Event-centric Media Management

Ansgar Scherp[a], Srikanth Agaram[b], Ramesh Jain[b]
[a]OFFIS - Institute for Information Technology, 26121 Oldenburg, Germany;
[b]University of California, Irvine, CA USA 92697-3435

ABSTRACT

The management of the vast amount of media assets captured at every day events such as meetings, birthday parties, vacation, and conferences has become an increasingly challenging problem. Today, most media management applications are media-centric. This means, they put the captured media assets into the center of the management. However, in recent years it has been proposed that *events* are a much better abstraction of human experience and thus provide a more appropriate means for managing media assets. Consequently, approaches that include events into their media management solution have been explored. However, they typically consider events only as some more metadata that can be extracted from the media assets. In addition, today's applications and approaches concentrate on particular problems such as event detection, tagging, sharing, classification, or clustering and are often focused on a single media type. In this paper, we argue for the benefits of an event-centric media management (EMMa) approach that looks at the problem of media management holistically. Based on a generic event model, we specify a media event model for the EMMa approach. The single phases and processes of the EMMa approach are defined in a general process chain for an event-centric media management, the EMMa cycle. This cycle follows the event concept throughout all phases and processes of the chain and puts the concept of events in the center of the media management. Based on the media event model and EMMa cycle, we design a component-based architecture for the EMMa approach and conduct an implementation of the approach.

Keywords: Events, Event-centric Media Management, Event Classification, Event Clustering, Multimedia Authoring

1. INTRODUCTION

Since the advent of digital cameras and video recorders, the management of media assets has increasingly become an essential aspect of our daily life. Today's approaches like Flickr (http://www.flickr.com), Picasa (http://picasa.google.com), and YouTube (http://www.youtube.com) for organizing and sharing our experiences focus on the media assets that accompany these experiences. Thus, these applications are media-driven or media-centric in order to manage human experience. In recent years, it has been reinforced that *events* are a much better abstraction of human experience [1]. Thus, events are much better for managing media assets captured during events. As a consequence, we find today approaches and applications like SenseCam [2], MyLifeBits [3, 4], PhotoCompas [5], World Explorer [6], FotoFiti [7], PhotoFinder [8, 9], and many more that integrate the concept of events into their media management solution. These are very important and valuable steps towards an event-centric media management. However, the existing approaches and applications typically consider events only as second-class entities that can be extracted from the media assets and are attached to them as additional metadata. Here, events are one concept among many such as the actual media management, a social network support, and others. In addition, they mostly focus on specific problems for event management like event detection, tagging, sharing, classification, or clustering and provide support for a single media type.

What is needed is an Event-centric Media Management (EMMa) approach that follows the event-concept consequently. The EMMa approach leverages events throughout all processes of the media management cycle from event creation up to the use and presentation of events. It defines the EMMa cycle, a general process chain for media event management. The concrete EMMa application implements the EMMa cycle. It supports multiple types of media assets and organizes and stores them in terms of events. The user can explore the events stored in the EMMa application using an interactive interface performing a blended browsing and querying. While exploring events, relationships between events can be defined and compositions of events created. Finally, multimedia presentations can be created of the events stored in the EMMa application and the media assets associated to these events, respectively.

The remainder of the paper is organized as follows: In Section 2, we discuss and compare in detail the media-centric approach and event-centric approach for media management and argue for the benefits of an event-centric approach. In

Section 3, we present the generic, domain-independent event model E. This event model is specialized with the media event model towards the specific needs and requirements for an event-centric media management in Section 4. The phases and processes of the general EMMa cycle for media event management are presented in Section 5. The architecture of the EMMa application is defined based on the EMMa cycle. It is described in detail in Section 6. Finally, we elaborate the different aspects of the EMMa application in Section 7, before we conclude the paper.

2. MEDIA-CENTRIC VS EVENT-CENTRIC MEDIA MANAGEMENT

The vast amount of digital cameras, video cameras, and cell phones with build-in photo and video capture capabilities more and more reinforces the need for an easy and practical management of the media assets captured. Consequently, a huge number of applications have been developed in recent years that provide support for the management of media assets. Well known examples are Flickr and Picasa for photos and YouTube for videos. These applications provide many useful features for media management including easy manipulation, sharing, annotation, and searching of the media assets. However, most of the applications we find today pursue a media-centric approach for their management efforts. This means that they consider the media assets as the primary objects for managing human experience. The events at which the media assets are captured are merely some metadata of these media assets. Such a media-centric management has some inherent drawbacks as a previous analysis of related work discovered [10]. Consequently, today’s applications for a media-centric media management typically lack in an effective support for one or multiple of the following aspects: real multimedia support, unified indexing, utilization of social context, and abstraction of human experience. We present these typical drawbacks of media-centric applications and contrast them with an event-centric approach for media management. We show the advantages of an event-centric approach over a media-centric approach and argue for its benefits for media management.

2.1 Multimedia Support

Today’s applications for media management typically provide support for one media type only. For example, we find Picasa, ACDSee (http://www.acdsee.com), and CorelPhotoAlbum (http://www.corel.com) for photos and YouTube and Google Video (http://video.google.com) for videos. Academic research is also mainly focused on media specific solutions such as AT&T Lab’s Shoebox [11, 12], Microsoft Research's MiAlbum [13], and Autoalbum [14]. Even applications that support multiple media types like SmartAlbum [15] typically do not explicitly allow for the particular characteristics of the different media types like support for different content information and context information.

In an event-centric approach, media assets are the documentary support for an event. The media assets are “merely” used to describe the events that took place. Thus, an event-centric approach provides a natural and intuitive support for media assets of different media types. As such, event-centric applications are media aware as they use media assets for describing events. At the same time, they are agnostic in regard of media types as the particular type of media assets is not important for the documentation. Such a natural support for multimedia is very promising. Studies show that different types of media assets are very important to the users [16].

2.2 Unified Indexing

Today’s state of the art search engines from Google (http://www.google.com), Yahoo! (http://search.yahoo.com), Microsoft (http://www.live.com), and others support for searching in different media types such as web sites, images, and videos. However, they index each media type separately and thus also the searches are separated completely. It is left up to the users to conduct independent searches to find media assets of different media type and collate them. Thus, it could be argued that they still are mono-media. Although there are media-centric applications that allow for searching over different media types such as Media Assistant [17], Google Desktop (http://desktop.google.com), and Quicksilver (http://www.blacktree.com), there is no natural or apparently common indexing scheme for different media types in a media-centric approach.

As events are media type agnostic (see Section 2.1), they provide a natural choice for a unified indexing over various media types. Event-centric applications like [18, 19, 20] show different types of media assets as different facets of the events they document. From the authors’ point of view, there is no other intuitive way to index multimedia assets from multiple users.

2.3 Social Context

Media assets taken by different users at the same event share a common social context. For example, consider a group of people taking a trip together to New York. Each member of this group is individually taking photos during the trip and subsequently creating a photo album of the media assets. In a media-centric approach, one cannot easily combine the different users' photographs and experiences conveyed with the created photo albums. Important information about the social context is missing like the people that participated in the trip and the single events that happened during the trip. Even applications that allow for annotating the photographs with event information like Flickr's tags do not provide a solution to the problem. As users can search by tags, the result set contains images not only of that particular trip but all images associated with the searched tags. To alleviate the problem, media-centric media management applications integrated social network functionalities. For example, the media management applications Picasa and Kodak Easyshare Gallery (http://www.kodakgallery.com) allow the users to share media assets with their friends.

However, in an event-centric media management approach, the support and exploitation of social context comes more natural than in media-centric applications. Events are by their nature embedded in a social context. Since the participants are part of an event description, the social context of the event is inherently captured in an event-based approach [21]. Leveraging the relations between participants in the social network, one can find which users may be interested in an event or share an event with a specific group in a social network like colleagues at work or school friends.

2.4 Human Experience

Most media management applications such as Picasa and ACDSee use a "files and folders" approach for organizing media assets. This means that the media assets are stored as files on the hard drive. Different folders are used to organize these media assets. However, each file can only appear in one folder to avoid redundancies. A solution to this problem could be to define references to the files from different folders. However, managing these references is difficult and error-prone. Other applications like Flickr and YouTube allow for annotating and searching media assets by tags. There is also application support for semi-automatically tagging the media assets such as ZoneTag Photos (http://zonetag.research.yahoo.com) for Flickr. However, tag-based applications are not an improvement to the "files and folders" approach. As tags do not impose any structure on the organization of the media assets, each of the vast amounts of different tags used to annotate the media assets can be considered an individual "folder" in the application. To provide for an appropriate support for managing human experience, a more flexible organization of the media assets is needed than the "files and folders" approach allows. At the same time, the organization of the media assets needs a more structural approach than it is provided by tags.

Research in cognitive neuroscience [22] and philosophy [23] shows that humans think in terms of events. Events provide a natural abstraction of the real world. They encapsulate the semi-structured nature of real world data such as media assets captured by the users. Events provide the flexibility needed for organizing these media assets and at the same time impose a structure on their organization that is appropriate to the human experience and reminisce of events.

2.5 Summary

As the discussion above shows, the media-centric approach is primarily organizing the media assets, i.e., the documentary support for events. In contrast, the event-centric approach puts the management of the actual experience of the users in focus. Thus, events are a more natural means for organizing and managing media assets and human experience. In an event-centric approach, the media assets captured during an event merely describe the experience and provide documentary support that a specific event actually happened and how it happened. In order to provide an event-centric media management application, a powerful event model is required. Such a generic, domain-independent event model is presented with E in Section 3. As E is applicable in a variety of domains, it needs to be adapted to the particularities of an event-centric media management. This specialization of E towards a media event model is conducted and presented in Section 4.

3. GENERIC EVENT MODEL E

The event model E [1] introduced by Westermann and Jain has been developed based on an analysis and abstraction of events in various domains such as research publications [21], personal media [24], meetings [19], enterprise collaboration [25], and sports [18]. It provides a generic structure for the definition of events and is extensible to the requirements of events in the most different concrete applications and domains. A UML diagram of the generic event model is shown in Figure 1. The primary objects in E are separated in events, descriptors, and entities. Events subsume telic events, (atelic) activities, and constellations. They are first-class objects and do not need any other supporting data. Telic events are tending toward an end or outcome whereas the atelic activities or short activities are describing processes (for telic and atelic see [26]). A constellation represents an association of arbitrary complexity between two or more events. Since such a discovered association may be a significant occurrence itself, constellations are event-like.

The descriptors exist to describe an event. Each descriptor may describe exactly one aspect of an event. Aspects include the spatial, temporal, structural, informational, experiential, and causal facets of events. E allows for an `Attribute-Value` that may be any arbitrary object. This can be used to add a descriptor of any data type to an event. Applications extend E by defining application-specific types of events and attribute values.

E allows event data to be linked to external references called entities. Entities are of two types: concepts and sensor data. Concepts can be attached to an external knowledge source and describe some abstract entity in the domain of the event. Events can also be linked to concrete physical data using the sensor data entity. References to sensor data can store among others type and size of the data. In the domain of media management, the sensor data are the media assets, i.e., the pictures taken and the video clips recorded.

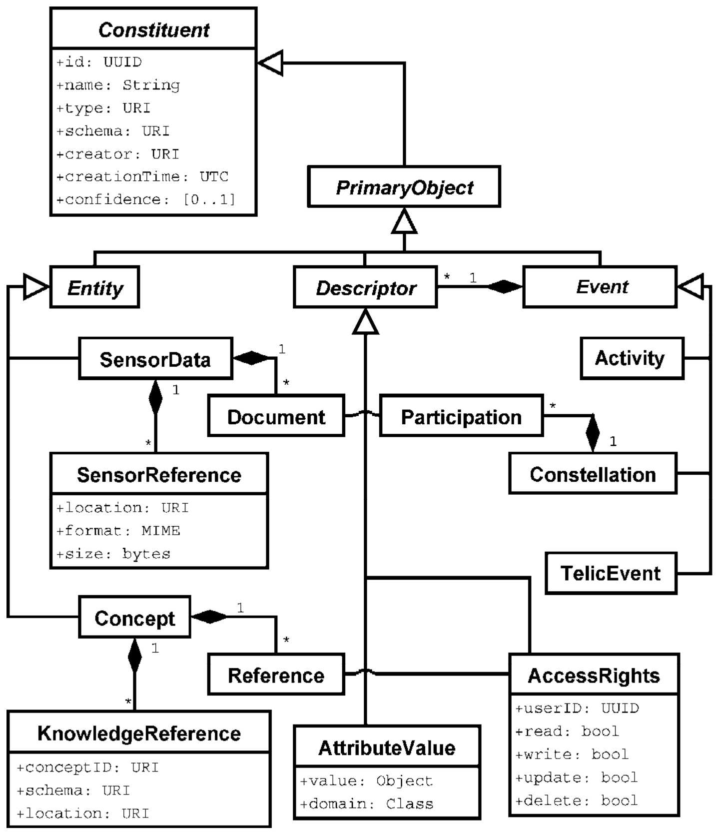

Figure 1: UML class diagram of the generic event model

The event structures provided by E are abstract. To make them useful in any given application and domain, we introduce the notion of event types. An event type is a specific structure of the relationships between a set of classes of E shown in Figure 1. The event type describes a "class" of events and their associated descriptors, types, and constraints on the values of those descriptors. Consequently, an event of a given event type are the objects of the classes defined in the event type. Thus, events can be said to be "instances" of event types. Event types allow defining relationships between different events in a concrete domain. They can be used to build hierarchies and ontologies. For it, E provides for defining inheritance relationships between different event types. For example, an event type "vacation" can be specialized among others towards a "beach holiday" event type and "activity holiday" event type. However, these inheritance relationships only exist between the event types and not between the instances of these types, i.e., the single concrete events. For example, a one week holiday on Hawaii is an instance of the "beach holiday" event type whereas a safari tour is an "activity holiday". Both events have the abstract event type "vacation" in common but there is no specialization relationship between the two concrete events (even though there can be other kind of relationships such as before, after, same participants, and others). Inheritance between event types is merely a means to structure the events that can happen in a specific domain.

The simplest events in E are atomic events, which are derived from `TelicEvent`. They cannot be further divided into other events. An atomic event can be linked to only one sensor data. Thus, in the domain of media management an atomic event can be linked to one media asset. For reasons of convenience, E allows the storage of multiple versions of the same media element in `SensorReference` objects. These `SensorReference` objects point to the actual media asset and may differ, e.g., in resolution and storage location. This allows applications to, e.g., use a thumbnail of an image for fast display of events and using a full resolution of the same image for processing. Or, if a specific media storage location is not accessible, it can use an alternate storage. The definition of the atomic event type is shown in Figure 2. For reasons of clarity, the classes common to the generic event model shown in Figure 1 are omitted in this and all further event type diagrams. Atomic events are the simplest form of events of which all other, more complex events are composed. Constellations are the most powerful type of events. In order to harness this power while ensuring utility, the constellation class must be extended and restricted to the specific requirements of a concrete application. A very generic specialization of constellations is the composite event type. Composite events can be used to express part-of relationship between events. The definition of the composite event type is shown in Figure 3.

Composite events can be distinguished based on their level of complexity [27]. Simpler, application-independent compositions are called elemental events. Events that are semantically recognized by the users of the application are called domain events. Application-specific composite event types can be defined by enumerating the event types of which it is composed as well as the relations between them. In general, events can be related in composite events by attributes that allow for defining part-of relationships such as time, location, and participants. The latter are either people or objects (real world items). In constellations, events can also be linked by dependence or causal relationship. Dependence exists when one event must occur for another to occur, e.g., “eat cake” can only happen subsequently to “bake cake”. Causal exists when an event occurrence implicates another event, e.g., “kick ball” causes “ball moves”.

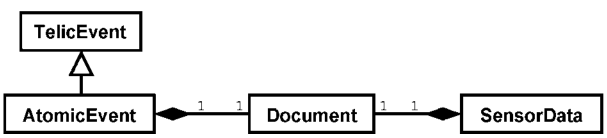

Figure 2: UML class diagram of the atomic event type

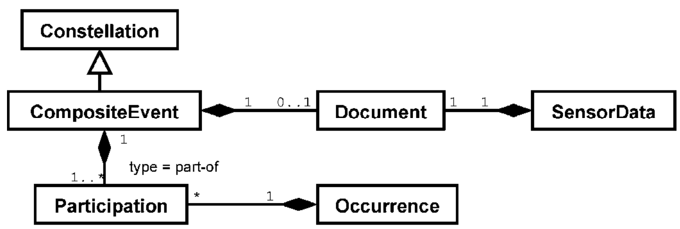

Figure 3: UML class diagram of the composite event type

4. MEDIA EVENT MODEL

In order to provide support for an event-centric media management application, the generic event model E defined in the previous section must be extended and specialized towards a media event model. This is conducted by defining application-specific event types for media management and by determining common descriptors for these media event types. The media events extending the generic model towards media management are shown in Figure 4. Their common descriptors are depicted in Figure 5. All media events contain descriptors for participant people and objects. In addition, comments can be added to media events (cf. the comment annotation feature of Flickr and others). All events can be described by tags. These can be either manually entered or automatically generated. Tags can be used to represent conceptual times like “Christmas” or locations such as “New York City”. Using the generic event model E, these tags may be stored in references to concepts that link to external knowledge sources such as the Getty Thesaurus of Geographic Names (http://www.getty.edu/research/conducting_research/vocabularies/tgn). The simplest events in the domain of media management are the `AtomicTextEvent`, `AtomicAudioEvent`, `AtomicVideoEvent`, and `AtomicPhotoEvent` that occur when a media asset is captured. They are derived from the common superclass `MediaSensorEvent`, which is an atomic event provided by the generic event model E.

The UML diagram of the atomic photo event type is shown in Figure 6. It contains metadata specific to photographs. In regard of `ContextInformation`, we consider among others the photos’ EXIF header [28], low-level content information like color histograms and textures, and high-level content information in form of visual characteristics like visual words. The `LowLevelContentInformation` class stores vectors of real number that represent color or texture characteristics. Finally, `VisualInformation` stores vectors of IDs of the visual characteristics. Atomic video, audio, and text event types are defined similarly.

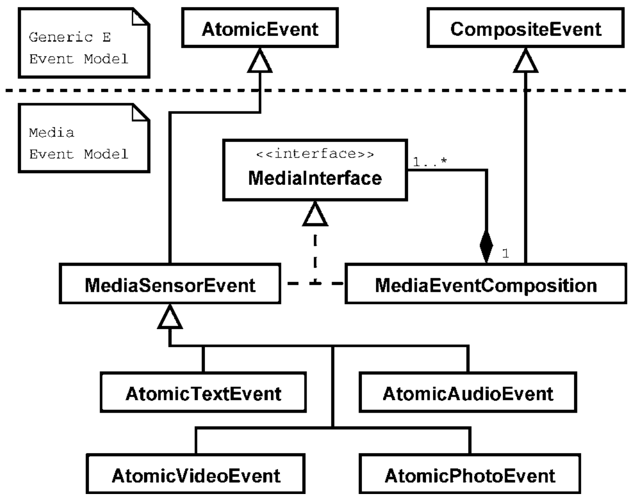

Figure 4: UML class diagram of the generic event model E specialized for the domain of media management

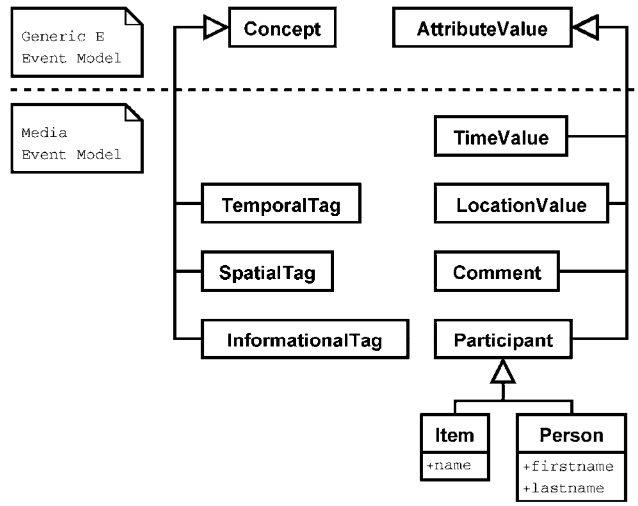

Figure 5: UML class diagram of the common descriptors of events for media management

The composition of atomic media events is modeled with the class `MediaEventComposition`, which is derived from the `CompositeEvent` of the generic event model. In general, the `MediaEventComposition` could contain any event, i.e., atomic event, composite event, and constellation as defined in the generic event model. To make sure that only media events can be used with the media event composition, the `MediaInterface` is introduced. Both, the `MediaSensorEvent` and `MediaEventComposition` implement it. This allows for constraining the media event compositions to only those classes that implement the `MediaInterface`, i.e., to media events. An example of a composite event type is the "birthday party" event type shown in Figure 7. It has attributes specific to a birthday party, e.g., it defines that there is a person whose birthday is celebrated. Other attributes of the "birthday party" event type are objects like a "cake" and "candles".

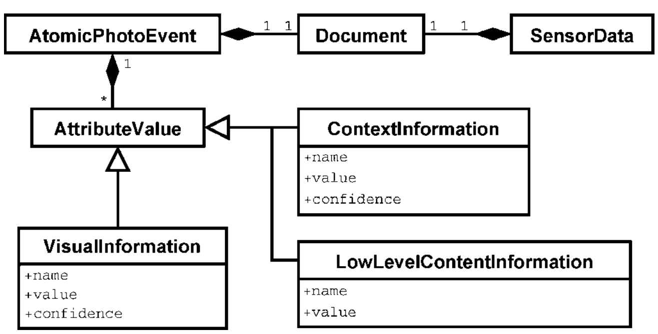

Figure 6: UML class diagram of the atomic photo event type

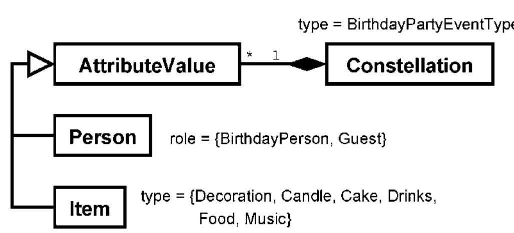

Figure 7: UML class diagram of the birthday party type

5. MEDIA EVENT MANAGEMENT CYCLE

In an event-centric media management approach, users typically follow specific processes of a general media event management cycle. This media event management cycle is depicted in Figure 8. It consists of several processes, organized in three phases of Event Digestion, Event Browsing and Annotation, and Event Presentation. In the first phase, the Event Digestion, users add new media assets to the media management application such as a set of new photos taken or video clips recorded. For each media asset, a new atomic media event is created and ingested into the eventbase. The atomic media events are digested and enriched with metadata by applying enrichment processes such as Feature Extraction, Classification, and Tagging. In the second phase, the Event Browsing and Annotation, the initially created atomic media events are assembled to composite media events. Here, spatial clustering and temporal clustering is conducted in the Clustering process to group the atomic media events to composite media events. In addition, clustering among other dimensions can be conducted such as color histogram or face detection. The subsequent Event Browsing process allows for interactively exploring the media eventbase, manually refining the created composite events, and defining new ones. In the Event Annotation process, users assign an event type to the composite event, e.g., the "birthday party" defined in Section 4, "dinner", or "meeting". Finally, in the Presentation Authoring phase the media events and their associated media assets are leveraged for creating multimedia albums. Multimedia albums are an extension of traditional page-based photo books with support for continuous media types such as audio and video and navigational interaction in form of hyperlinks. For creating a multimedia album, first appropriate media events are selected by the Event Query process. These can be atomic media events as well as composite media events. Then, the associated sensor data of the media events, i.e., the media assets are assembled by the Media Assembly process in time, space, and interaction into a coherent multimedia presentation. Finally, the multimedia presentation is delivered to the end user for consumption by the Delivery and Presentation process. The media events used for creating a multimedia presentation form a new composite media event. Thus, they are stored back in the media eventbase together with the created multimedia presentation as its sensor data. The media event model presented in Section 4 is used to communicate between the different processes and phases of the chain.

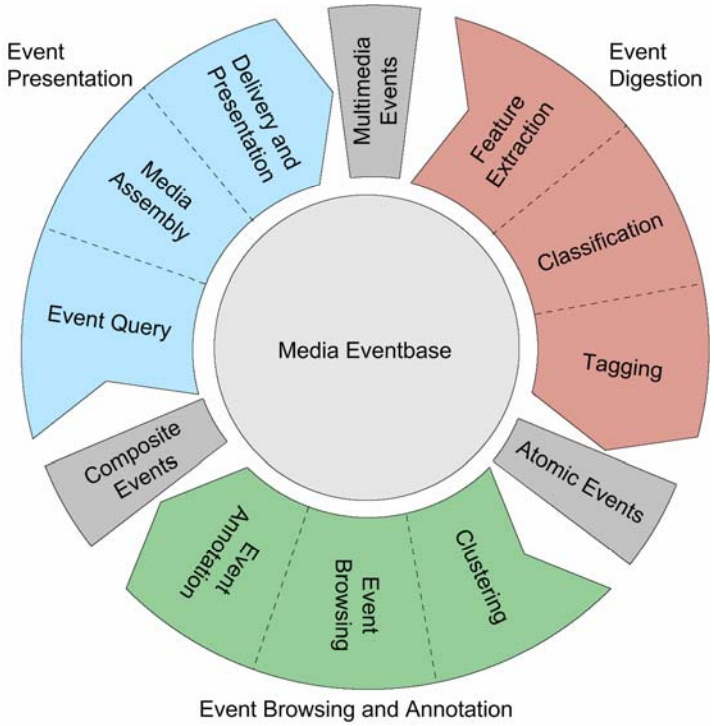

Figure 8: Media event management cycle

6. ARCHITECTURE OF THE EMMA APPLICATION

Based on the media event management cycle, we developed the component-based architecture of the EMMa application. This architecture is depicted in Figure 9. As the figure shows, we are pursuing (so far) a traditional client/server approach for the EMMa architecture. It consists of several layers realized by distinct software components. The components are described along the layers from bottom to top: The server side of the EMMa architecture consists of the Storage Layer and the Service Layer. The Storage Layer provides three different databases for storing and managing media events, media assets associated to the events, and user data such as login, password, and access rights to events. On top of the three databases are three wrapper components that abstract from accessing the databases and provide services to the upper layer. The Active Record framework [29] on the left hand side abstracts from accessing the media events stored in the relational database MySQL (http://www.mysql.com). It allows the Service Layer to ingest, retrieve, and manipulate events. In the middle, another Active Record framework abstracts from accessing the user data. For storing the user data, we also use a MySQL database. The Filesystem Wrapper on the right hand side abstracts from accessing media assets assigned to events. The reference between the media events and media assets is build by using URIs pointing to the media content. This allows employing multiple media storage solutions. Each of it could be optimized for a specific media type. Currently, we use a simple filesystem solution to store our media assets. Access to these media assets is provided by a Tomcat web server. In a future version, the simple filesystem solution will be replaced by a media database.

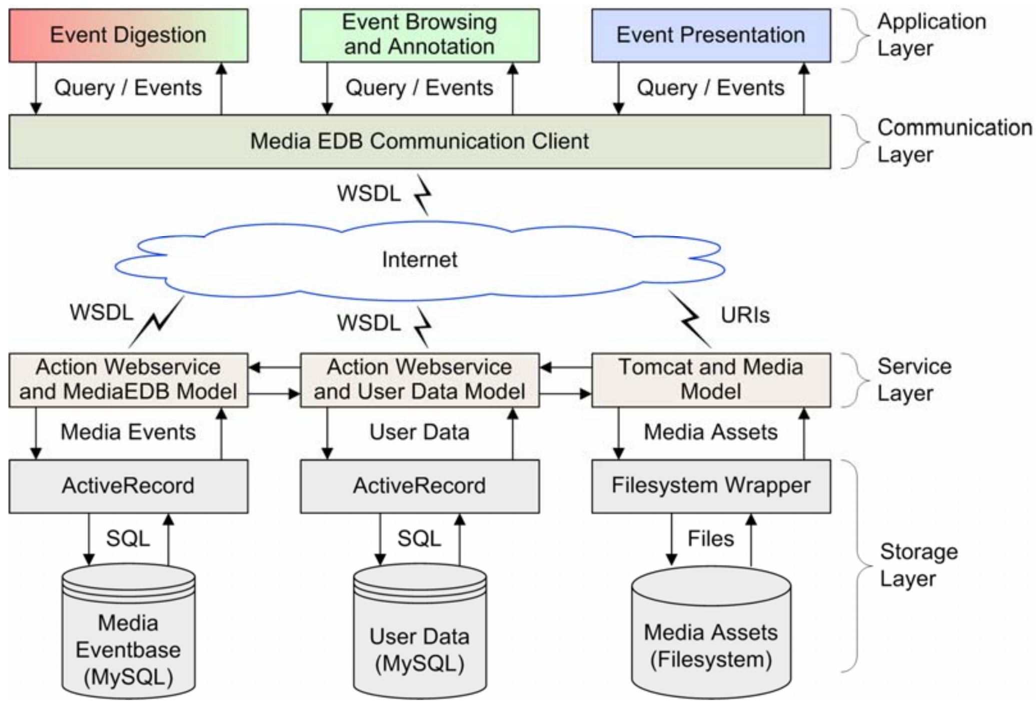

Figure 9: Architecture of the event-centric media management application

On top of the Storage Layer resides the Service Layer. It provides access to the media events, user data, and media assets from the Internet. The Action Webservice and MediaEDB Model component implements the media event model defined in Section 4. It provides via Action Webservice (http://manuals.rubyonrails.com/read/book/10) functionality such as storing atomic events and composite events, retrieving events, deleting events, and updating events. The component is implemented by using Ruby on Rails [30]. The functionality is provided to the client in form of WSDL [31] web services, i.e., clients exchange the media event information in form of XML documents. For access control and authorization purposes, the Action Webservice and MediaEDB model component also connects to the Action Webservice and User Data Model component. The Tomcat and Media Model component provides access to the media assets of the events such as photos and videos. Like the component for accessing the media events, also the Tomcat and Media Model component uses the Action Webservice and User Data Model component for access control and authorization purposes.

The client side of the EMMa architecture consists of a Communication Layer and an Application Layer. The Communication Layer is provided by the Media EDB Communication Client component. The component controls the communication between the upper Application Layer and the lower Service Layer. The overall goal of the Communication Layer is to provide an easy to programming interface for accessing the media eventbase, user database, and media assets storage. For it, the Media EDB Communication Layer component converts media events retrieved via WSDL from the Action Webservice and Media EDB component into Java objects and vice versa. At the same time, it allows clients to access the user data and media storages. Thus, it unifies the access to the media events, user data, and media assets for the event-centric media management functionalities of the Application Layer. The Application Layer provides the user interface of the EMMa architecture and consists of three components for Event Digestion, Event Browsing and Annotation, and Event Presentation. These components correspond to the three phases of the media event management cycle presented in Section 5. Each component has several subcomponents as the architecture of the Application Layer in Figure

10 shows. Initially, a batch of media assets is converted into atomic media events through the Converter component. These atomic media events go through an Event Digestion Manager that enriches them with metadata such as EXIF, classifications, and tags. In addition, event clustering is conducted (either on the initial batch of atomic media events only or on all media events stored in the eventbase). The Event Browsing and Annotation Manager component allows for interactively exploring the eventbase. It is supported by six view components each providing a distinct view on the media events. These views are space, time, persons, items, activities, and tags. Finally, the event presentation is provided by the components for Event Query and Media Assembly that select events from the media eventbase and organizes the associated media assets into a multimedia album. This album is transformed by the Transformation component into today's presentation formats like SMIL [32], SVG [33], and Flash [34] and is delivered to the users for consumption. Finally, the events used for the multimedia album create a new composite event that is stored back in the media eventbase.

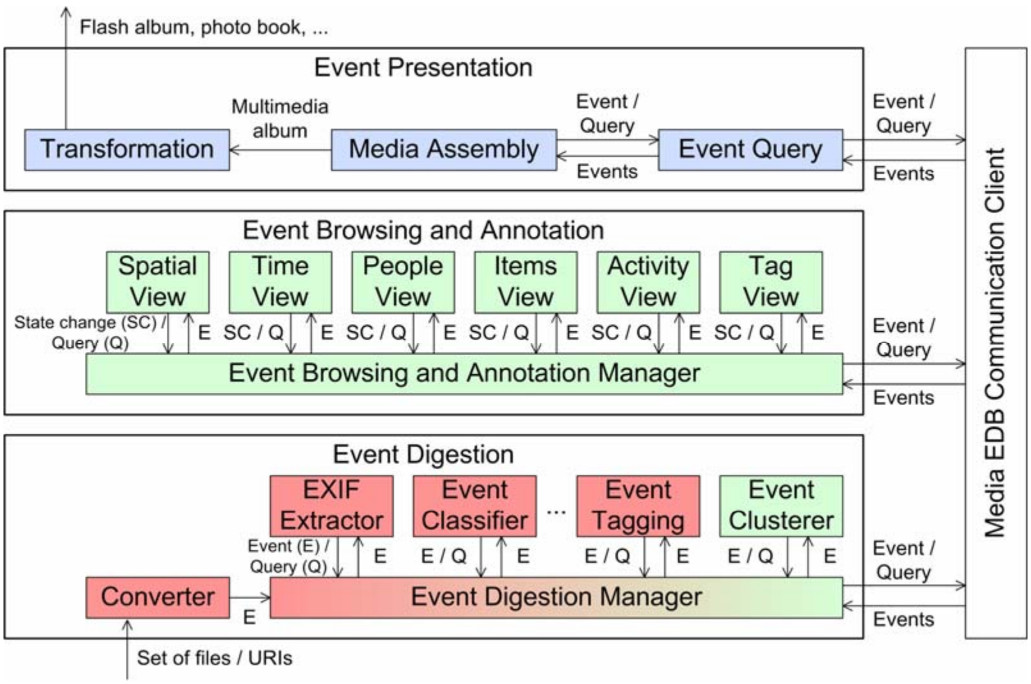

Figure 10: Application layer of the event-centric media management application

7. EMMA APPLICATION

In the previous section, we presented in detail the EMMa architecture. We described its layers and components as well as the communication between them. The Application Layer of this architecture is actually providing the user interface of our event-centric media management application. In this section, we describe in detail the user interface of the EMMa application and elaborate the support the application provides for the processes of the general media event management cycle introduced in Section 5. We begin with the processes of the Event Digestion phase. Subsequently, we elaborate the processes of the Event Browsing and Annotation phase, before the Event Presentation phase's processes are described.

7.1 Processes of the Event Digestion Phase

The first phase of the media event management cycle considers ingesting new atomic media events such as a picture taken and a video clip recorded. For each set of newly ingested atomic media events, one composite event is created. This composite event comprises the ingested atomic media events as its parts. It can be considered as the *film roll* of the ingested atomic events (compare the notion of film rolls in the analog picture taking process). Subsequently, the atomic media events are digested, i.e., they are enriched with further metadata. In the Event Digestion phase, the users select via a file dialog a folder containing the media assets that shall be ingested as atomic events into the EMMa application. The files in this folder are scanned and the processes Feature Extraction, Classification, and Tagging of the media event management cycle are applied.

Feature Extraction When new media assets are ingested as atomic media events into the eventbase, some feature extraction algorithms are applied to enhance the events with basic metadata. The media assets are processed to extract their context information and content information. For example, for photos the content information is provided in form of color histogram and texture. Context information is retrieved from the EXIF headers which contain among others the camera settings, time, and location.

Classification Based on the results of the feature extraction, we classify the ingested atomic media events. For example, for atomic photo events we use image retrieval and classification approaches based on the automatic camera settings retrieved from the EXIF headers. With automatic camera settings, we retrieve the optical parameters such as focal length, aperture, and exposure time. The classification is conducted by applying unsupervised learning algorithms to cluster images, e.g., [35, 36]. The optical parameters of a particular photograph are used to compute a light metric and its depth of field. A training set of more than 2500 photographs was hierarchically clustered using these two parameters. The photographs were also manually annotated with 50 overlapping classes. The optical characteristics are able to differentiate many classes. The classes with the highest likelihood for a given photo are used to annotate the corresponding event.

Tagging Having conducted the Feature Extraction and Classification processes, there is an optional Tagging process of the atomic media events. With tagging, we mean manually adding keywords to the atomic media events as it is provided by today's media sharing applications like Flickr and YouTube. The user can also accept or reject the classes automatically detected in the previous process. Tags added to the events include but are not limited to describe the activity shown in the event, location of the event, and other context data. Tags for describing different aspects of events are stored separately, i.e., location tags are clearly differentiated from informational tags that describe the content of media. For the Tagging process, we are also looking at tag-suggestions applications such as ZoneTag Photos (see Section 2.4) and the Mobile Media Metadata approach [37].

7.2 Processes of the Event Browsing and Annotation Phase

Once the atomic media events are successfully ingested into the eventbase and the events are digested in terms of adding content information and context information, the atomic media events are passed to the Event Browsing and Annotation phase of the media event cycle. In addition to interactively exploring and annotating the media events, the purpose of this phase is to determine further composite events. These composite events can be determined manually, semi-automatically, or fully automatically. So far, we are focusing on the manual and semi-automatic creation of semantically valid composite media events. The atomic media events are passed in the Event Browsing and Annotation phase through the three processes Clustering, Event Browsing, and Event Annotation.

Clustering A semi-automatic creation of composite events is conducted in the Clustering process, where atomic media events are grouped according to specific event dimensions such as time and space into clusters. In practice, events are hierarchical [38] and range from elemental level to domain level [27]. Thus, the clusters eventually determine hierarchical composite events. Time information is the most important dimension to calculate them. On the domain level, we pursue a spatial clustering of media events based on GPS information. This feature has been developed in one of our previous projects [20, 39]. In addition to the domain level event detection by clustering, we also support elemental level event detection. For determining elemental level events, we pursue an event clustering approach that combines both time information and visual information in form of color histograms. As introduced above, the Clustering process is semi-automatically. This means that after applying various algorithms to cluster the media events along different dimensions, the automatically calculated clusters will be presented to the users who can modify and further refine them. Once the users confirm specific clusters, new composite media events are created for each cluster and added to the media eventbase.

Event Browsing Once composite events have been determined in the Clustering process, the events are presented to the user. The users can navigate through the events stored in the eventbase in a blended querying and browsing approach [40]. This means that while the users are browsing through the media events displayed on the screen, queries on the eventbase are executed in the background to populate and present the browsing results. Both steps, the querying and browsing, are conducted in small turns.

The users can browse through the events according to six different querying dimensions as specified in the EMMa architecture in Section 6. These dimensions are the events' time, location, involved persons, used or displayed items, activities, and tags. Each of the querying dimensions has a corresponding browsing view. For example, for the location view the media events are visualized using a map as shown by the interface prototype in Figure 11. The map can be searched by either manually panning the map or searching for locations by name. Composite media events are shown on the map as polygons that outline their spatial extent. Atomic media events of a composite media event are shown as icons within the polygons. Clicking on an event's thumbnail brings up other details like participants and time. For the time view a timeline is chosen to present the media events as the interface prototype in Figure 12 depicts. The timeline presents the events in chronological order. It also shows the events one level above and one level below the current media event to establish its context. The events' details, the media assets related to them, and their component events are displayed above the timeline. The granularity of the timeline can be varied to change the time span of events displayed. For the EMMa application, we implemented so far prototypes of the time and location view of events.

Figure 11: Interface prototype of the location view of the EMMa application

Figure 12: Interface prototype of the timeline view of the EMMa application

The most interesting aspect of our user interface for a blended browsing and querying in the media eventbase will be a seamless transition between the different views. For example, once a user has selected a time span in the timeline view, he or she will be able to switch to a map presentation of the events previously shown in the timeline view. Once switched to the map view, the users can click on one of the events to get further information about it such as the activity, items, or persons involved. If the user clicks on a person's icon, the application will switch to a social network graph presentation showing the relationships between the persons involved in the event. The overall goal is to provide an intelligent and smooth integration of the different views on events in one interface. The blended browsing and querying is not only used for exploring the eventbase but also to further group atomic media events into composite events, i.e., to put the atomic events into relations. These newly created composite media events can optionally be annotated in the Event Annotation process using predefined event types.

Event Annotation Having created a new composite media event, the users can assign an event type (see Section 3) to this event. For our event-centric media management application, we have defined different media event types, e.g., for birthdays, conferences, meetings, dinners, and others. In a future version, the EMMa application will be able to automatically fill in the parameters defined for these event types such as the "birthday child" or "participants of a meeting" with concrete values extracted from the composite event. This can be done automatically for parameters that have a clearly defined part-of relationship like time, location, people, and items. For other, more ambiguously defined event type parameters, we will conduct a semi-automatic or even manual approach and rely on the input from the user for filling in the concrete values. The Event Annotation process can be extended by arbitrary event types. However, so far the event types cannot be edited by the users. A future enhancement of our EMMa application will allow the users themselves to specify new event types and then assign this type to the created composite event.

In addition to the described manual annotation of composite media events by the users, also a (semi-)automatic event type determination could be conducted. Here, event types are determined in a *bottom-up* approach. This means that we could determine the event type based on the atomic media events used for a composite event. The atomic media events are analyzed and an appropriate event type will be assigned based on their characteristics and descriptors.

7.3 Processes of the Event Presentation Phase

In the first two phases of our media management cycle, we consider ingesting and digesting atomic media events, browsing through events, creating composite events, and annotating them. In the last phase, the Presentation Authoring, we actually use the media events stored in the media eventbase. We create multimedia presentations in form of page-based multimedia albums. These albums are composed of the media assets associated with the media events stored in the eventbase. The Event Presentation phase is based on earlier work in the area of authoring personalized multimedia presentations [41, 42, 43, 44, 45]. A context-driven authoring tool for creating page-based multimedia presentations [45] is adapted and enhanced for processing media events and creating multimedia albums based on the events' media assets. The Event Presentation phase consists of the three processes Event Query, Media Assembly, and Delivery and Presentation.

Event Query The first step for creating a multimedia album is to select the atomic media events and composite media events that shall be presented in the album. For it, we specify query parameters selecting appropriate events. These query parameters are according to the six dimensions as introduced in the Event Browsing process in Section 7.2. The result set is a list of events that fulfill the query. This list can be ranked according to the query dimensions. It is used as input to the subsequent Media Assembly process. In addition to the explicit Event Query process in the media event management cycle, we can also directly switch from the Event Browsing process presented in Section 7.2 to the Media Assembly process. This means that once users are in a specific view of the Event Browsing process, they can switch by one click to the Media Assembly process using the latest browsing view as input.

Media Assembly The Media Assembly process determines how the media assets associated with the events can be optimally arranged in an album. The created album can be targeted at different end devices such as Desktop PCs, PDAs, and cell phones. For it, the descriptors of the media events are used to arrange the media assets on different pages of the album. This can be among others cluster information based on time, space, and visual information such as color histograms. The media assets associated with the composite events are arranged in time and space on different pages of the album. In addition, navigational interaction in form of hyperlinks can be defined. Figure 13 shows a screenshot of an album created for an England trip. It depicts the media assets captured at different events during the trip such as a visit at the London Bridge. After the application has suggested a specific design and layout, the users can manually modify this layout. In addition, media assets can be removed and media assets from other events added. Once the users are satisfied with the assembly result, the multimedia album can be delivered and presented to their recipients. This is conducted in the Delivery and Presentation process.

Figure 13: Screenshot of the authoring tool for multimedia albums employing media events

Delivery and Presentation When the Media Assembly process is finished and a new multimedia album is created, we export the multimedia album in one of today's presentation formats' such as SMIL, SVG, and Flash and actually deliver and present it to the users. When the album is delivered in a specific presentation format to a user, the created multimedia presentation is considered determining a new composite media event. This means that based on the events actually used for authoring a multimedia album, we create a new composite media event and store it in the eventbase. This composite media event reflects the different media events conveyed in the album in form of part-of relationships. These part-of relationships between the created composite media event and the media events used in the album are described by very generic media event types such as album pages and sub-albums (a set of related pages in a multimedia album). The created multimedia presentation is stored as sensor data to the created composite event. Thus, the temporal course, spatial layout and navigational interaction defined with the multimedia presentation is stored as part of the experiential aspect of it. Consequently, the created composite media event can be considered as a real *multimedia event*.

8. CONCLUSION AND FUTURE WORK

In this paper, we compared the media-centric approach and event-centric approach for media management. We argued for the benefits of an event-centric media management such as natural support for multimedia, a common schema for a unified indexing of different media types, a better abstraction of the real world and human experience, and a more intuitive support for multiple users. We introduced the generic, domain-independent event model E as foundation of event-centric applications. To make the generic event model applicable for the domain of media management, we defined with the media event model a specialization of E. As conceptual basis of our event-centric media management application, we introduced the media event management cycle. The cycle shows the different phases and processes involved in an event-centric media management. It has been used as input to define the architecture of the EMMa application. Finally, we presented the different aspects of the user interface of the EMMa application. Users can ingest new media assets in form of atomic media events, browse through the media events stored in the eventbase, and employ the media assets to create multimedia albums.

The EMMa application presented in this paper is a work in progress. Although the application already provides support for all phases of the media event management cycle it is still in an early state. Thus, there is much scope for improvement and adding of further event processing features. For example, so far we assume for reasons of simplicity that the EXIF data is correct, i.e., we assume that the users always set the right time and date in their cameras. Thus, an important aspect for a reliable EMMa application is to make sure that misaligned camera settings are somehow realigned or at least signaled to the users for correction. Although, the EMMa application is media type agnostic, we are focusing so far on media events for photos and conducting research on video events. Thus, other media types such as audio and text documents need to be considered in future. In addition, one could think of different architectural styles for the EMMa application. For example, multiple event servers and media servers could be linked together, perhaps using a peer-to-peer network as proposed in [46]. Different architectural styles could be analyzed in regard of scalability and reliability. Also performance such as responsiveness and throughput can be measured and compared. In addition, besides the six views currently identified for the user interface we could define additional views that combine more than one aspect. For example, a spatio-temporal view could present the events in a three dimensional environment. Finally, user studies need to be conducted to evaluate the usability and efficacy of the EMMa application.

Acknowledgment We thank our colleagues Bo Gong and Pinaki Sinha for providing their clustering and classification features. In addition, we thank Utz Westermann for his input on the media event model and Amarnath Gupta for his general advice on event-based media management. We also thank our students Tanmay Goal and Ritesh Subramanian for their valuated effort on implementing the event-centric media management application. This research is partly supported by a Marie Curie International Fellowship within the 6th European Community Framework Programme.

REFERENCES

1. U. Westermann and Ramesh Jain, “Toward a Common Event Model for Multimedia Applications”, IEEE MultiMedia 14(1), 19-29 (2007), IEEE Computer Society Press.
2. J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell, “Passive capture and ensuing issues for a personal lifetime store”, Proc. of the 1st ACM workshop on Continuous archival and retrieval of personal experiences; New York, NY, USA, 48-55 (2004), ACM Press.
3. J. Gemmell, G. Bell, and R. Lueder, “MyLifeBits: a personal database for everything”, Commun. ACM 49(1), (2006), ACM Press. [Also published as Microsoft Technical Report MSR-TR-2006-23]

4. J. Gemmell, A. Aris, and R. Lueder, “Telling stories with MyLifeBits”, IEEE International Conference on Multimedia and Expo Amsterdam, Netherlands, July (2005).
5. M. Naaman, R. B. Yeh, H. Garcia-Molina, and A. Paepcke, “Leveraging context to resolve identity in photo albums”, Proc. of the 5th ACM/IEEE-CS joint conference on Digital libraries; Denver, CO, USA, (2005), ACM Press.
6. S. Ahern, M. Naaman, R. Nair, J. Yang, “World Explorer: Visualizing Aggregate Data from Unstructured Text in Geo-Referenced Collections”, ACM/IEEE-CS Joint Conference on Digital Libraries; Vancouver, Canada, June (2007).
7. B. N. Lee and W. C. and E. Y. Chang, “Fotofiti: web service for photo management”, MULTIMEDIA, Santa Barbara, CA, USA, (2006).
8. B. Shneiderman and H. Kang, “Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos”, International Conference on Information Visualisation, (2000), IEEE Computer Society.
9. H. Kang and B. Shneiderman, “Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder”, International Conference on Multimedia and Expo, New York City, NY, USA, August (2000).
10. U. Westermann and R. Jain, “E - A Generic Event Model for Event-Centric Multimedia Data Management in eChronicle Applications”, IEEE Int. Workshop on Electronic Chronicles, (2006).
11. T. J. Mills, D. Pye, D. Sinclair, and K. R. Wood, “ShoeBox: A Digital Photo Management System”, AT&T Laboratories Cambridge, (2000).
12. K. Rodden and K. R. Wood, ”How do People manage their Digital Photographs?”, Conference on Human Factors and Computing Systems, (2003), ACM Press.
13. L. Wenyin, Y. Sun, and H. Zhang, “MiAlbum - A system for home photo management using the semi-automatic image annotation approach”, MULTIMEDIA, Marina del Rey, California, USA, (2000), ACM Press.
14. J. C. Platt, “AutoAlbum: Clustering Digital Photographs using Probabilistic Model Merging”, IEEE Workshop on Content-based Access of Image and Video Libraries, (2000), IEEE Computer Society Press.
15. T. Tan, J. Chen, P. Mulhem,and M. Kankanhalli, “SmartAlbum: a multi-modal photo annotation system”, MULTIMEDIA, Juan-les-Pins, France, (2002), ACM Press.
16. F. Bentley, C. Metcalf, and G. Harboe, “Personal vs. commercial content: the similarities between consumer use of photos and music”, CHI, Montreal, Quebec, Canada, (2006).
17. B. Gandhi, A. Martinez, F. Bentley, “Intelligent multimedia content management on mobile devices”, Multimedia and Expo, (2004), IEEE Computer Society Press.
18. G.S. Pingali, A. Opalach, Y.D. Jean, and I.B. Carlbom, “Instantly indexed multimedia databases of real world events”, IEEE Transactions on Multimedia, 4(2), (2002).
19. R. Jain, P. Kim, and Z. Li, “Experiential meeting system”, ACM SIGMM workshop on experiential telepresence, Berkeley, California, USA, (2003).
20. U. Westermann, S. Agaram, B. Gong, and Ramesh Jain, “Event-centric multimedia data management for reconnaissance mission analysis and reporting”, MULTIMEDIA, (2006), ACM Press.
21. B. Gong, R. Singh, and Ramesh Jain, “ResearchExplorer: Gaining Insights through Exploration in Multimedia Scientific Data”, ACM SIGMM Int. workshop on Multimedia information retrieval, (2004), ACM Press.
22. Y. Wang and Y. Wang, “Cognitive Models of the Brain”, Proc. of the 1st IEEE Int. Conf. on Cognitive Informatics, (2002), IEEE Computer Society Press.
23. R. Casati and A. Varzi, “Events”, Stanford Encyclopedia of Philosophy, (2007), URL: http://plato.stanford.edu/entries/events/
24. P. Appan and H. Sundaram, “Networked multimedia event exploration”, MULTIMEDIA, (2004), ACM Press.
25. P. Kim, M. Podlaseck, and G. Pingali, “Personal chronicling tools for enhancing information archival and collaboration in enterprises”, Continuous archival and retrieval of personal experiences, (2004), ACM Press.
26. P. Terenziani and R. T. Snodgrass: Reconciling Point-Based and Interval-Based Semantics in Temporal Relational Databases: A Treatment of the Telic/Atelic Distinction, IEEE Transactions in Knowledge and Data Engineering 5(16), May (2004).
27. R. Jain, P. Kim, and Z. Li, “Experiential meeting system”, Workshop on Experiential telepresence, Berkeley, California, (2003), ACM Press.
28. Technical Standardization Committee on AV & IT Storage Systems and Equipment, “Exchangeable image file format for digital still cameras: EXIF”, Version 2.2, (2002), URL: http://www.exif.org/specifications.html
29. R. M. Lerner, “At the forge: working with activeRecord”, Linux J., Nr. 140, (2005).
30. D. Thomas, D. H. Hansson, L. Breedt, M. Clark, J. D. Davidson, J. Gehtland, and A. Schwarz, “Agile Web Development with Rails”, Pragmatic Bookshelf, (2006).
31. W3C, “Web Services Description Language”, March (2001), URL http://www.w3.org/TR/wsdl/

32. W3C, “Synchronized Multimedia Integration Language”, August (2001), URL http://www.w3.org/TR/smil20/
33. W3C, “Scalable Vector Graphics”, October (2004), URL http://www.w3.org/TR/2004/WD-SVG12-20040510/
34. Adobe Systems, “Flash Professional 8”, (2007), URL http://www.macromedia.com/software/flash/flashpro/
35. J. R. Smith and S.-F. Chang, “VisualSEEk: A fully automated content-based image query system”, MULTIMEDIA, (1996), ACM Press.
36. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years”, IEEE Trans. on PAMI, Vol. 22, (2000), ACM Press.
37. M. Davis and R. Sarvas, “Mobile Media Metadata for Mobile Imaging”, International Conference on Multimedia and Expo, Special Session on Mobile Imaging in Taipei, Taiwan, IEEE Computer Society Press, (2004).
38. A. Graham et al, “Time as Essence for photo browsing through personal digital libraries”, JCDL, (2002).
39. B. Gong, U. Westermann, S. Agaram, and R. Jain, “Event Discovery in Multimedia Reconnaissance Data Using Spatio-Temporal Clustering”, AAAI Workshop on Event Extraction and Synthesis, (2006).
40. K. Munroe, B. Ludäscher, and Y. Papakonstantinou, “BBQ: Blended Browsing and Querying of XML in a Lazy Mediator System”, Int. Conf. on Extending Database Technology, March (2005).
41. A. Scherp, “A Component Framework for Personalized Multimedia Applications”, OlWIR, Oldenburg, Germany, February (2007). PhD Thesis, http://ansgarscherp.net/dissertation/
42. A. Scherp and R. Jain, “Towards an ecosystem for semantics”, Workshop on multimedia information retrieval on The many faces of multimedia semantics, Augsburg, Bavaria, Germany, (2007), ACM Press.
43. A. Scherp and S. Boll, “Paving the Last Mile for Multi-Channel Multimedia Presentation Generation”, Multi Media Modeling, Y.-P. P. Chen (ed.), January (2005), IEEE Computer Society.
44. A. Scherp and S. Boll, “MM4U - A framework for creating personalized multimedia content”, In: Managing Multimedia Semantics, S. Nepal and U. Srinivasan (ed.), (2005), Idea Group Publishing.
45. A. Scherp and S. Boll, “Context-driven smart authoring of multimedia content with *x*SMART”, ACM Int. Conf. on Multimedia, (2005), ACM Press.
46. S. Boll and U. Westermann, “MediÆther: an event space for context-aware multimedia experiences”, Workshop on Experiential telepresence, Berkeley, California, (2003).