E – A Generic Event Model for Event-Centric Multimedia Data Management in eChronicle Applications

Utz Westermann, Ramesh Jain
Computer Science Department
University of California
Irvine CA 92697, USA
{utz.westermann, rcjain}@uci.edu

Abstract

eChronicle applications are inherently event-centric, enabling users to find and explore important events in an application domain and providing unified access to any media that document them. Today's multimedia data management components such as multimedia databases, however, are largely media-centric, considering events – if at all – just as one of many pieces of media metadata. Obfuscating event exploration and event-driven access to media, they are only of limited use for the implementation of eChronicle applications. Using a concrete eChronicle application in the defense domain, this paper motivates the need for event-centric multimedia data management components. As a foundation, the paper proposes the E *multimedia event model and discusses essential design considerations for the development of that model.* E*'s genericity and profound adaptability to varying application needs make the model a suitable foundation for reusable multimedia data management components that are useful not only for eChronicle applications, but for any multimedia application where event-driven access to content is of interest.*

1. Introduction

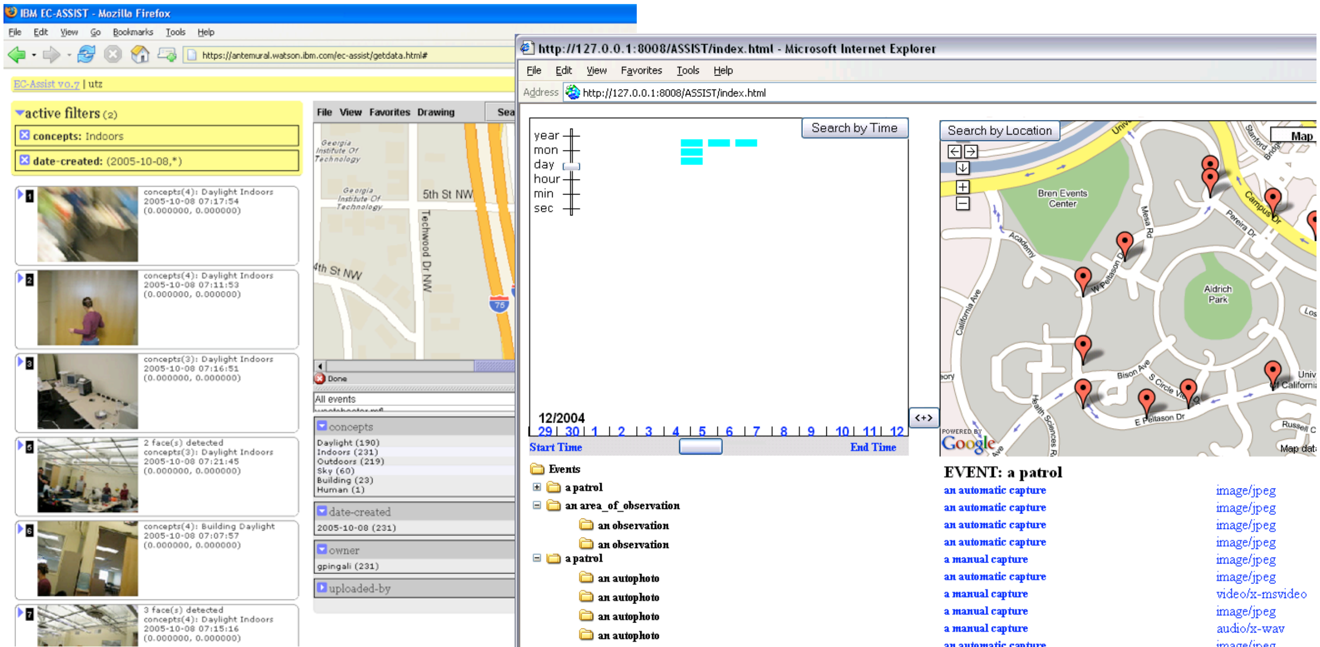
Emerging multimedia applications such as eChronicles [15] and life logs [10] aim at providing ways of exploring the course of real-world events and unified access to any media about those. In the DARPA project "Electronic Chronicling and Group Wear for Advanced Soldier Information Systems and Technology" (EC-ASSIST) – on which we work with partners from IBM T. J. Watson Research Center, Georgia Tech, MIT, and AWare Tech – we are developing a multimedia eChronicle that helps soldiers going out on reconnaissance missions with wearable sensors and media production devices analyzing and reporting the important events of their missions after their return.

However, current infrastructure components for multimedia data management such as multimedia databases and multimedia retrieval systems are media-centric. They focus on media and their descriptive metadata, often even limiting themselves to a single media type such as image or video [19]. The particular events documented by media – if at all considered – form just one part of media metadata; there is no dedicated support for exploring the course of events and uniformly accessing any kind of documenting media. In order to obtain adequate implementation platforms for eChronicle applications, there is the need for event-centric multimedia data management components that treat events as first-class citizens just like eChronicles do.

To this end, we make several essential contributions in this paper: using EC-ASSIST, we illustrate the inadequacy of traditional media-centric multimedia data management components for the implementation of eChronicle applications, highlighting the potential benefits of event-centric data management components in this regard. We then discuss essential design considerations for a multimedia event model that can serve as a common foundation of event-centric multimedia data management components, ranging from common multimedia event query and rule languages to reusable multimedia event exploration environments, databases, query and inference engines based on these languages. Finally, we propose E, a generic multimedia event model developed along these considerations.

Due to its genericity and adaptability, E and multimedia data management components based on E are applicable to not only EC-ASSIST and other eChronicle applications, but generally to any multimedia application in which event-driven organization and access of media is of interest, for example news media databases.

The paper is organized as follows: Section 2 motivates the need for event-centric multimedia data management components. Section 3 presents important design consider-



(a) (b)

Figure 1. Media-centric (a) vs. event-centric (b) media management

ations for a multimedia event model for event-centric data management components. Section 4 with E presents and semi-formally defines such an event model. Section 5 examines related work. Section 6 provides a conclusion and outlines current and future work.

2. Event-Centric Media Management

Using the after-mission analysis and reporting eChronicle being developed in EC-ASSIST as an illustration, we now elaborate on why media-centric multimedia data management components are not much of a help for the implementation of eChronicle applications and why event-centric data management components are needed instead.

In fact, the first prototype of the EC-ASSIST after-mission analysis eChronicle – developed by our partners from IBM – has been built on top of a traditional media-centric multimedia database. In the database, the various media (including videos, photos, and audio recordings) captured both manually and automatically during missions are indexed by context data (including time and GPS position of their capture), by content categories such as indoors, gunshot, or car determined by content analysis, and by other metadata such as the results of speech transcription and face and number plate detection.

While this first prototype was positively received by soldiers in a recent field trial, we are aware of inherent limitations incurred by the use of a media-centric multimedia database as the storage backend. For illustration, Figure 1 contrasts the prototype's current user interface (a) with an experimental alternative user interface we have implemented on top of an E-based event store (b), both of which directly mirror the corresponding approach to multimedia data management of their respective backend. The interface of the prototype depicted in Figure 1 (a) permits the filtering of the *media* collected on a mission straight-forwardly along the various context data, content categories, and other metadata in the media-centric multimedia database; a map view is available for filtering along location. The experimental interface on top of the event store, in contrast, permits the filtering of the *events* that occurred during missions along time, location, and structure; the media collected on a mission are accessible via the events they document.

Note that it is not our intention here to compare and assess the particular designs of these user interfaces, but rather to offer the reader visual cues to follow the ensuing discussion of the implications and opportunities involved with the use of media- or event-centric multimedia data management components for eChronicle applications like EC-ASSIST.

A major problem of using media-centric multimedia data management components for the implementation of eChronicle applications is the *impedance mismatch* between the primary objects of interest to data management and the primary objects of interest to users. Media-centric data management components faciliate querying along media and their metadata, and the result of those queries are again media. When performing after-mission analysis in EC-ASSIST, however, soldiers primarily want to get hold of important events that occurred on their missions such as observations, encounters, or insurgents arrested and their characteristics. Media are only one source of information among others from which insights about such events can be obtained. Unlike event-centric multimedia data management components, media-centric components typically do not provide immediate support for the querying of events.

In event-centric multimedia data management components, events provide a *unified index* onto any media and sensor data that document an event regardless of type. This has two benefits: events provide a *multimodal abstraction* from raw media and sensor data, which promotes the application of multimodal content and sensor analysis for event detection and inference. In the context of the eChronicle for after-mission analysis in EC-ASSIST this is of particular relevance. The recent field trial has shown, for example, that audio recordings during missions were heavily distorted, leaving the detection of events like soldiers greeting people by means of speech transcription alone very unreliable; but the situation can be improved by also taking sensor data from accelerometers on the soldiers' bodies into account to detect handshakes. In contrast, media-centric multimedia data management components per se consider any media and sensor data about an event in insolation.

The second benefit is *scalability*. When analyzing data from hundreds of missions and selecting an area on a map view, media-centric multimedia data management components will interpret the query such that any scrap of media captured during a mission in that area is to be returned. But the soldier is more likely to share the interpretation of event-centric multimedia data management components to deliver the important events that occurred in the area, abstracting from the potentially many media that document them.

Moreover, events are a single concept that can be seamlessly applied to any *level of abstraction*. If based on event-centric multimedia database components, the EC-ASSIST after-mission analysis eChronicle can not only capture low-level media creation and content analysis events such as "manual photo" or "gunshot", but also hierarchically compose them (e.g., via spatio-temporal clustering) into higher-level domain events like "observation" or "encounter" close to the user's thinking. Although media and their metadata stored in media-centric multimedia data management components could be seen as media creation and content analysis events, composite event cannot be represented without adding data modeling constructs such as media folders or complex media objects [19].

Treating events as first-class citizens independent of media, event-centric multimedia data management components can handle *multiple views* onto the same media gracefully. This is useful for the EC-ASSIST eChronicle, since two soldiers may have different opinions about the course of a mission. Also, in addition to composing the events of a single patrol into one hierarchical structure, it may be just as valid to group events across patrols, for instance, to identify "area of observation" events out of spatially close clusters of "observation" events from different missions. By storing interpretations of content in media metadata, in contrast, media-centric multimedia data management components tend to superimpose global views onto content.

For mission analysis in EC-ASSIST, not only events are of user interest but also the *associations* between events, like structural, causality, temporal, and spatial relationships. Due to their focus on media, however, media-centric multimedia data management components typically (if at all) consider only associations between media. This makes the representation of inter-event associations in the after-mission analysis eChronicle difficult.

Finally, events are also central concept in distributed system design. Basing the after-mission analysis eChronicle on top of event-centric multimedia data management components suits the distributed nature of media and sensor data production in EC-ASSIST. It also opens up the system to future extensions such as the analysis of live events from ongoing missions or the provision of a mobile notification system that informs soldiers on patrol about important events in other missions close by.

These lines of argumentation in favor of event-centric multimedia data management components can be analogously extended not only to other eChronicle applications, but also to classic areas of multimedia data management: when users search for media, they are very often on the lookout for information about events. This is very obvious in the case of news and sports media retrieval [30, 9]; less obvious but indicated by studies is that people searching or browsing their personal media collections also mainly orient themselves along events [28].

3. A Multimedia Event Model: Design Considerations

As a foundation for the development of event-centric multimedia data management components that are useful for eChronicle applications like the EC-ASSIST after-mission analysis tool, the development of a common multimedia event model is highly desirable. A common model opens up the way towards reusable and interoperable components like common event stores, middleware for event propagation and notification, event detectors, and event inference engines; common formalisms and languages for multimedia event processing including event query languages and rule languages for the detection of new events;

and reusable user interface components for event querying and exploration.

A common multimedia event model may even constitute the germ of an Event Web [14]: a world-wide infrastructure of interlinked events and related media as opposed to the network of interlinked hypermedia documents provided by the World-Wide Web.

When designing an event model that is supposed to meet these expectations, several aspects need to be considered:

- The event model should be defined formally such that it facilitates the development of event query and rule languages with sound semantics that allow optimized query and rule processing. However, we leave the issue of formality aside for the rest of the paper. For the development of E, we have opted for an agile two step approach: the model has first been prototyped via object-oriented design in order to be able to apply the model and react to necessary changes quickly – the paper presents this state of design. The transcription of the semi-formal model to its final formal form is current work in progress.

- The model should allow globally unique and uniform identification of events, such that it is possible to address events from different, even distributed sources.

- A suitable multimedia event model should be expressive. It should provide rich means to capture elementary aspects of event description: the *temporal aspect* – i.e., the time when an event occurred – the *spatial aspect* – i.e., the place where an event occurred – the *informational aspect* – i.e., description of the kind of event occurring and the involved entities – the *experiential aspect* – i.e., media or other sensor data providing documentation on the course of an event – the *structural aspect* – i.e., sub-events occurring during the course of an event – and the *causal aspect* – i.e., the events that led to a given event.

- In order to cover the experiential aspects of events and to provide unified media indexing, the event model should offer a high degree of media-awareness. Events should be capable of referring to a variety of different documenting media or sensor data from different sources. The representation of elementary metadata and features of the media and sensor data referenced should also be possible to support the selection of media documenting events or similarity-based retrieval.

- Nevertheless, a multimedia event model should still maintain media-independence. Since events may be documented by many different media, they do not form a descriptive property of a single medium alone and cannot reasonably be made part of its metadata; events exist independent of documenting media.

- To open access paths from events to the entities of the domain and information about those entities, the model should show knowledge-awareness. It should allow references to external information in various knowledge sources, ranging from ontologies or databases to address books or calendars.

- A suitable event model should offer explicit support for the representation of associations to be able to capture important structural, causal, or spatio-temporal relationships between events.

- Current media and sensor data analysis methods for the automatic detection of events, event properties, and associations between events are unreliable and imprecise. Thus, a suitable multimedia event model – and indirectly, any query and rule languages on top of it – must offer intrinsic support for uncertainty, not only with regard to events but also to event properties and associations.

- Different applications need to describe the different aspects of events in different fashions. The events in the EC-ASSIST eChronicle application are different from the events in a research eChronicle [11]. The event model should be adaptable and extensible to suit the needs of many different applications.

A popular class of event models – often encountered in current eChronicle applications (e.g., [1, 11, 16]) and in distributed event notification and data stream management systems (e.g., [4, 6, 8]) – uses simple tuples to represent events. Why not just follow this wide-spread modeling approach for the development of a common multimedia event model and "be done with it"?

Figure 2 provides a sketch of a typical tuple-oriented event model design. In the sketch, an event tuple addresses the event's informational aspect by elements providing an event id, the event type, and a set of additional generic attribute-value pairs that further describe the what of an event. Its temporal aspect is captured by a timestamp, its spatial aspect by a GPS position.[1] The structural aspect is represented by a set of ids of subevents that occurred during the event. Similarly, the causal aspect is represented by a set of ids of the events that caused the present event. Finally, the experiential aspect is captured by a set of network addresses in form of URLs that refer to sources of documenting media and sensor data.

While straightforward at first glance, this approach to event model design does not take the above design considerations sufficiently into account and yield a broadly applicable foundation for event-centric multimedia data man-

[1] We could just as well use time intervals and geographic areas without changing the general validity of the ensuing line of argumention.

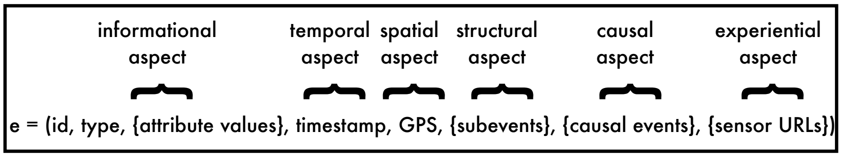


Figure 2. Simple tuple-based event model

agement components. Firstly, uncertainty is not considered. Although it would be possible to add an uncertainty attribute-value pair to the informational aspect, uncertainty may not just affect a complete event but also any of its properties. This is hard to capture with a simple tuple model.

Secondly, the hard-wired representation of an event's temporal and spatial aspects in form of timestamps and GPS positions limits the model's applicability. While timestamps and GPS positions are perfect for the modeling of the time and place of discrete, usually low-level technical events (e.g., "gunshot" in the EC-ASSIST scenario), they are not adequate to represent continuous events with temporal and spatial extensions (e.g., "patrol" in EC-ASSIST). In other applications, it may even be inadequate to use time intervals or geographic areas: for example, one might want to express the temporal aspect of an event by referring to an abstract temporal concept for which no reasonable exact time interval can be given (e.g., "Christmas") or the spatial aspect via a spatial association to another event, because the position of an event is only known relatively to another.

Thirdly, using references to capture sub-events and causal events is not an appropriate substitute for explicit structural and causal associations. This imposes a single structure and causality onto an event as a global fact. But as already explained: soldiers in the EC-ASSIST eChronicle application, for instance, may have different views onto the course of a patrol and the reasons for that course.

Fourthly, the experiential aspect of an event is not captured sufficiently in the sketched model, since media metadata is not considered.

Recognizing these problems of tuple-oriented event model designs, we have chosen to follow a different route for the development of E, as outlined in the next section.

4. A Multimedia Event Model: E

Following the design considerations set forth in the previous section, we have developed the E multimedia event model as a generic foundation for event-centric multimedia data management components. The basic principles behind E are:

- *Cover the primary objects of interest:* These not only include events but also media and other sensor data that document the course of the events as well as concepts to which the events are related. This establishes E's media- and knowledge-awareness.
- *Address the essential aspects of event description:* The model takes explicit account of the temporal, spatial, informational, experiential, structural, and causal aspects for the description of events.
- *Provide rich event descriptors:* Events can be described by attribute values of arbitrary complexity, simple tags, references to concepts, media, and other sensor data, and may participate in associations. Thus, E is highly expressive.
- *Permit the characterization of any event aspect with any kind of descriptor:* Departing from any other event model we know of, E does not prescribe which descriptors to use for an event's various aspects. For example, the spatial aspect of an event could be captured via an attribute value representing a geographic point, a more complex attribute value representing a geographic area, a spatial association to another event, a geographic concept denoting a well-known landmark, or even a stream of sensor data originating from a GPS receiver. This results in a high degree of adaptability to application needs and makes E broadly applicable.
- *Support uncertainty:* In E, events as well as their descriptors can be attributed with confidence values, covering uncertainty not only on the event but also on the level of event properties.
- *Offer genericity and extensibility:* E does not predefine any types of events, sensor data, and concepts just as it does not predefine any attribute, tag, concept reference or association types. These can be defined as needed for an application. A consequence of this generic extensibility is the need for an event schema language for the definition of the various types allowed by an application. This schema language is still work in progress.

Figure 3 gives an overview of the basic structure of the E model. The various constituents of the model share a common set of characteristics. Each constituent is globally and uniformly identifiable a 128 bit universally unique

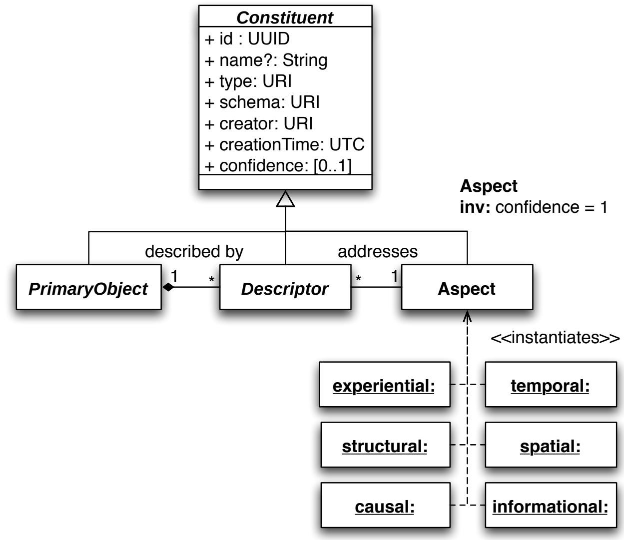


Figure 3. E event model overview (UML class diagram)

identifier (UUID). These can be created in large numbers in a distributed fashion without coordination overhead, which is beneficial for the use of E for distributed event-centric multimedia data management components. A constituent's id can further be augmented by a human-readable name.

The semantics of a constituent are denoted by its type, which is given by a uniform resource identifier (URI). E leaves the definition of suitable types to applications. Focusing on the modeling of events, however, E does not cover such type definitions and defers that to an event schema language yet to be developed. The type of each constituent is further augmented by the schema which it is part of, again identified by a URI. This permits a future event schema processor to look up the declaration of a constituent type and establishes a namespace mechanism.

Moreover, E records for each constituent the creator and a timestamp of its technical creation. Denoted by a URI, the creator of a constituent can be anything from a software component or sensor to a user. Along with the creation time, this can, for instance, be useful for synchronization. Note that the creation time of an event is independent of its temporal aspect: the technical creation of an event can happen before the event takes place (e.g., when a future event is announced) just as well as after the event has occurred (e.g., when an event is reported after it has taken place).

Finally, each constituent has a confidence value from $[0..1]$, allowing for uncertainty at every level of the model.

E distinguishes three major categories of constituents: the primary objects of interest including events, concepts, media and sensor data; descriptors such as attribute values or tags that can be attached to these primary objects of interest; and the descriptive aspects to which the descriptors refer. Predefined are the temporal, spatial, informational, experiential, structural, and causal aspects; if reasonable for an application, however, new aspects can be introduced. As factual constituents that are either relevant or not, E requires all aspects to have a confidence of 1.

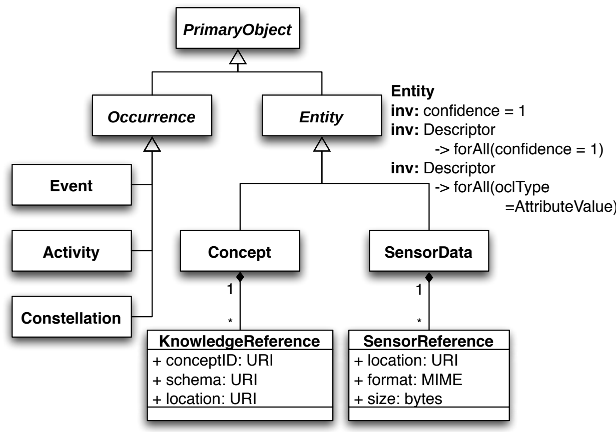


Figure 4. Primary objects of E (UML class diagram)

Figure 4 shows the primary objects in the E multimedia event model. E distinguishes three different kinds of event-like occurrences of importance: events, activities, and constellations. The difference between events and activitites is that events are telic – i.e., the occurrence signified by an event is cumulating and not yet valid for any fraction of the occurrence – whereas activities are atelic – i.e., the occurrence signified by an activity is also valid for any fraction. "patrol" is an event while "enemy resistance" is an activity.[2]

By means of constellations, E can explicitly represent associations between two or more occurrences: events, activities, but also other constellations. The reason why E models constellations as occurrences is that a discovered relationship between occurrences may be an important occurrence by itself: in the EC-ASSIST eChronicle application, for example, the constellation that a "gunshot" event occurred during the activity "search" can be an important occurrence for after-mission analysis.

Concepts and sensor data form the second group of E's primary objects of interest, subsumed as entities. Entities are factual: they either exist or do not. Entities thus always have a confidence of 1, just as any descriptor attached to them. Concepts model abstract entities of a domain to which events and other occurrences are related. E further

[2] This distinction largely depends on the observers viewpoint: a patrol can be equally well regarded as an event of type "patrol" or an activity of type "patrolling". Nevertheless, E permits this distinction if useful.

allows references to external information sources about a concept to be attached, such as databases, calendars, or RDF descriptions. Each of those references uses URIs to encode the id of the concept in the external source, the schema of the external source, and network access information. This makes E knowledge-aware.

By means of sensor data entities, E captures media and other sensor data that document the course of events or other occurrences. E further separates between sensor data on an abstract level and references to sensor sources that provide the data on a physical level: a single medium may be available in different storage locations such as web servers or databases. References to sensor sources encompass network access information, encoding format, and size (if available). Sensor data entities contribute to the media-awareness of E.

Like other constituents, E's primary objects have a type property, allowing the distinction of any classes of events, activities, and constellations as well as any categories of concepts and sensor data required by an application.

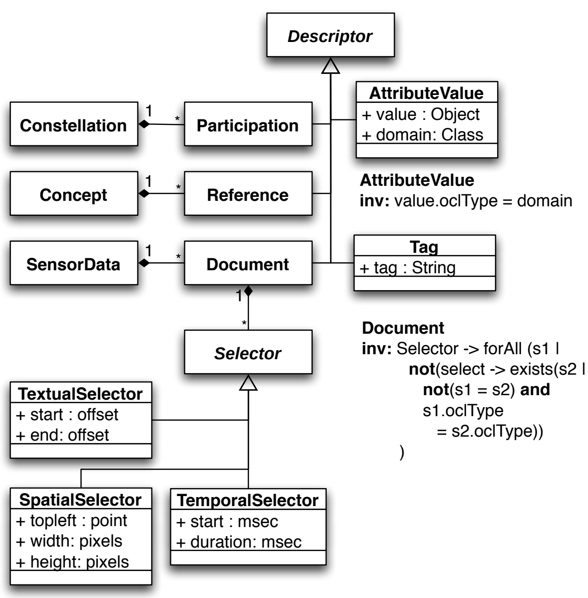


Figure 5. Object descriptors of E (UML class diagram)

Figure 5 depicts the various descriptors that E offers for the description of the primary objects of interest. Descriptors are second-class objects; their existence depends on the primary objects they describe. Any descriptor refers to exactly on aspect of description; however, E does not impose any limitations on which descriptor to use for which aspect, allowing the model to suit very different application needs.

Attribute value descriptors allow one to attach typed attribute-value pairs to a primary object. The attribute for which a value is provided is identified by the type property common to all constituents; the value itself is represented by an object of a class that suits the domain of the attribute. Thus, attribute values can range from rather primitive values – e.g., a timestamp describing the temporal aspect of an event – to complex values – e.g., a color histogram providing some metadata for a piece sensor data.

Attribute values are the only descriptors that are not just applicable to occurrences but also to entities. This restriction may seem somewhat arbitrary. But although E should be knowledge- and media-aware, we do not want to recreate yet another full-fledged knowledge representation model that distinguishes itself from similar models such as RDF or Topic Maps only by incorporating some explicit event concept. Instead, our focus is purely on the representation of events for multimedia data management: we thus have events pointing to concepts and media in external knowledge and media sources, relying on existing technology.

While it would thus be consequent to not allow the description of entities with descriptors at all, we opted to make a pragmatic exception for attribute values: it may be useful for applications to have access to basic metadata and features directly from within the E model (e.g., when choosing between sensor data documenting an event) or to frequently accessed basic information about concepts.

Document descriptors allow any occurrence – event, activity, or constellation – to refer to sensor data that document the occurrence. The kind or role of the documentation provided is given by the document descriptor's type property. It is possible to only refer to parts of sensor data by the use of selectors. Current selectors permit the selection of parts by means of time intervals in temporal data, rectangular spatial regions in visual data, regions in textual data, or a combination of these. More selectors can be added to the model. For reasons of simplicity, it is only allowed to attach one selector of each kind to a document descriptor.

Reference descriptors interlink occurrences with concepts to which they are related. The role the concept plays for the occurrence is given by the reference's type property. Quite similar to references is the tag descriptor. But instead of linking to a formal, predefined concept, the concept referred to by a tag is given by a free text string. This enables "folksonomy" applications where events are being relaxedly annotated with free tags in a collaborative effort.

Finally, the participation desciptor interlinks an occurrence with a constellation in which it is involved. the role the occurrence plays in that constellation is given by the participation's type property.

Figure 6 exemplifies how the different constituents of E could play together for the representation of reconnaissance mission events. The example basically describes a "gunshot" event that occurred as part of a search activity. For reasons of space, the example is tailored to show each composition:Constellation

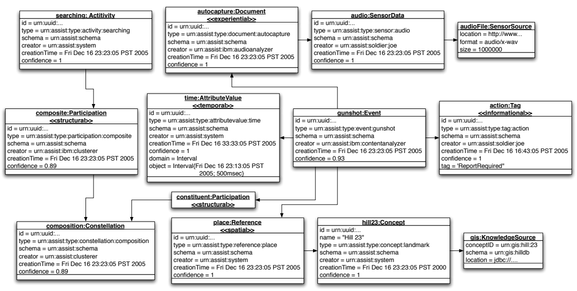


Figure 6. E event model example (UML object diagram)

model constituent, with the exception of aspects, once; the event schema applied thus diverges from the one we used for the experimential eChronicle interface of Figure 1 (b).

5. Related Work

Having presented the E multimedia event model, we now examine models from related areas. No model we know of provides an equally suitable foundation for event-centric multimedia data management components.

Events are a long standing concept in multimedia document models like SMIL [3] and MHEG-5 [13] and programming frameworks like QuickTime [2] and the Java Media Framework [29]. But the event models encountered in this domain have a very limited focus on the representation of presentation events with regard to temporal synchronization and user interaction and cannot be reasonably generalized for event-centric multimedia data management components.

Events also appear in multimedia databases and information retrieval. Specialized, however, on single media types and/or single application domains [19], the event models in this area are mainly tailored to the representation of low-level technical media events (e.g., scene changes or constellations of objects in videos [7, 23]) or application-specific domain-level events (e.g., sports or news [30, 9]), hardly offering foundations for reusable event-centric multimedia data management components. The few more generic event models – for instance, as provided by the video database SMOOTH [20] – lack media-independence spoiling unified indexing of media of arbitrary type.

Events further play a traditional role in middleware, in particular in event notification systems [26, 6, 4] and of late also in some data stream management systems [8]. The models in this domain capture events mostly as simple tuples with all the problems described in Section 3: failure of providing explict support for uncertainty, lack of dedicated associations hindering, for instance, adequate representation of event structure and causality, and superimposed representations of an event's temporal or spatial aspect (e.g., [8]) preventing a broader applicability for event-centric multimedia data management components.

Many of the event models used by the various eChronicle applications (such as for the exploration of events in sports [27], meetings [16, 12], research [11], and business processes [18]), life log applications [10], and event-oriented media managers [1, 21] that have been proposed recently use simple tuple models for the representation of events as well and suffer from similar problems. These models are also tailored to their specific domains, lacking extensibility and adaptability to serve as foundations for infrastructural event-centric multimedia data management components.

Only a few generic components for event-centric multimedia data management have been proposed in the literature. MedSMan [22] is a generic media stream querying and event detection system, the Multimedia Event Infrastructure (MEP) [24] and MediÆther [5] offer distributed systems for the propagation of and notification about multimedia events, and Pack et al. [25] and Kim et al. [17] propose generic multimedia event storage infrastructures. While offering broader applicability than the application-specific models of the eChronicle applications mentioned before, most of the event models in these approaches again constitute simple tuple models with all incurring problems. Kim et al. [17] even have a very unconventional notion of events: in their work, an event is not an occurrence of importance but merely a timestamped attribute-value pair produced a result of media or sensor stream processing.

6. Conclusion

In this paper, we have argued that next-generation eChronicle and life log applications are in need of event-centric multimedia data management components and, using the EC-ASSIST project as an illustration, outlined the problems with today's media-centric data management tools and infrastructures. We have discussed elementary design considerations for a suitable multimedia event model at the heart of an event-centric media management. We have then semi-formally presented E, a generic multimedia event model that pays attention to these considerations.

Based on E, we have implemented a generic file-based event store as a first event-centric multimedia data management component using VisualWorks Smalltalk. The event store offers a navigational object-oriented API structured along the class diagrams of Section 4. Via this API, applications can access the events in the store and navigate the relationships between them.

On top of the E event store, we have implemented the experimental event-centric web user interface for after-mission analysis in EC-ASSIST depicted by Figure 1 (b). The navigational API of the event store kept the implementation of the demonstrator's event query logic relatively simple; due to the current lack of an event schema language for E , however, the code of demonstrator is burdened with additional validation code that ensures the integrity of the EC-ASSIST event data in the store.

The provision of E opens up many interesting research directions. Inter alia, we are working on the following:

- We are formalizing E.

- Based on the formal definition of E, we are developing an adequate event query algebra and a rule language for the inference of events out of the occurrence of other events. A unique characteristic of both algebra and rule language is going to be support for uncertainty.

- We are working on an event schema language for E that permits the modular definition of the various types of descriptors that are applicable to the different types of events, activities, constellations, concepts, and sensor data of a domain.

- We are developing a distributed publish/subscribe infrastructure that provides a middleware for integrating distributed event detectors, event inference engines, and event stores with event-driven multimedia applications on the basis of E.

- We are developing an event presentation and exploration environment based on E.

- We are applying E and the developed tools and techniques not only to reconnaissance mission analysis in the EC-ASSIST project but also to other domains such as situational awareness for emergency response and video observation systems.

We are convinced that event-centricity is the next leap in multimedia data management. With the E event model, we are providing an essential foundation in that direction.

Acknowledgements

We would like to thank Bo Gong and Srikanth Agaram for their efforts in implementing and setting up the experimental event-centric user interface.

References

[1] P. Appan and H. Sundaram. Networked Multimedia Event Exploration. In *Proc. of the 12th ACM International Conference on Multimedia (MULTIMEDIA04)*, New York, New York, Oct. 2004.

[2] Apple Computer Inc. *QuickTime API Reference*. Apple Computer Inc., Cupertino, California, 2005.

[3] J. Ayars, D. Bulterman, A. Cohen, et al. Synchronized Multimedia Integration Language (SMIL 2.0). W3C Recommendation, World Wide Web Consortium (W3C), Aug. 2001.

[4] J. Bacon, K. Moody, J. Bates, et al. Generic Support for Distributed Applications. *IEEE Computer*, 33(3), 2001.

[5] S. Boll and U. Westermann. MediÆther an Event Space for Context-Aware Multimedia Experiences. In *Proc. of the 2003 ACM SIGMM Workshop on Experiential Telepresence (ETP03)*, Berkeley, CA, Nov. 2003.

[6] A. Carzaniga, D. Rosenblum, and A. Wolf. Design and Evaluation of a Wide-Area Event Notification Service. *ACM Transactions on Computer Systems*, 19(3), 2001.

[7] D. DeMenthon and D. Doermann. Video Retrieval Using Spatio-Temporal Descriptors. In *Proc. of the 11th ACM International Conference on Multimedia (MULTIMEDIA03)*, Berkeley, CA, Nov. 2003.

[8] A. Demers, J. Gehrke, M. Hong, et al. A General Algebra and Implementation for Monitoring Event Streams. Technical Report TR2005-1997, Cornell University Library, 2005.

[9] P. Duygulu, J. Pan, and D. Forsyth. Towards Auto-Documentary: Tracking the Evolution of News Stories. In *Proc. of the 12th ACM International Conference on Multimedia (MULTIMEDIA04)*, New York, New York, Oct. 2004.

[10] J. Gemmell, R. Lueder, and G. Bell. The MyLifeBits Lifetime Store. In *Proc. of the 2003 ACM SIGMM Workshop on Experiential Telepresence (ETP03)*, Berkeley, CA, Nov. 2003.

[11] B. Gong, R. Singh, and R. Jain. ResearchExplorer: Gaining Insights through Exploration in Multimedia Scientific Data. In *Proc. of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR04)*, New York, New York, Oct. 2004.

[12] W. He, P. Xiang, and Y. Shi. A Semantic Framework for Meeting Data Retrieval. In *Proc. of the 2nd ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE '05)*, Singapore, Singapore, Nov. 2005.

[13] ISO/IEC JTC1/SC29. Information Technology - Coding of Multimedia and Hypermedia Information - Part 5: Support for Base-Level Interactive Applications. International Standard 13522-5:1997, International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), 1997.

[14] R. Jain. Experiential Computing. *Communications of the ACM*, 46(7), 2003.

[15] R. Jain. Multimedia Electronic Chronicles. *IEEE MultiMedia*, 10(3), 2003.

[16] R. Jain, P. Kim, and Z. Li. Experiential Meeting System. In *Proc. of the 2003 ACM SIGMM Workshop on Experiential Telepresence (ETP03)*, Berkeley, CA, Nov. 2003.

[17] P. Kim, U. Gargi, and R. Jain. Event-Based Multimedia Chronicling System. In *Proc. of the 2nd ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE '05)*, Singapore, Singapore, Nov. 2005.

[18] P. Kim, M. Podlaseck, and G. Pingali. Personal Chronicling Tools for Enhancing Information Archival and Collaboration in Enterprises. In *Proc. of the First ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE '04)*, New York, New York, Oct. 2004.

[19] R. King, N. Popitsch, and U. Westermann. METIS: A Flexible Foundation for the Unified Management of Multimedia Assets. *Multimedia Tools and Applications*, to appear 2006.

[20] H. Kosch, R. Tusch, L. Böszörményi, et al. SMOOTH a Distributed Multimedia Database System. In *Proc. of the 27th International Conference on Very Large Data Bases (VLDB 2001)*, Rome, Italy, 2001.

[21] J. Lim and Q. Tian. Home Photo Content Modeling for Personalized Event-Based Retrieval. *IEEE Multimedia*, 10(4), 2003.

[22] B. Liu, A. Gupta, and R. Jain. MedSMan: A Streaming Data Management System over Life Multimedia. In *Proc. of the 13th ACM International Conference on Multimedia (MULTIMEDIA05)*, Singapore, Singapore, Nov. 2005.

[23] R. Nevatia, T. Zhao, and S. Hongeng. Hierarchical Language-based Representation of Events in Video Streams. In *Proc. of the 2nd IEEE Workshop on Event Mining: Detection and Recognition of Events in Video*, Madison, Wisconsin, June 2003.

[24] D. Olsson and A. Nilsson. MEP - A Media Event Platform. In *Proc. of the 34th Hawaii International Conference on System Sciences (HICSS-34)*, Maui, Hawaii, Jan. 2001.

[25] D. Pack, R. Singh, S. Brennan, and R. Jain. An Event Model and its Implementation For Multimedia Representation and Retrieval. In *Proc. of the 2004 IEEE International Conference on Multimedia and Expo (ICME04)*, Taipei, Taiwan, June 2004.

[26] P. Pietzuch, B. Shand, and J. Bacon. A Framework for Event Composition in Distributed Systems. In *Proc. of the 4th ACM/IFIP/USENIX International Conference on Middleware (Middleware'03)*, Rio de Janeiro, Brazil, June 2003.

[27] G. Pingali, A. Opalach, Y. Jean, and I. Carlbom. Instantly Indexed Multimedia Databases of Real World Events. *IEEE Transactions on Multimedia*, 4(2), 2002.

[28] K. Rodden and K. Wood. How do People Manage their Digital Photographs? In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI03)*, Ft. Lauderdale, Florida, Apr. 2003.

[29] Sun Microsystems Inc. *Java Media Framework API Guide – JMF 2.0*. Sun Microsystems Inc., Santa Clara, California, 1999.

[30] H. Xu and T.-S. Chua. The Fusion of Audio-Visual Features and External Knowledge for Event Detection in Team Sports Video. In *Proc. of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR04)*, New York, New York, Oct. 2004.