential Meeting System," *Proc. 2003 ACM SIGMM Workshop Experiential Telepresence* (ETP), ACM Press, 2003, pp. 1-12.
11. K. Aizawa et al., "Efficient Retrieval of Life Log Based on Context and Content," *Proc. 1st ACM Workshop Continuous Archival and Retrieval of Personal Experiences* (CARPE), ACM Press, 2004, pp. 22-31.
12. J. Gemmell, R. Lueder, and G. Bell, "The MyLifeBits Lifetime Store," *Proc. 2003 ACM SIGMM Workshop Experiential Telepresence* (ETP), ACM Press, 2003, pp. 80-83.
13. K. Rodden and K.R. Wood, "How Do People Manage Their Digital Photographs?" *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI), ACM Press, 2003, pp. 409-416.
14. J.H. Lim and Q. Tian, "Home Photo Content Modeling for Personalized Event-Based Retrieval," *IEEE MultiMedia*, vol. 10, no. 4, 2003, pp. 28-37.

rules for deriving new events from others, and event-notification languages for specifying notification requests to inform applications about events of interest. An underlying common event model—when integrating the events of several applications in a homogeneous view—also permits the languages' seamless application for processing events uniformly across integrated systems.

- *Common event retrieval and mining environments*. Together, a common multimedia event model and common event-processing language provide a foundation for common graphical tools and environments supporting the search for and mining of events. A common base model for different applications will also promote the use of these environments for retrieving and mining events across applications.

- *Common event exploration and visualization tools*. A common multimedia event model lets users create tools for exploring, visualizing, and experiencing the course of events that are reusable in various applications. A common multimedia event model also enables the use of such tools for exploring and visualizing events across applications.

- *Reference*. From an academic perspective, a common multimedia event model provides a reference framework for classifying, comparing, and evaluating event support in different applications.

Requirements

A common multimedia event model must exhibit a number of capabilities and characteristics.

Formal definition

To provide an adequate foundation for developing common event-processing languages, it's important to define the event model formally. Only by effectively defining the event model's primitives and structure can we unambiguously specify the semantics of the query, rule, and notification language primitives. Such semantics permit automatic rewriting and optimization of queries, rules, and notification requests.

A formally defined event model is also valuable for integrating events from different multimedia applications with heterogeneous event models and for the unified indexing of media from different media collections. A precise specification of modeling primitives helps clarify the mapping of these collections' heterogeneous event models or media description schemes into a common homogeneous event view. It also permits a formal specification of these mappings and paves the way to the sound definition of rule languages for such mappings—provided that the event models and description schemes of the applications and media collections being integrated are also formally defined.

For similar reasons, a formal definition increases an event model's suitability for comparing and classifying event support in different multimedia applications.

Media independence

A common multimedia event model must also consider events as first-class entities that are independent of the media documenting them. This differs from traditional multimedia content analysis, databases, and metadata formats where events are usually seen as part of the media description that naturally depends on the medium's existence. Thus, events aren't considered metadata describing a medium; instead, media are considered metadata documenting the course of an event.

Media independence is necessary to establish a base model that suits many applications. Often, events might exist that aren't documented by media. A soccer eChronicle, for example, might contain high-level composite events, such as a league season, that consists of subevents, such as the individual matches in the season. Although the subevents might be extensively documented by media, the composite event might not be directly documented by media at all.

Moreover, real-world events can't be made properties of individual media. In the soccer eChronicle, many different media might describe a match event: the video streams from the different cameras capturing the match from different perspectives, the audio streams from the microphones capturing the stadium roar, photographs of the match highlights, the edited video finally broadcast, newspaper reviews, and so on. It isn't reasonable to tie the match event's existence to any one of those media. It also isn't reasonable to duplicate the event's description with each medium's description.

Uniform global event identification

Along with the demand that events form first-class entities is the natural requirement that a suitable common multimedia event model provide means for the global (that is, media-independent) unique identification of individual events. Given that the model should also be a basis for integrating and correlating events from different applications, the event model's identification system permits the unique identification of events in a uniform fashion, regardless of the events' sources.

Discreteness and continuousness

Two basic views of events are prevalent. The first view regards events as discrete, instantaneous incidents or state changes in a system. In multimedia, discrete events often appear in multimedia presentation formats and programming frameworks. Examples include a "begin" event, denoting the instantaneous state change in a Synchronized Multimedia Integration Language (SMIL) presentation that the medium's presentation has started, or a "photo taken" event in a life log application.

The second, more cognitive view regards events as incidents or activities with experiential relevance to people possessing an extension in time. Users frequently encounter such continuous events in eChronicles and media managers. In the soccer eChronicle example, league seasons, match days, matches, and intramatch events are continuous events, just like a vacation documented in a digital photo album.

To provide a base model for representing events in a broad variety of contexts and allowing the integration of possibly discrete and continuous events from different applications, a common multimedia event model should be able to represent both categories of events.

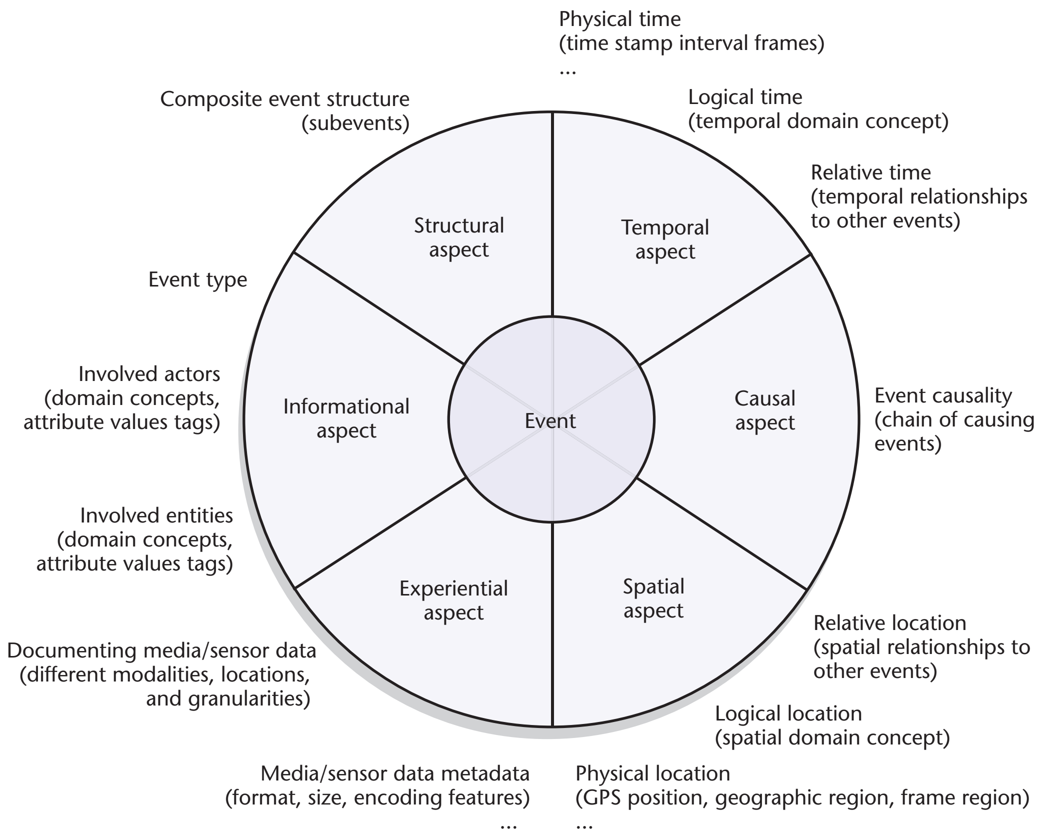


Figure 1. Basic aspects of event description.

Expressiveness

An event model that aims at establishing a common foundation for a wide diversity of applications, event-centric multimedia, reusable event management infrastructure and tools, and application integration should address several elementary aspects of event description (see Figure 1).

Temporal aspect. Events are inherently related to the concept of time. It isn't surprising that an event's temporal aspect is often of much interest for an application. For instance, this aspect is relevant in the soccer eChronicle when penalties occur.

Applications might be interested in an event's temporal aspects on a physical level, such as the event occurrence's time stamp and duration (in the case of continuous events). Depending on an application, different temporal units of measure might apply. You could express the time of an event's occurrence in a global time measure (for example, date, time, and time zone), using a relative time measure (such as the game minutes in the soccer example), or in relation to media (for example, the frame numbers of a video documenting an event).

Applications might also be interested in time on a logical level by referring to an application domain's abstract temporal concepts. It might be important for the soccer eChronicle that the penalty took place in overtime and in the last match of the season.

Independent of the description level, applications might need to express events' temporal aspects not just absolutely by referring to global time points and intervals, but also relatively by expressing events' temporal relationships to other events, using interval relations, for instance. The soccer eChronicle might want to explicitly express that during the abovementioned penalty, a brawl occurred between the opposing teams behind the referee's back.

Thus, a suitable common multimedia event model should comprehensively capture an event's temporal aspect, both on a physical and logical level, and in an absolute or relative manner.

Spatial aspect. A common multimedia event model should also show location awareness and support different ways of capturing the spatial aspect in an event's description.

The soccer eChronicle might be interested in a foul play's exact physical position on the playing field or the area covered by a continuous event, such as a counterattack followed by an intercept. Different units of measurement—global, local, and media-related—might be suitable for expressing the spatial aspect on a physical level.

On a logical level, the eChronicle might also be interested in an event's relationships to a domain's spatial concepts, such as in the fact that a foul occurs in the penalty area or that a match takes place in a certain stadium.

Analogously to the temporal aspects, applications might want to express an event's location not only in an absolute manner but also relatively to other events' locations (for example, using topological relations).

Informational aspect. An event model should provide information about the events that occur. This information includes the event type, such as a free kick in the soccer eChronicle. As with many other type systems, we can taxonomically organize event types. A free kick event constitutes a match event, a soccer event, and, more generally, a sports event.

Adequate coverage of the event's informational aspect might require further description. This can include the actors and entities involved in an event (for example, players Jens Lehmann and Esteban Cambiasso and the ball) and their roles (goalkeeper, striker, ball, and so on). It might also involve further parameters describing the event (such as the new score) or the entities (such as the players' numbers).

Depending on the application, different methods might be adequate to capture actors and entities involved in an event. Users could refer to predefined concepts from a domain ontology or database, encode actors and entities in attribute values, or apply free text tags provided in collaborative folksonomy-style applications.

Experiential aspect. Multimedia applications should offer users engaging ways of exploring and experiencing a course of events to let them gain insights into how the events evolved. Access to rich media documenting events is a natural prerequisite for this. Thus, although we've demanded that events exist independently of media, a common event model that aims to serve as a base model for multimedia applications must show media awareness and let events refer to such media.

An event model's media referencing scheme should be capable of addressing media of different types, ranging from traditional discrete and continuous media such as images or videos and complex media such as multimedia presentations to essentially any kind of sensor data that's available on the course of events, such as a stream of ball positions in the soccer eChronicle. It should also be able to address media from different sources, ranging from file systems, Web servers, streaming servers, and content management systems, to multimedia databases. This lets the model serve as a basis for the unified indexing of different media types in different media collections.

Further, the referencing scheme must be aware of the media's structure, allowing references to complete media as well as to parts of media. Because of the different granularity of events, the user must be able to select the media parts that document a given event, such as a spatial region in an image or a temporal segment in a video. Although it's sufficient for a soccer match event to refer to the full 90-minute TV broadcast as a documentation of its course, it would not be adequate for a 30-second subevent goal to only refer to the full broadcast.

A media-aware event model should also permit the representation of basic metadata about documenting media, such as the media type, format, and size, or features such as color histograms, textures, and audio spectrums. This opens up content-based access to events from within event-processing languages, retrieval and mining environments, and visualization and exploration tools founded on the model. This can also guide an application in selecting which media that's documenting an event to present to the user.

Structural aspect. Events are a modeling concept that's applicable at many different levels of abstraction. In the soccer example, we could aggregate low-level in-media events such as person-, object-, and text- (number) recognition events to intermediate in-match events such as shots, fouls, and goals. We could further accumulate these events to higher-level events such as the match, accumulate matches to seasons, and so on.

Thus, the exploration of the subevents that occurred as part of a more complex event offers important insights into an event's course. A suitable common multimedia event model should consequently support the representation of event composition.

Causal aspect. Offering answers about an event's cause is another essential task of many applications. In the soccer eChronicle, users are likely interested in the chain of events that led to the dramatic turnover of last year's champion-league final in the second half. A suitable common multimedia event model should therefore offer means to express causality and permit the explicit representation of chains of causal events for individual events.

Complex data support

Time intervals, geographical positions and regions, references to media, intramedia selectors, and media features are all examples of data types that an event model might need to capture for event description and that quickly go beyond the complexity of primitive types such as strings, integers, or float values. To provide a suitable foundation for an efficient event-management infrastructure or event-mining tools and environments, a common event model should be capable of representing such complex data in a type-adequate manner.

Association support

Not only are the occurrence of events of interest and the media documenting them important, but so are the relationships between events. A common event model should be able to express events' temporal and spatial relationships as well as their structural and causal relationships and offer a dedicated notion of associations.

It isn't enough to implement associations between events through pointers or foreign key attributes. Such an approach makes associations parts of the events and thus superimposes a global view onto events and their interrelationships. But different users—for example, security experts working with a building-surveillance eChronicle—might have different, even conflicting views of the events that led to the theft of a product prototype. By supporting event associations as an explicit construct, an event model lets users establish different views of the same set of events. It also lets users integrate events from different applications, which might interrelate events along different dimensions.

Knowledge awareness

Reasonable event processing, retrieval, mining, exploration, and visualization require access not only to events and their descriptions but also to general knowledge of the domain in which the events are embedded. In the soccer eChronicle, for example, users might want to know not only the players involved in a match event, but also their affiliation (for example, team, number, and nationality) and season statistics (such as goals and yellow cards).

A common event model should be able to express events' temporal and spatial relationships as well as their structural and causal relationships.

A common event model should be aware of such domain knowledge. The model should provide referencing schemes that permit events to be related to entities and concepts from a broad variety of knowledge sources—such as external databases, knowledge bases, ontologies, and taxonomies—to establish bridges for multimedia applications to domain knowledge.

Uncertainty support

The detection of events from media or sensor data is imprecise and subject to varying but generally limited degrees of reliability. To provide a framework that lets users adequately represent the output and the integration of media and sensor data analysis tools and event detectors, a common event model should reflect event uncertainty. Uncertainty is not only important at event granularity, but also with regard to event properties, which we can extract from media and sensor data with only limited precision. A common event model should let users express uncertainty and ambiguity with regard to these properties or events' participation in associations.

Capturing uncertainty in a common event model has implications for the development of event-processing languages and software components that use these languages. Consideration of uncertainty makes event querying more of a probabilistic information-retrieval task than a classic database-querying task. Likewise, event-rule and event-notification languages must incorporate probabilistic methods to handle event detection and inference in the presence of uncertainty.

Extensibility and adaptability also promote the applicability of an event model for the integration of events from heterogeneous applications.

Extensibility and adaptability

Multimedia applications will describe events in considerably different ways. For example, a soccer eChronicle will characterize events using different types, properties, and associations from those a meeting system uses. To support many diverse applications and let users develop common event management infrastructures, event retrieval and mining environments, and event exploration and visualization tools, a common event model must be both

- extensible, to let users add the event types, properties, and associations an application requires; and
- adaptable, to let users select the representations for the different event description aspects (described earlier) that best suit the application's needs.

Extensibility and adaptability also promote the applicability of an event model for the integration of events from heterogeneous applications. The event models of the applications being integrated might include properties and association types that can't be uniformly mapped into a common event view. Instead of ignoring these properties and association types in the common view, extensibility and adaptability permit such heterogeneities to be preserved.

An event model offering a high degree of extensibility and adaptability should include an event schema or ontology language that lets users specify the configuration of a given domain's model, including the applicable event types, properties, and associations.

Current technology

Given these essential requirements for a common event model for multimedia, we review characteristic properties of event notions that have emerged in various areas in and around multimedia. A common model should be capable of representing events from all of these areas. However, as our review shows, a suitable common foundation for multimedia events has yet to be developed.

Multimedia presentation formats and programming frameworks

The simple event models offered by multimedia presentation formats such as SMIL and the Multimedia and Hypermedia Experts Group's MHEG-5 (see the sidebar) and programming frameworks such as the Java Media Framework and QuickTime are specialized on the specification of presentation dynamics. They cover discrete events tightly connected to instantaneous state changes in media presentations. They not only lack media independence but also genericity to serve a broader purpose. In addition, given their use for denoting deterministic presentation state changes, they don't consider uncertainty. Hence, multimedia presentation formats and programming framework event models don't offer suitable base models for the common representation of events from diverse multimedia application fields, such as multimedia content analysis, databases, or experiential computing.

Multimedia content analysis

Although event detection on various abstraction levels and for different domains is a central topic in content analysis, the focus has mostly been on the use of content features for detecting events within media and less on the modeling of the detected events or their use for detection of higher-level events. Thus, event models applied in multimedia content analysis, if made at all explicit, typically lack media independence as well as adaptability or extensibility to other domains (see the sidebar for a discussion of these models), and have limited expressiveness because they focus on one domain or abstraction level's needs.

The Video Event Representation Language (see the sidebar), for example, provides an event model for representing video events and an ontology language for inferring video events from other events. VERL is founded on first-order logic and thus defined in a formally sound manner. It is also generic, so users can define application-specific event properties and predicates.

However, VERL doesn't meet all of the requirements for a common multimedia event model. It represents only in-video events, using frames and pixels to express an event's time and location in relation to a piece of video content. VERL thus lacks media independence, prohibiting unified indexing of all media documenting real-world events regardless of modality.

Moreover, although users can express (mainly spatiotemporal) relationships between events in event inference rules using predicates, VERL doesn't offer an explicit association construct for representing relationships between events once they've been derived. This hinders creating user-specific views of events and their interrelationships. Finally, despite the uncertainty and ambiguity inherent in video analysis, VERL doesn't support capturing uncertainty and reasoning in its presence.

Multimedia databases

Typical multimedia databases store media as well as the metadata and content features describing the media. Thus multimedia databases mostly specialize in a given media type.[1] Only a few systems—such as IBM's S3 system, Smooth, or Snoek and Worring's work (as discussed in the sidebar)—consider events as explicit concepts and provide more or less expressive and generic event models that can be adapted to different applications' needs. But as with multimedia content analysis, even these event models lack media independence, because they regard events merely as additional pieces of metadata of the particular media type managed. They don't support the unified indexing of any documenting media available for a particular event.

Multimedia frameworks and metadata formats

The MPEG-7 metadata standard offers an expressive description scheme for event exchange and representation, but it still regards events as part of media metadata, thus lacking media independence. MPEG-7 lets users describe discrete and continuous events in detail, including time and location (on physical and logical levels). The model allows the addressing of concepts in external knowledge sources for the description of events and is thus knowledge aware. Because it's interoperable with the other MPEG-7 description schemes and descriptors, it's also highly media aware.

But the event description scheme also inherits MPEG-7's ambiguity, which it incurs, among other things, from its use of generic XML as its data model, its ability to introduce arbitrary XML schemas as new media description schemes, and its widespread use of optional and arbitrarily repeatable elements in the predefined media description schemes with often unclear semantics of those combinations. Hence, the event description scheme doesn't provide a sound formal basis for the common representation of events in multimedia. Also, MPEG-7 still regards events as part of audiovisual media metadata, thus lacking media independence.

As mentioned in the "Events in Multimedia" sidebar, events also play a role in the MPEG-21 multimedia framework. However, the MPEG-21 event model focuses on the specification of access rights to media content and isn't reasonably extensible or adaptable to other uses.

Experiential computing

Although experiential computing applications such as eChronicles, life logs, and event-centric media managers provide a high degree of media independence, the event models they define are ad hoc, targeted at suiting immediate application needs. The models can't be easily adapted to other applications. For example, the modeling of the temporal and spatial aspects of events is typically hardwired, and dedicated mechanisms for specifying application-specific extensions are lacking. Further characteristic deficiencies are lacking in formality, support for associations besides simple super- or subevent relationships, and consideration of uncertainty.

Appan and Sundaram[7] gave an interesting exception to the common lack of formality. They defined a formal event model for the event-oriented exploration of digital photos taken by different users. However, their model is tailored to the application and it's hard to apply elsewhere.

Only a little progress has been made toward generic infrastructural components for experiential computing. The MediÆther system[6] provides a distributed object space for propagating multimedia events. Pack et al.[3] and Kim, Gargi, and Jain[8] proposed event databases. However, the event models underlying these generic components still exhibit considerable deficiencies with regard to their suitability as frameworks for the common representation of events in multimedia. While being more generic, the models generally superimpose the representation of central aspects of event description, thereby reducing the models' adaptability to different applications' needs. This is especially true for the representation of events' temporal and spatial aspects. Moreover, they generally don't consider event uncertainty or ambiguity.

These models provide only limited support for explicit associations interlinking individual events; if such support is available, it focuses mostly on super- or subevent compositions.[3] In addition, the models are either not defined in a formally sound manner or aren't very expressive. Pack et al., for instance, specified their event model in terms of a coarse-grained, semistructured XML schema that's to be refined for concrete applications; Kim, Gargi, and Jain defined their model formally but maintain an unusual and simplistic event notion. They don't regard events as incidents of importance but as timestamped attribute/value pairs carrying results of sensor data or multimedia content analysis.

Event notification systems

Events are a long-established and integral notion in publish/subscribe middleware. Event notification systems typically offer generic event models with freely definable properties that users can extend and adapt to the needs of an application domain—theoretically, to multimedia as well. Interest in events and event notification is also appearing in the data stream management domain.[5]

Event models of event notification systems and data stream management systems focus mainly on discrete events that indicate state changes in distributed applications. They also commonly restrict event properties to simple data types such as string, float, or integer, and don't support complex data. This limits the models' potential for media awareness. Moreover, although event notification systems typically support sophisticated rule languages for inferring events from the occurrence of others,[9] event models focus on atomic events and are weak concerning the explicit representation of associations between individual events, such as composition or causality. Finally, they don't consider event uncertainty.

For the common representation of events in multimedia applications, event notification systems and data stream management systems therefore don't provide direct solutions. An interesting development is thus the emergence of more media-aware event notification systems. Liu et al.[4] have provided a continuous event querying and notification system on top of a multimedia data stream management system. Although one of this system's strengths is that it is media-aware and based on a formally sound definition, the event model applied isn't very expressive. The system considers events merely as flags for incidents of importance without much additional descriptions, and it doesn't consider uncertainty.

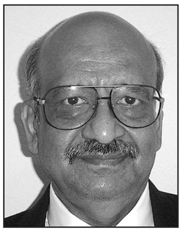
Conclusion

Our article has laid the initial grounds for developing a common and formally grounded event model that suits the needs of many applications by formulating and systematically deriving a catalog of basic requirements. We've begun with the design of a multimedia event model that comprehensively addresses these requirements.[10] The need remains for a joint community effort to further explore and achieve a broader acceptance of the idea of event-centric multimedia computing and to establish a widely accepted standard model for events in multimedia. **MM**

References

1. R. King, N. Popitsch, and U. Westermann, "METIS: A Flexible Foundation for the Unified Management of Multimedia Assets," *Multimedia Tools and Applications*, 2007, to appear.
2. J. Hunter, "Reconciling MPEG-7 and MPEG-21 Semantics through a Common Event-Aware Metadata Model," *Computing Research Repository (CoRR)*, Oct. 2002.
3. D. Pack et al., "An Event Model and Its Implementation for Multimedia Representation and Retrieval," *Proc. 2004 IEEE Int'l Conf. Multimedia and Expo* (ICME), IEEE Press, 2004, pp. 1611-1614.
4. B. Liu, A. Gupta, and R. Jain, "MedSMan: A Streaming Data Management System over Life Multimedia," *Proc. 13th ACM Int'l Conf. Multimedia* (Multimedia), ACM Press, 2005, pp. 171-180.
5. A. Demers et al., *A General Algebra and Implementation for Monitoring Event Streams*, tech. report TR2005-1997, Cornell Univ. Library, 2005.
6. S. Boll and U. Westermann, "MediÆther—An Event Space for Context-Aware Multimedia Experiences," *Proc. 2003 ACM SIGMM Workshop Experiential Telepresence* (ETP), ACM Press, 2003, pp. 21-30.
7. P. Appan and H. Sundaram, "Networked Multimedia Event Exploration," *Proc. 12th Ann. ACM Int'l Conf. Multimedia* (Multimedia), ACM Press, 2004, pp. 40-47.
8. P. Kim, U. Gargi, and R. Jain, "Event-Based Multimedia Chronicling Systems," *Proc. 2nd ACM Workshop Continuous Archival and Retrieval of Personal Experiences* (CARPE), ACM Press, 2005, pp. 1-12.
9. P.R. Pietzuch, B. Shand, and J. Bacon, "A Framework for Event Composition in Distributed Systems," *Proc. ACM/IFIP/Usenix Int'l Middleware*

Conf. (Middleware 2003), Springer Verlag, 2003, pp. 62-82.
10. U. Westermann and R. Jain, "Events in Multimedia Electronic Chronicles (E-Chronicles)," *Int'l J. Semantic Web and Information Systems,* vol. 2, no. 2, pp. 1-23.



Ramesh Jain is the Bren Professor in Information & Computer Sciences at the University of California, Irvine. He is an active researcher in multimedia information systems, image databases, machine vision, and intelligent systems. He recently cofounded Seraja Technologies Pvt. Ltd. to address the needs of the emerging EventWeb. He was the founding editor in chief of *IEEE MultiMedia* and is a fellow of the ACM, IEEE, AAAI, IAPR, and SPIE.

Utz Westermann recently joined Seraja Technologies, where he leads the design and development of an event-centric multimedia storage back end. He was a postdoctoral scholar at the University of California, Irvine, at the time of writing of this article. His research interests include multimedia content management, multimedia databases, and mobile multimedia. Westermann has a doctoral degree in technical sciences from the Technical University of Vienna, Austria.

Contact Ramesh Jain at jain@ics.uci.edu.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.