MediÆther — an Event Space for Context-Aware Multimedia Experiences

Susanne Boll
University of Oldenburg
D-26135 Oldenburg, Germany
boll@informatik.uni-oldenburg.de

Utz Westermann
University of Vienna
A-1010 Vienna, Austria
gerd-utz.westermann@univie.ac.at

ABSTRACT

Today, more and more events of interest all over the world such as press conferences, rock concerts, soccer matches, breaking news, etc. become available on the Internet as multimedia content like pictures, video and audio streams, or web pages. These events are bound not only to a certain topic but also to their location and time. The questions is how recent multimedia events from all over the world can be made available and "experienceable" to people considering their different personal contexts such as their location and interests. Not least because the content on today's WWW infrastructure is not explicitly published along with contextual information such as time and location, the information access via the big search engines — apart from well-known deficiencies handling non-textual content — must remain context-insensitive. In this paper, we present the MediÆther multimedia event space, a decentralized peer-to-peer infrastructure that allows to publish, to find and to be notified about multimedia events of interest. The paper defines an event model along which multimedia events, their location, time and media data are published in the space. Thereby, MediÆther provides a suitable basis for multimedia applications such as personalized multimedia sports and news tickers or multimedia city tours that consider user interests and context like time and location to create a "personal experience."

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; H.3.4 [**Systems and Software**]: User profiles and alert services; H.5.1 [**Multimedia Information Systems**]

General Terms

Human Factors, Design.

Keywords

multimedia event distribution and notification, peer-to-peer.

ETP'03, November 7, 2003, Berkeley, California, USA.

1. INTRODUCTION

Events of public or personal interest happen all over the world at all times during the day. This can be the press conference at which the mayor of New York reports on the power shortage of the north-eastern part of the US but also Lance Armstrong crossing the finishing line of the Tour de France in Paris or the opening of the new Leopold Museum in Vienna. Today, such events are increasingly recorded and made publicly available on the Internet as multimedia content in form of video streams, audio streams, digital images, web-pages, etc., created with different devices by different people at many different places at all times during the day.

And the world wants to take part in these events! In the same way in which the events happen distributed in time and place, people located at different places want to consume multimedia content covering these events, probably even at different points in time. To make this possible, it is important to permit people to efficiently become informed of and to find content covering events that are of relevance to them among the media flood lashing down on people today on the Internet. Relevance of content is not only decided by people's personal interest: a big share of events covered by multimedia content is also of value for a limited duration of time or a geographical region only. For example, a potpourri of last weekend's goals scored in the German soccer league will probably not be able to preserve the interest of the German people far beyond the following weekend and will certainly not receive much attention from the US. Similarly, a cultural event like the opening of the Leopold Museum in Vienna is certainly more of interest to Austrians than it is to Americans.

Consequently, this calls for a distributed publishing infrastructure for multimedia content on the Internet that permits users to find and to become aware of contents created at different times and locations under consideration of the users' individual contexts including time, interests, and location. However, what today's Internet offers to users as support for finding interesting multimedia content is mainly limited to the big search engines like Google [2] and distributed file sharing infrastructures like Gnutella [1]. Both kinds of systems have in common that they do not permit an adequate lookup of multimedia content such as videos, images, and audios but instead are bound to text — the text contained in web pages as in the case of search engines or filenames as in the case of file sharing infrastructures. Also, search engines and file sharing infrastructures do not reflect contextual information for the content like its relevance with regard to time and location. Finally, they do not provide

a proper notification when new contents of interest become available. Instead, users have to be actively and continuously on the search for new contents. Although notification systems are known on the Internet, they typically constitute limited services like e-mail newsletters such as Springer Link [14] that are offered by single information providers with a restricted scope and usually do not cover the temporal and spatial context of content.

In this paper, we propose the MediÆther multimedia event space as a generic, peer-to-peer-based publishing infrastructure for multimedia content that permits to find and to become aware of content dealing with events of their interest in a more appropriate manner. In the distributed space spanned by the MediÆther, content providers can place descriptions of events of interest which are covered by their content. The modeling of events not only comprises metadata about the event itself such as its topic, the time and location when and where the event takes place. It also comprises a link to the actual media data that covers the event along with extensive, RDF-based metadata describing the media data. The MediÆther allows to efficiently search for events along these characteristics: events can be retrieved that cover a certain topic, are close by, or happened recently. Moreover, the MediÆther is able to accept notification requests along the same characteristics and to asynchronously inform applications and users if new events satisfying these requests have become available within the space.

Thereby, the MediÆther infrastructure constitutes a suitable foundation for a wide range of multimedia applications. Different content providers could set up dedicated MediÆthers for news, sports, culture, etc. On top of these spaces, a variety of context-aware multimedia applications could then be realized such as personalized multimedia sports and news tickers, personal multimedia cultural event guides, or location-based city guide which bring multimedia information regarding cultural events, buildings, etc. close by to the tourist's position onto his or her smart phone.

The remainder of this paper is organized as follows: Section 2 introduces the general idea of the MediÆther before Section 3 introduces the notion of a multimedia event in this work and the underlying data model. Section 4 presents the design and realization of the multimedia event space. The application Section 6 illustrates how different application can use the MediÆther space for context-aware services such as a multimedia event finder and a personalized multimedia sports ticker. Section 7 addresses the related approaches in the field before the paper is concluded in Section 8.

2. GENERAL IDEA OF THE MEDIÆTHER

The general idea behind the MediÆther is to provide a "multimedia event space" for the Internet, an infrastructure supporting the realization of context-aware applications by allowing to publish, find, and be notified about distributed multimedia content that covers events of interest like soccer games, world records, war news, pop concerts, weather reports, cd releases, etc. along with their spatial and temporal context.

This idea is illustrated in Figure 1. The space, indicated by the grey circle alluding outer space in the figure, is a logically central but physically distributed place into which multimedia events are entered and from which multimedia events can be retrieved. Each single multimedia event is represented in the figure by a small circle. The figure shows the example of a multimedia news space. On the right, the figure shows some examples of publishers that insert events into the space. For example, a sports news agency might enter the victory of Lance Armstrong in the Tour de France in 2003 with a video showing him cycling up the Champs Elysees. A weather channel may enter the latest weather news with a video showing the weather at a certain place. A news agency can use it to publish the results of the coalition negotiations in Austria with a picture going with is. Besides the actual media data covering the event such multimedia events carry descriptive information about time and location such that the retrieval of multimedia events of interest can be sensitive to topic, time, and location.

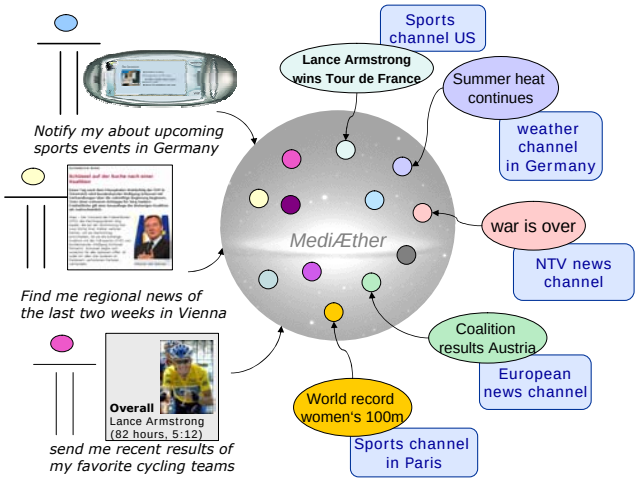

Figure 1: General idea of the MediÆther

When events enter the MediÆther, they are matched against a set of notification request that have been previously registered by a user (application) to become notified when new events of interest become available. In addition to this notification, the multimedia events entering the space remain there until they reach their specific expiration date to be retrieved and selected by interested users.

To the left of the figure, we show some examples of user applications that could be built on top of the example multimedia news space. There might be a handheld application that delivers multimedia content covering upcoming German sports events onto one's handheld. There might also be a conventional web application realizing a multimedia news service covering Austrian news not more than two weeks old or a multimedia e-mail newsletter delivering the results of the favorite cycling team directly into the user's mailbox.

There are two important issues that we do not address with the MediÆther idea: first, the MediÆther is not intended as a multimedia database or archive (though it would be possible to (mis-)use it that way). As a MediÆther will have to cope with large numbers of multimedia events created in tight intervals, speed and scalability are crucial factors that require a main memory implementation not primarily intending a long-term preservation of an ever-increasing number of multimedia events as well as a limitation of the expressiveness of the querying and notification facilities, more than it might desirable for the realization of a multimedia database. Events can expire as the event space is primarily

aiming at distribution and notification of events of interest and not for storing and archiving event. One could connect the space to a multimedia database which archives the events for long-term usage and more sophisticated analysis.

Second, it is important to note that we do not strive to create one single global event space with the MediÆther idea that is to be used by all applications – this would be infeasible regarding the vast amounts of contents perpetually created on the Internet. Rather, we want to provide a generic infrastructure that can be used to easily set up domain-specific multimedia event spaces for publishing distributed multimedia contents.

3. MULTIMEDIA EVENTS

Within the MediÆther infrastructure, multimedia events are the central information unit which needs to be defined. Before presenting the actual data model for multimedia events, however, we will define the notion of a multimedia event in our research context, as the notion of an event can be seen in different ways and our multimedia event type is a very specific one.

3.1 The notion of multimedia events

The notion of an event can be found in different fields of computer science: in active databases, in event notification systems, in programming languages. Typically, an event in these fields is seen as a state change, and it carries a message that describes the state change to an application. Even though we use the term "event" in this work, we rather use it in a sense as it is used in everyday language, a cultural event, a political event, a family event.

So we define: *A multimedia event describes the availability of multimedia content that covers something of importance in a given domain that happens at or during a certain time at a certain place.* It is important to note that we see an event as "capturable by media", so it is strongly related to the media data that actually covers the event. A multimedia event such as the victory of Lance Armstrong of the Tour de France in 2003 comes in one piece along with the actual video showing the Lance Armstrong when he cycles towards the finishing line in Paris. Note that the definition of an event is kept simple as we do not want to anticipate its later usage.

As motivated above, temporal and spatial information about an event and its media data are crucial aspects for our multimedia event space. Consequently, we want to capture where and when this event takes place. So a multimedia event stores also that Lance Armstrong is cycling up the Champs Elysees in Paris in France, at the last day of the Tour de France in 2003, published Sunday July 29, 2003 at 11:18 AM. Besides such important spatial and temporal information, multimedia events have descriptive information and media information associated with them and further information about the publisher of the event. Hence, one can summarize a multimedia event with the following formula: multimedia event = media data + time + location + publisher + topic + additional descriptive information.

All the information describing an event is called the metadata of the event. The event model gives us the basis for a wide range of application that exploit the metadata of the events and let applications and their users experience multimedia data from all over the world from their different personal perspective. That is, a real life event can be represented by one ore more multimedia events that include media data showing or illustrating the event with its relevant temporal, spatial and topical context. With this event model, it is possible to find multimedia events that took place at some place or that happened during a certain time, were published by certain authors, cover a certain topic, will take place close by in the near future, etc. One can also be informed about interesting things going on without having to actively search for them. By this, applications on top of a MediÆther can create a personal experience of events world-wide and let people become part of events just as if they were there in person. The applications can use the media and meta data of the event and of course combine it with other information sources to actually create this experience.

3.2 The expressiveness of a multimedia event

For the representation of the multimedia events in the space, a suitable data model is needed. As we indicated before, we would like to model the different contextual aspects of an event to be able to allow a very personal, localized, and topic-oriented access and notification. Consequently, one might think of defining very complex multimedia event objects that carry comprehensive information about each multimedia event. However, one has to consider that the MediÆther is intended to provide a large number of distributed user applications with a quick and targeted access to a huge number of multimedia events in the space. When deciding on a data model for events, one design decision was on how many information does an event need to carry such that it suits the task and application domain. However, if this information is not a lightweight bundle but a burden, then expressiveness is bought for lower scalability. As we expect a high number of events and frequent requests, quick response times are an important issue. So we aim at finding a model that achieves a good balance between these two sides of the same coin.

Therefore, we decided to model a multimedia event so that the basic information such as the media data, time, location, and topic are represented explicitly. These aspects form the spine of the MediÆther. We consider this information sufficient to efficiently answer queries for content that include time, location, and topic while at the same avoiding a heavy data model. These features of an event that are essential for a quick and targeted access to the events are modelled explicitly and are not to hidden in a generic data structure. Further descriptive information about the event, however, still can be provided in an generic fashion which gives application access to a more detailed, semantic description of the event and offers flexibility. In the next subsection, we will discuss our design decisions along the Entity-Relation-Diagram in Figure 2 that shows the proposed the multimedia event data model.

3.3 The data model for multimedia events

The central element of the data model as shown in the ER diagram is the entity multimedia event that, as its name suggests, represents multimedia events. To make events distinguishable from each other in an event space, a Universal Unique Identifier (UUID) is provided for each event with the attribute eventUUID. The reason why we use a UUID for the identification of events is that these can be easily and uniquely generated in a distributed environment. We also use UUIDs for the modelling of the so-called multime-

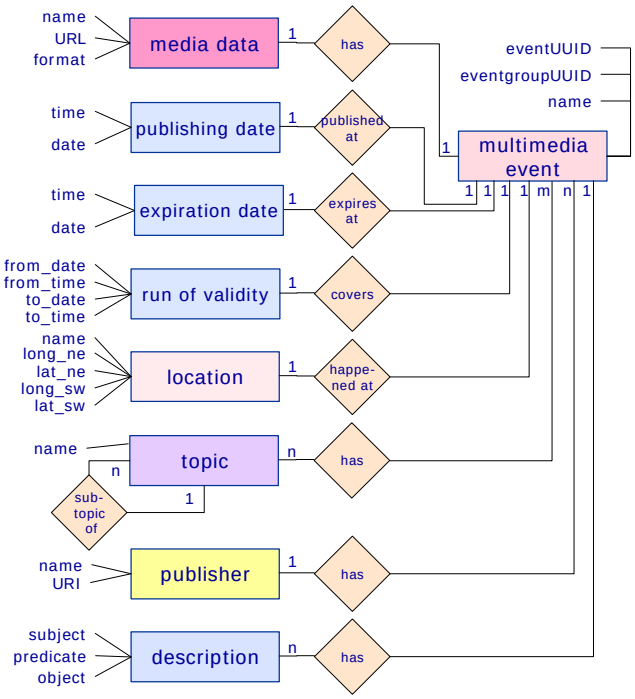

Figure 2: Data model for multimedia events

dia event group to which an event belongs, as represented by the attribute eventgroupUUID. We will come back to this attribute in Section 3.4 addressing relationships between different events. Another attribute is the name of an event which provides a human readable description.

A multimedia event is further described by its publisher, represented by the entity publisher to the bottom left of figure. The publisher, characterized by a human readable name and an URI for identification, forms the description of the person or organization that entered the event to the system. This information can be used as a means to establish a certain trust into the quality of multimedia events.

The left side of the ER diagram also provides the more complex metadata associated with an event. We now present entities covering the media, temporal, spatial, and topic dimensions of a multimedia event before we come to further descriptive entities:

The media dimension: The entity media data at the top left of the figure addresses the media data which covers a given multimedia event. The media data is referred to by an URL, can be given a name, and its type and format can be described. Using URLs for referencing media data, media data at different locations can be accessed over different protocols like HTTP or FTP and even streaming protocols such as RTSP. We associate only one medium per event to keep the event model simple. To cope with different media covering the same event, the already-mentioned event groups can be used.

The temporal dimension: To the left, the diagram shows the entities that are related to time and that put a multimedia event into a temporal context. First, there is the actual publishing date of the event, i.e., the point in time when an event entered the MediÆther. The expiration date can be of sense for multimedia event spaces that want to have a natural circulation of events. For example, sport news events in a space that need not to archive events could keep them only for a certain time like two weeks. Events that expire are removed from the space, an archiving functionality can be achieved for example by a database coupled to the space. The run of validity can be used to assign the duration in time that an event covers. With this entity, the duration which the multimedia event covers is decoupled from the current date and time and also from the date and time the event has been created. A new album of a pop star can be published (and by this be announced) at a certain date even though the CD will be available on the market only from a later date on which forms the beginning of the event's run of validity. In the example of Lance Armstrong, this means that the event could have a creation date `Sunday July 29, 2003 at 11:18 MET`, that is the date and time when it is published. The expiration date, however, might be set to a month after the publication date to `Sunday August 31, 2003 at 12:00 MET`. And the run of validity could be set to `July 5th, 2003, 10:00 MET to July 27th, 2003, 16:00 MET`, the duration of the Tour de France in 2003.

The spatial dimension: The location entity is used to describe the place where a multimedia event happens to put it into a spatial context. In our model this is a 2D geographic area. This is the simplest representation that allows a connection of the location to any other geo-referenced material. We model the location as a rectangular region specified by two coordinates defining each latitude and longitude of the north-east and south-west corner of a rectangular area. So for example, the location of the victory of Lance Armstrong in Paris could be associated with the geographical location of the finishing line, a new exhibition in the Leopold Museum in Vienna could be associated with the location of the museum building.

The topic dimension: As a simple form of semantic description, a multimedia event can be associated with one or more topics, e.g., "sports" or "politics". Topics are organized hierarchically thereby allowing to realize a domain-dependent taxonomy of multimedia events. The bandwidth of the description of events by topic depends on the taxonomy provided by the domain.

For a better understanding of the data model as introduced so far, we present an instance of a multimedia event in Figure 3. The data structure indicated in the figure holds the event that Lance Armstrong wins the Tour de France in Paris. Following the different entities of the ER diagram, the event instance describes the different aspects of multimedia event.

The topics of an event are an admittedly simple structure, however, a fast means to access a large set of events by hierarchically organized terms of a domain. Apart from these topics, a second kind of generic, semantic description of a multimedia event can be provided on the basis of RDF graphs using the remaining entity description. The three attributes of this entity indicate that it represents RDF statements being tuples consisting of subject, predicate, and object. A multimedia event can be associated with a set of

```
multimedia event: {
    uuid_event:       b1027a86-c062-4dfb-a33e-b564bc2e7219
    uuid_eventgroup:  b1027a86-d062-61fb-a33e-b464baa34219
    name:             Lance Armstrongs wins Tour de France
    media data:       VictoryImage,
                      http://.../lance_wins.jpg,
                      JPEG
    creation_date:    Sunday July 29, 2003 at 11:18 MET
    expiration date:  Sunday August 31, 2003 at 12:00 MET
    run of validity:  July 5th, 2003,
                      10:00 MET to July 27th, 2003,16:00 MET
    location:         Paris, France;
                      long_nw 1234, lat_nw 4567
                      long_se 1245, lat_se 4545
    topics:           sports, cycling, outdoor, men
    publisher:        SportNews Europe, http://www.eusportnews.com
    description:      ... }
```

Figure 3: Instance of a multimedia event

such descriptive tuples, which make statements with relevance to the event and follow a domain-specific ontology previously defined with languages such as DAML+OIL or RDF schema. By this, for example, one can express that an event is a live stream, that an event is about the Tour de France, that is has two major persons associated with it, that each of the two cycle for certain teams and so on, as indicated by Figure 4.

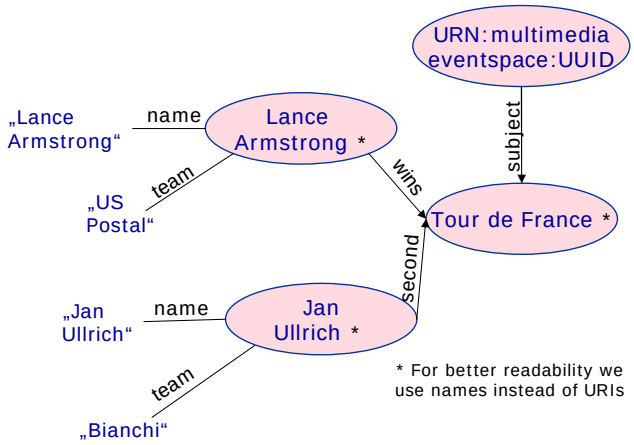

Figure 4: Instance of the RDF description of a multimedia event

Instead of explicitly taking account of RDF statements in the event model, one could also think of simply referring to a file by a URL that carries the RDF statements describing the multimedia event. But then this metadata would be residing in a kind of black box outside the scope of the MediÆther and the space could not consider these RDF statements to find or notify users of events of interest.

Given the fact that the model permits multimedia events to be described by arbitrary RDF statements, the question comes up why not all the other dimensions of multimedia events that we have explicitly captured in the model are modelled as RDF statements. We think that modelling everything in RDF is not an option as RDF is an expressive model which also requires complex query mechanisms — which contradicts our aim of providing a lean data model permitting an efficient implementation for even large numbers of events. If the multimedia event model explicitly represents the most relevant of an event's facets like time, location, and topic, in contrast, it is relatively easy to provide highly optimized access to events along these dimensions.

The description of events with RDF statements in the model is primarily intended as a means to provide applications with additional information about the event but not to be used by the MediÆther for event retrieval and notification in the first place. No inference or more complex traversals on RDF graphs is done by the space. This would lead efficient search for and notification of events in large spaces into the devil's kitchen.

The data model should not be seen as either complete or perfect. This is the model we developed to support a multimedia event space for a wide range of applications with a simple event model such that the space can perform well.

3.4 Multimedia events and their relationships to other events

In our approach, we model simple, unrelated multimedia events: temporal, causal, or structural relationships between event as, e.g., addressed in [12], are not made explicit in the model. The reason for this is again that of simplicity and efficiency: assuming that the proposed multimedia events space is populated by a huge set of events, an efficient distribution of events and a quick retrieval and notification are essential which would become difficult when the space would have to take account of relationships between events.

One could thus claim that knowledge about events and their interrelationships which is available at publishing of a set of events like all the vacation pictures taken in Rome is given away by our event model. However, somewhat contributing to a modeling of relationships between events, our model provides the event group attribute. Multimedia events that are related to each other in reality can be assigned the same event group. For example, all multimedia events belonging to the same soccer match could belong to the same event group. But the model does not make explicit how the events of a group are interrelated. For that purpose, one could employ the RDF-based description part of events to model relationships between events with an RDF graph outside the scope of the model.

4. THE MULTIMEDIA EVENT SPACE

The MediÆther logically forms a single place where the events go that are entered into the system, where events expire and are removed from the space, where events reside and "wait" until they are found, and where new events are evaluated against notification requests such that users and applications can be notified if there are any new events of interest. Though the space seems to be logically centralized it is a physically distributed set of interconnected nodes that each forms an entry point to the space. These nodes are organized in a peer-to-peer network. Each peer provides interfaces for publishing, searching, and registering notification requests for multimedia events. Each peer distributes published events to other peers and receives events published at other peers as well.

This realization of the MediÆther infrastructure as a distributed peer-to-peer network is a tribute to the fact that

multimedia contents treating events of interest are continuously created at many different places all over the world.

Figure 5 illustrates the general organization of MediÆther as a peer-to-peer network. The example MediÆther given in the figure, is realized by four interconnected peers. Each of the single peers of the MediÆther holds its own local multimedia event space as indicated by the large circle at each peer. Currently in the example, each local space is filled by a set of multimedia events. The multimedia events published at one peer are distributed among the local spaces according to the peer-to-peer network topology such that every space hold the same events, i.e., the multimedia events are replicated at each peer. The multimedia events that reside in the four spaces in the example differ slightly to indicate that this can happen due to propagation of events through the peer-to-peer network and some events might have not been received yet. Even though propagation of events takes time, replicated event spaces at the different peers make the system robust and also permit large numbers of queries and notification request as these can be completely handled locally by the peers. The alternative would be different events and different notification requests at different peers. In consequence, to satisfy a user's request, one would run into distributed query answering and distributed notification issues in a potentially unreliable peer-to-peer network. If the space is replicated, the worst thing that can happen is that a user might miss some events, that some events might be received "late", or a peer is out of work. An application thus accesses the MediÆther always by using the interfaces for event publication, querying, and the registration of notification request offered by the closest peer. The distribution of published events among the different peers of the network is left to the underlying peer-to-peer infrastructure, JXTA [15] in our case.

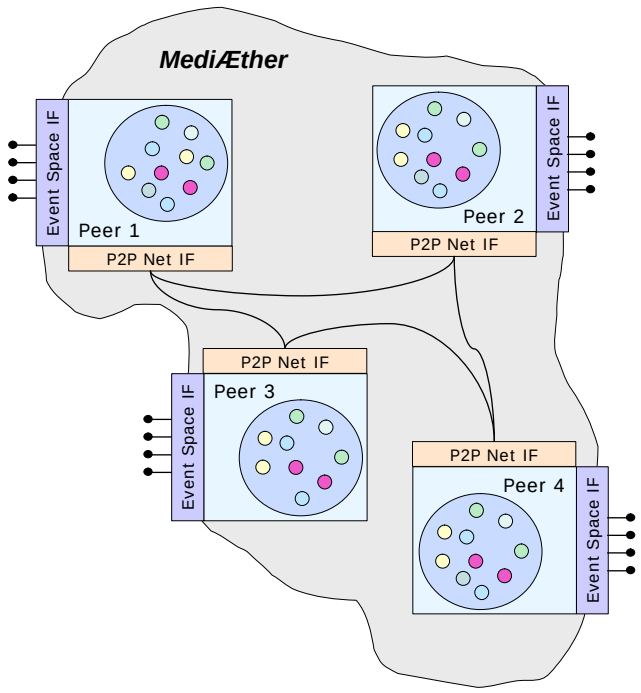

Figure 5: The peer to peer based MediÆther

This discussion of the design and the functionality of the multimedia event space is refined in the following sections. In Subsection 4.1, we explain how each of the single multimedia event spaces stores and efficiently accesses the multimedia events. In the following Subsections 4.2 and 4.3, we present in detail how the events are published, retrieved, and notified.

4.1 Organization of multimedia events in the local space

Each peer in the MediÆther implements the data model for multimedia events that we have presented previously. It keeps all events contained in its local space, i.e., those published locally or received from other peers, in main memory thoroughly indexed along the different dimensions that are of interest for the querying and filtering of multimedia events to permit efficient access to and notification among even large numbers of events.

Event organization for retrieving multimedia events

As a basic means for access to the events of the local space, the UUIDs of events and the event groups to which they belong are indexed with hash tables to support quick lookup by equality. This is typically exploited by applications to quickly access events that they have retrieved or been notified of.

For more sophisticated access to multimedia events, it is important to have efficient access to events via their most relevant characteristics, namely time, location, and topic. Consequently, suitable indexes are imposed on these dimensions. To index the different temporal dimensions of events, we employ R-tree index structures that not only permit efficient access to events that happen at a certain point in time but also to events that happened during a certain duration. Such R-trees are maintained not only for the run of validity of multimedia events but also for their publishing and expiration date. The publishing date of an event is likely to be of interest for many applications while an indexing of the expiration date is useful to permit an efficient removal of expired events.

The same applies for the spatial dimension of events: we use an R-tree to index the location of multimedia events to permit an efficient retrieval of events that happen in or that overlap with a certain geographical region.

For the last of the three central dimensions of an event, namely its topics, we use the domain dependent taxonomy of topics as an index to efficiently access events along it. The taxonomy is realized as a tree in main memory with its nodes representing the individual topics and subtopics. Each event that covers a certain topic is referenced by the respective tree node.

Figure 6 illustrates the organization of events and indexes for querying events in the space. It shows one instance of a multimedia event and indicates the event's different dimensions that are associated with the respective indexes.

Finally, each peer additionally indexes the RDF-based description part of multimedia events to permit an efficient though admittedly simple access to multimedia events via this aspect as well. In fact, each element of an descriptive RDF statement is indexed by a dedicated hash table: i.e., there is one hash table each for the the subject, the predicate, and the object of an RDF statement. Concerning the statement "Lance Armstrong wins the Tour de France", the hash table for the subject would associate the

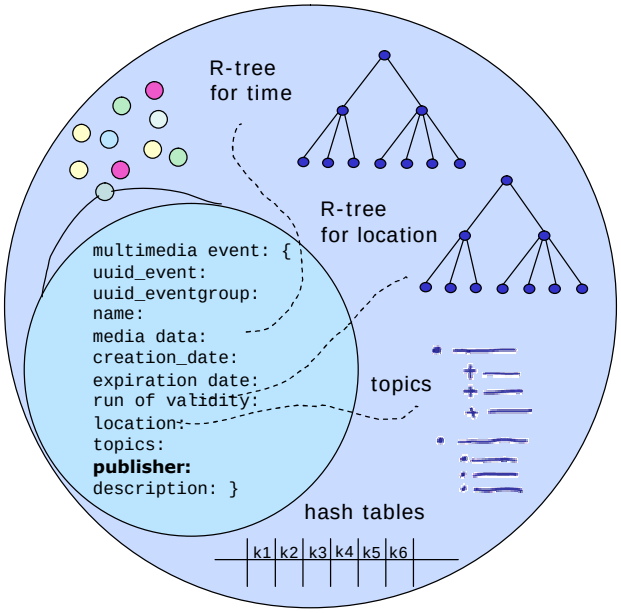

Figure 6: Organization of multimedia events for retrieval

resource "Lance Armstrong" with every multimedia event in the space that is described by a statement with that subject, the hash table for the predicate would associate the resource "wins" with every multimedia event in the space that is described by a statement with that predicate, and the hash table for the object would associate the resource "Tour de France" with every multimedia event in the space that is described by a statement with that object. Accordingly, a peer is able to quickly answer queries for multimedia events which are described by RDF statements with certain subjects, predicates, and objects.

Event organization for event notification

So far, we have described an organization of events that is suitable to efficiently look up multimedia events in the space along the different dimensions of multimedia events. This does not yet apply for the notification of events, however. In the latter case, applications deposit sets of matching criteria forming notification requests with the space and the space notifies the respective requestors whenever matching events have entered the space. For example, an application might place a request asking for events which took place during the last two days in Europe and have something to do with the German soccer team "Borussia Dortmund". But simply checking all notification requests sequentially when a new event enters the space is not efficient if we assume that there might be large numbers of notification requests and events occur in tight intervals. Again, an appropriate indexing is needed to speed up notification.

In our first design, the indexing to support an efficient notification is kept simple: for each dimension of a multimedia event addressed by a matching criterium in the notification request, e.g., the location of an event, the request is associated with the corresponding index covering that dimension for querying. In this case, the request would be inserted into that node of the R-tree indexing the location of multimedia events which suits the requested location. If then a new event enters the space, it is inserted into that node of the same R-tree which suits the event's location. After the insertion of the event, all requests associated with the R-tree node into which the event has been inserted and those associated with the nodes above that node are informed.

This is illustrated by Figure 7. In the figure, a notification request for multimedia events that hold matching criteria for four different dimensions is inserted into the indexes covering these dimensions. Two of these criteria and indexes are depicted: namely, the requestor is interested in events that are related to the topic "Lance Armstrong" and that have a run of validity [c,d]. In case that a multimedia event addressing the topic "Lance Armstrong" is inserted into the space, all requests anchored to that topic in the topic index – especially our example request – and to the topics above that topic are informed. Similarly, if an event with a run of validity [e,f] enters the space, all request associated with the node of the R-tree corresponding to that interval and those requests associated with the nodes above that node – again especially our example request – are informed.

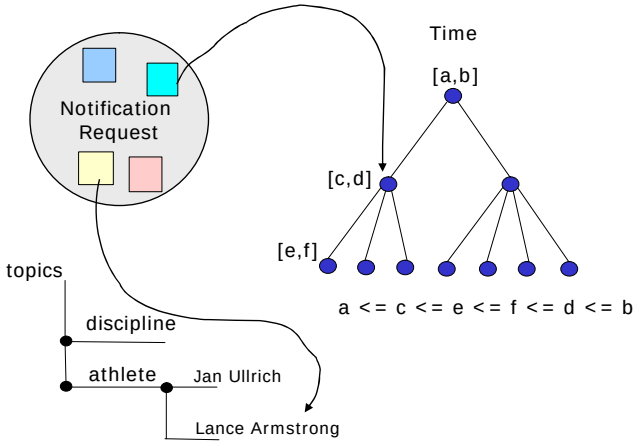

Figure 7: Organization for notification of multimedia events

As all the different criteria of a notification request are anchored with the corresponding indexes, an additional step is necessary to ensure that applications are only notified when all criteria have been fulfilled by a multimedia event. This is done by calculating the intersection of the UUIDs of the events which match the different criteria of a given request.

We are well aware that there are better strategies for an efficient event notification, e.g., [3, 10, 4]. We therefore plan to implement and evaluate an index structure for event notification that is based on based on the theory of finite state automata and has been presented in [3] and refined in [4].

4.2 The publication of events

Each MediÆther peer provides an interface to let applications publish their multimedia events within the space, as indicated on the right of Figure 5. To publish an event, a providing application needs to call that interface at one of the peers, typically the one closest to the application. The event is inserted into the peer's local space and indexed as explained above. Then it is evaluated following the pattern

described previously whether any notification requests at the local peer are matched by the event and the applications having issued these request are notified asynchronously.

For distribution in the peer-to-peer network, the event is then serialized into an XML format and handed over to the underlying peer-to-peer infrastructure which distributes it along the networks's topology. In our case, the underlying JXTA implementation broadcasts the serialized event to the so-called simple and rendevouz peers which are known to the local peer at which the event was originally published: simple peers just receive and treat the event, rendevouz peers additionally transport the event to further known peers in the network. In any case, if a peer receives a message containing a multimedia event, it is inserted into the peer's local space and treated as if it would have been published there locally. The receiving peer also informs the sender by means of a still-alive message that it is still online.

4.3 Searching and registering for events

Multimedia events that have been published in the space can be searched. For this, each peer provides a simple query interface. This interface allows to use the different dimensions of multimedia events to specify the events that are of interest for a query. The interface supports exact-match queries as well as range queries. Different dimensions specified within a query are combined by AND, i.e., all dimensions queried must be matched such that a multimedia event is delivered as a query result.

Applications that would like to query for events use the query interface of the nearest peer. As the multimedia event space is replicated, the query is carried out only at the local space of that peer exploiting the various indexes we have described before.

Similar to the query interface, the MediÆther peers also provide an interface with which applications can deposit notification requests. Again, applications only register with their nearest peer. In case that a notification request is satisfied by a new event at the local peer, the application is asynchronously notified of the event via SMTP.

5. THE PROTOTYPE

We have performed a prototypical implementation of the MediÆther multimedia event space using Java and the JXTA peer-to-peer infrastructure [15]. The MediÆther peers are realized as JXTA peers that implement the local event spaces on the basis of the main memory organization and indexing scheme we have described before in Subsection 4.1 and provide the different interfaces for event publication, distribution, querying, and notification.

To be able to use and demonstrate the prototype independent of a multimedia application on top of it, we have also developed a graphical user interface that gives visual access to the different interfaces of the MediÆther peers. The Figures 8, 9, and 10 give an idea of that GUI.

To publish events, as Figure 8 indicates, a form is provided that allows to enter the different characteristics of a multimedia event. On submission of this data, the multimedia event is inserted into the local space and the distribution in the peer-to-peer network via JXTA is triggered.

To search for events, as Figure 9 indicates, there is a further form with which one can specify the values for the different dimensions characterizing the events of interest. On submission of the query, it is evaluated at the local peer

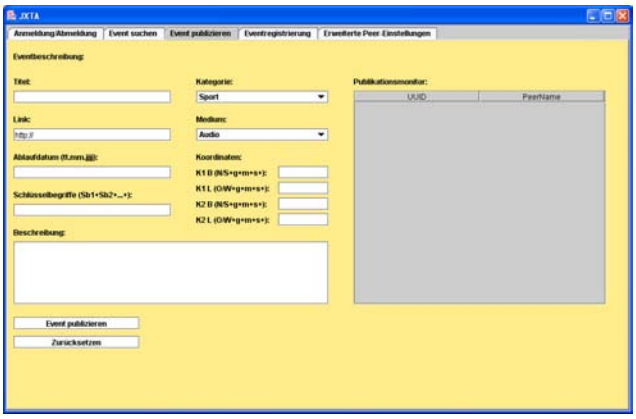

Figure 8: GUI to publish events

and the result events are retrieved and listed in the GUI for further inspection.

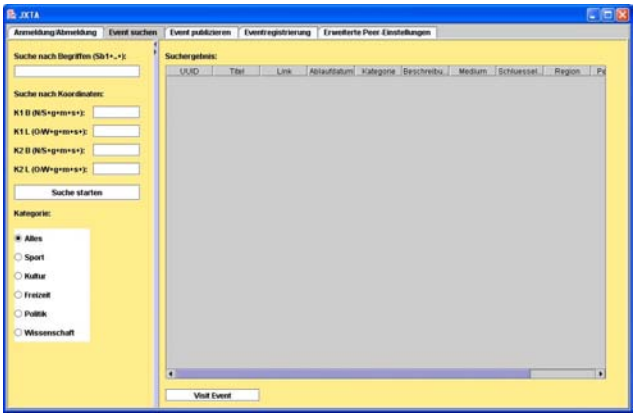

Figure 9: GUI for searching for events

To register for events, as Figure 9 indicates, another form is available with which one can specify the values for the different dimensions that characterize the events one wants to be notified about. One can also specify the e-mail address to which notifications should be delivered via SMTP. Upon submission, the notification request is registered with the local event space and e-mails are sent to the specified e-mail address as soon as qualifying events enter the local space.

6. APPLICATIONS OF THE MULTIMEDIA EVENT SPACE

The prototype we have presented in the previous section only serves as a demonstrator of the MediÆther and its interfaces. The underlying implementation of MediÆther can be used to develop attractive, context-aware multimedia applications on top of it that allow users to be part of these multimedia events which they are interested in and which they like. Two sample applications that we envision and that illustrate the potential of the multimedia event space are a "personalized sports news ticker" and a "personal multimedia city experience" which we present in the following.

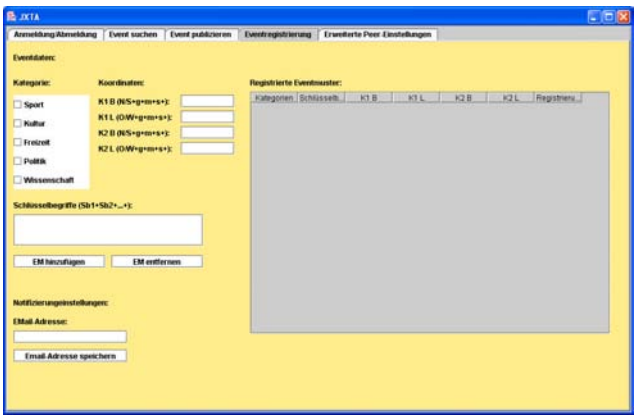

Figure 10: GUI for registering for events

One can imagine a MediÆther specialized on sports that is fed by sports channels with multimedia content covering recent sports events using a common sports taxonomy. One possible application on top of this sports MediÆther could be a *personalized sports news ticker* that allows people to be up-to-date with their favorite sports teams world-wide. Users specify their interests, favorite teams, regional preferences and so on with a client application. The sports news ticker application maps this to corresponding notification requests for multimedia events and registers them with peers that are part of the sports MediÆther.

In turn, the MediÆther notifies the sports ticker application about new events that match the personal profile. Whereas the demonstrator GUI of our MediÆther prototype only lists the multimedia events, the sports news ticker application actually processes them. From the new sports events it receives, the client application can create a list of recent events of interest to the user and compose them to a multimedia SMIL presentation. In such a presentation, the sports ticker application can combine the multimedia data of the event, the available meta data, and additional information, e.g., from a soccer player database. The result is a multimedia presentation that can be viewed with a SMIL player over the Web.

In this application, the context-awareness of the multimedia event space that considers the individual preferences of a user is the most central aspect in combination with its notification capabilities. A user neither needs to visit different information sources and also does not need to actively search for events of interest but is notified about them in time.

For another group of applications, a further MediÆther could be set up to disseminate multimedia content that covers major cultural events in large European cities. Multimedia concert announcements, exhibitions, and the like are published in the space. A multimedia application that provides personalized city tours to its users can now exploit the space as a search place for up-to-date events in the respective city. So, a *personal multimedia city experience* application can create personalized, location-aware multimedia tours for their users to visit the sightseeing highlights in town. Exploiting the location of the multimedia events in the space in combination with the user preferences about art styles and historical epochs, the personal city tour can be enriched with information about exhibitions close by, concerts of favorite artists happening during the stay, etc. In this type of application, the metadata of the multimedia events in the event space is exploited for the retrieval of the most suitable events and their multimedia data that fit the the personal city tour.

7. RELATED WORK

Related work comes from the fields of modelling multimedia data and context, peer-to-peer networking, and event notification systems. In the field of multimedia data modelling, there are to our knowledge no other approaches that define a notion of multimedia events that is comparable to ours. With regard to our general idea, however, we find the idea of an "Event Web" that has been recently presented in visionary articles [7, 8]. The work presented in this paper could be regarded as a possible technical foundation for the realization of such an Event Web.

For the prototypical realization of the MediÆther infrastructure, we base on SUN's JXTA peer-to-peer networking framework [15]. With regard to the topology of a network of MediÆther peers, we therefore completely rely on the standard JXTA implementation. In further development steps, we want to decouple our implementation from JXTA and evaluate in more detail optimized approaches to using peer-to-peer technology for event notification systems, such as recently presented in [16] that takes the topology of a peer-to-peer network and couples it with the flexible routing of a publish/subscribe system.

The MediÆther constitutes an infrastructure for event notification and retrieval. For the retrieval of events we base on well-known index structures such as R-trees and hash tables for an efficient querying of multimedia events. For notification, we find several suitable index structures in the field of event notification systems [3, 10, 4] which we also plan to evaluate and integrate into future versions of MediÆther.

For the notion of events and their modeling, we have also been looking at the domain of event notification systems. Summarizing, one can say that the central characteristic of events in these system is that an event happens instanteously and describes a state change. An event is regarded as, e.g., an "occurrence of a state transition at a certain point in time" [5] or as an "instantaneous effect of the (normal or abnormal) termination of an invocation of an operation on an object" [13] and is typically modelled as a collection of generic (attribute,value) pairs associated with a timestamp introducing a temporal aspect to the notion of events. In contrast to our multimedia events, however, location is not explicitly modelled although a very recent work takes up on the idea of combining event notification systems and location-based services [6]. And though Luckham [9, 11] discusses broader views of events — i.e., they are modelled as tuples of information containing, e.g., who generated the event, what activity was done with which data values, the time and duration of that activity, etc. — a relationship to multimedia content and its descriptive metadata is not given.

As mentioned before, our view of a multimedia event is more that of an event in a colloquial sense like a sports event or a cultural event. We moreover differ from these event notions in the sense that an event has a duration and is not characterizing a discrete state transition. Additionally,

we do not treat multimedia events simply as generic sets of (attribute,value) pairs but explicitly model the central aspects that characterize a multimedia event such as media data, topic, time, and location.

8. CONCLUSION

In this paper, we have presented the design of the MediÆther, a distributed multimedia event space that allows applications to publish, to find, and to be notified about multimedia content covering events of interest going on world-wide. MediÆther's notion of multimedia events not only comprises the media data covering an event of interest but also its context, i.e., time, location, and topic of the event. By this, user applications can get a very targeted and personalized access to the different multimedia events.

We have proposed a data model for multimedia events, described the overall architecture of the MediÆther as a peer-to-peer infrastructure that suits the distributed nature of publishing and accessing multimedia events, and shown how MediÆther provides efficient querying and notification of events by means of an extensive indexing. We have illustrated a prototypical implementation of the MediÆther on the basis of Java and JXTA and outlined two application scenarios in which the MediÆther can contribute to a personal experience of multimedia events world-wide.

The focus of this paper has been the design of the infrastructure of a distributed multimedia event space. We are currently carrying out performance evaluations along with optimizations concerning the index structures used for the efficient event retrieval and notification as well as the topology of MediÆther peer networks. In this respect, we plan to incorporate existing and evolving techniques from the field of distributed event notification systems to achieve a state which proves that the concept of the MediÆther can go along with a performant realization.

We are further implementing the sports news ticker application on top of the MediÆther. With this application, we give one example of how such a distributed multimedia event space can be a valuable contribution to today's people desire for targeted multimedia information world-wide.

Acknowledgements

We would like to thank Ramesh Jain for the inspiring discussions we had in this research field and Annika Hinze for her valuable comments in the related field of Event Notification Systems.

9. REFERENCES

[1] Gnutella.com, 2003. http://www.gnutella.com/.

[2] Google Technology Inc, 2003. http://www.google.com.

[3] J. Gough and G. Smith. Efficient recognition of events in a distributed system. In *Proceedings of ACSC18*, Adelaide, Australia, 1995.

[4] A. Hinze. An adaptive integrating event notification service. Technical report, Free University of Berlin, Germany, 2003.

[5] A. Hinze and D. Faensen. A unified model of internet scale alerting services. In L. C. K. Hui and D. L. Lee, editors, *Internet Applications, 5th International Computer Science Conference, ICSC'99*, volume 1749 of *LNCS*, pages 284–293, 1999.

[6] A. Hinze and A. Voisard. Combining even notification services and location-based services in tourism. Technical report, Freie Universitaet Berlin, 2003. Technical Report Number tr-b-03-06.

[7] R. Jain. Event web. In *IEEE MultiMedia*, volume 6. April-June 1999.

[8] R. Jain. Experiential computing. In *CACM*, volume 46. July 2003.

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

[10] D. Luckham and J. Vera. Event based concepts and language for system architecture. In *In Proceedings of the Workshop on Studies of Software Design*, May 1993.

[11] D. Luckham and J. Vera. Event based concepts and language for system architecture. In *In Proceedings of the Workshop on Studies of Software Design*, May 1993.

[12] L. Perrochon, W. Mann, S. Kasriel, and D. C. Luckham. Event mining with event processing networks. In N. Zhong and L. Zhou, editors, *Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (PAKDD-99)*, volume 1574 of *LNAI*, pages 474–478, Berlin, Apr. 26–28 1999. Springer.

[13] D. S. Rosenblum and A. L. Wolf. A design framework for Internet-scale event observation and notification. In M. Jazayeri and H. Schauer, editors, *Proceedings of the 6th European conference on Foundations of Software Engineering (ESEC/FSE '97)*, volume 1301 of *LNCS*, pages 344–360. Springer / ACM Press, 1997.

[14] Springer Link, 2003. http://www.springerlink.de, Springer-Verlag Heidelberg, Germany.

[15] Sun Microsystems Inc, 2003. JXTA Project, www.jxta.org.

[16] W. W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, and A. P. Buchmann. A peer-to-peer approach to content-based publish/subscribe. In H.-A. Jacobsen, editor, *In Proceedings of the 2nd International Workshop on Distributed Event-Based Systems (DEBS'03)*. ACM Press, June 2003.