Contextualized Attention Metadata
Hans-Christian Schmitz, Martin Wolpers, Uwe Kirschenmann, Katja Niemann
CAPLE-Group, Fraunhofer FIT
{hans-christian.schmitz, martin.wolpers, uwe.kirschenmann, katja.niemann}@fit.fraunhofer.de

Introduction
Contextualized attention metadata are data about users’ foci of attention and activities. They describe which data objects attract the users’ attention, which actions users perform with these objects and what the use contexts are. Contextualized attention metadata are a prerequisite for generating context-specific user profiles that help to personalize and optimize task and information environments. They can also be employed for annotating data objects with information about their users and usages, thereby rendering possible object classifications according to use frequency, use contexts and user groups. Last but not least, they can be crucial for supporting cooperative work: they can be utilized for monitoring distributed task processing, for identifying critical information and making it mutual knowledge, and for bringing together working groups (Schuff et al. 2007, Hauser et al. 2008, Adomavicius, Tuzhilin 2005, among others).

The following is a scenario for an attention aware system that exploits contextualized attention metadata (Wolpers et al. 2007, Rapp 2006): a lecturer began but then interrupted the design of an online course. Now she resumes her work. In previous interaction, the system generated a task profile. Now it recognises the ongoing task of designing the online course and recovers the working environment accordingly. It presents a task history and opens unfinished documents with the appropriate tools. While working on the course material, the lecturer only wants to receive information related to the course. Consequently, the system temporarily hides emails not relevant to the current task, among others. The lecturer wants to be provided with new learning material that is suitable with respect to the course and the lecturer’s activity profile which was generated in previous sessions. The system therefore automatically generates search queries and acquires relevant information; it accommodates to the lecturer’s preferences, tasks and activities. Furthermore, let it be given that a similar online course has been held before. The learning objects used in this course have been annotated with contextualized attention metadata captured from the students who were enrolled in the course. The lecturer analyses the metadata and finds out how the students used the course and which kinds of learning objects attracted their attention. She derives the employed learning strategies, compares the learning strategies of students who finished the course with a high grade with the strategies of students who finished with a low grade and infers which learning strategies were most effective and should be supported. She uses this information in order to tailor the new course according to the students’ needs and preferences. When the design of the new course is finished and the course is online, the students’ actual usages are monitored. The learning system advises the students of important material that they have not found and used so far. It hints at other students who might help with specific problems, and it helps to set up effective working groups. It actively supports a collaborative learning process.

In the example scenario, contextualized attention metadata are captured from the students’ interactions with the online course and from the lecturer’s usage of application programs for designing the course. The metadata are fused for generating user profiles, which in turn are used for course optimization (evaluation of individual learning strategies) and information retrieval (automatic search for material with respect to the students’ and the lecturer’s
profiles), among others. Moreover, the metadata are used for describing and classifying documents, in particular learning objects (preferences of students). Also, a task profile (designing an online course) is used for adapting the lecturer’s working environment.

In this paper, we first define our basic terminology. Then we describe CAM (Contextualized Attention Metadata) as a formalized digital representation of user attention. We continue by outlining a general framework for capturing, storing and analyzing contextualized attention metadata. Finally, we discuss ways of exploiting these metadata.

**Basic Terminology**

Let us first define our basic terminology: the theoretical term “attention” can refer both to cognitive mechanisms of data selection and to actual data selection behaviour, that is, the attending to-behaviour of agents (Mole forthcoming). We use the term “attention” as referring to attending to-behaviour. Agents can attend to things of diverse types. They can attend to objects – e.g., *Katja attends to the email that she has just received* –, to properties – e.g., *Uwe attends to the style and orthography of a particular paper* –, and to propositions – e.g., *Martin attends to the fact that he has exceeded his disk quota* –, among others. By attending to something, an agent selects this ‘thing’ for further cognitive processing. Agents do not always attend to the same thing, their foci of attention change continuously. Therefore, attention is dynamic.

If we want to detect an agent’s attention, we have to observe her. We cannot observe her attention directly, but we can observe her activities. From her activities we infer which things she attends to. To this end, we presume that certain activities require attention. For example, we presume that an agent who opens a web page and, after a while, clicks on a link, attends to this page. Thus, from the observations of the agent’s activities we conclude that she is (or, has been) in a particular attentional state. This conclusion is reliable although in principle defeasible, since it cannot be excluded that the agent opens the web page and clicks on the link by accident while attending to something entirely different. Furthermore, from our observations we can conclude that the agent attends to the particular web page but we cannot conclude that she attends to its content, style or orthography. Thus, our observations underdetermine the agent’s attention in this respect. In order to infer that she attended to a particular property of the page, we have to incorporate and interpret further observations. For example, it seems reliable to infer that she attended to the content of the web page from the observation that she copied some part of the page and pasted it into another document.

Attention is selection. Selection is not fully defined by the things being selected but also by the set from which these things are being selected. Thus, the actual attention of an agent is defined both by the things the agent currently focuses on and by the available but unfocussed things. In laboratory situations, it can be possible to determine the set of available things. Let it be given that a test subject has to focus her attention on the colour of a particular triangle displayed on a screen. The set of available things comprises all objects on the screen, including their forms (triangle, square, etc.), sizes and colours; the focussed colour of the chosen triangle belongs to this set of available things. In real-world situations, however, it might be (and mostly will be) impossible to determine the set of available things. First, the boundary between availability and unavailability is vague. Consider the case that Uwe is attending to a web page. Which are the available things Uwe selects from – all pages of the particular web site, all pages related by links, or even the entire world wide web? Do parts and properties of web pages also belong to the available ‘things’? These questions cannot be easily answered because we are not provided with a definite criterion for defining the boundary between availability and unavailability. Secondly, the number of available things
can be just too large for practically enumerating them. Therefore, in real-world situations we are mostly unable to exhaustively define the attention of an agent.

An expression (or a sequence of expressions) that specifies an agent’s attention is an attention representation. Attention representations serve purposes. They can, for instance, form the basis for a theory of the cognitive mechanisms of data selection; in market research, they can be used for analysing which products have raised a customer’s interest (Hauser et al. 2008); they can form the basis for detecting the tasks and goals of an agent, for analysing her learning strategies, and so on. On the one hand, there are purposes that do not require fully specified attention representations (including enumerations of all available objects). For some purposes – e.g. for a product recommender system –, it can be sufficient to evaluate only the objects that an agent attended to without taking the alternatives into account. It can also be that information on how often she attended to a certain object and on the chronological order of her attention can be ignored. Thus, it might be sufficient to represent her attention by just naming the set of objects she attended to. On the other hand, some purposes require more than pure attention representations; additional information on the agent, on the activities correlated with her attention – when she attended to a text document: did she read or write it? –, and on further contextual parameters might be required. The cognitive mechanisms of data selection, for example, are most probably context-sensitive; thus, if these mechanisms are to be explored, information on the contexts of actual attending-to behaviour is required.

To conclude so far: There are interesting attention-aware systems that do not require exhaustively specified attention representations. The underdeterminacy of attention therefore need not be a serious problem for the design of an attention aware system. However, for some purposes pure attention representations are not sufficient. To meet these purposes, attention representations have to be enriched, for instance, with further information on the observed agent’s activities and the context in which she is acting.

Our aim is to observe computer users, to record their attentions and to use these recordings for diverse purposes like the detection of their tasks and goals, the generation of attention-based user- and object-profiles, and so on. We record a user’s attention rather on a macro-level than on a micro-level. In our terms, micro-level attention refers to highly dynamic, short-term attending to-behaviour like, for instances, focussing on single words while reading a text. Contrary, macro-level attention refers to a more stable, long term attending to-behaviour like, for instance, attending to a text in its entirety by reading or writing it. For recording user attention, we need tools for capturing observations. Moreover, we need a formalised digital representation to be able to describe, merge, store and process streams of observations, that is, we need an attention metadata format. We want user-observation to be non-obtrusive; we assure non-obtrusiveness by only capturing attention data from the computer applications being used. Consequently, attention metadata as defined by us are at least underdetermining representations of attending to-behaviour of computer users while interacting with application programs. They can also contain further information on the agent’s actions, the contexts of action and on the data objects in question.

Let us take stock: (i) We understand attention as attending to-behaviour. (ii) Attention metadata are data that represent continuously changing attention. That is, attention metadata are representations of attention. (iii) Attention metadata schemata are representation formats. (iv) Our aim is to observe computer-users and record their attentions. We observe activities in which a computer user carries out an action on a data objects like a file or an email message. We call these activities events. An example for an event is ‘user x opens file y’. (v) Our observations depend on our observation instruments. Thus, our observations are restricted, we do not observe everything. (vi) Our observations are represented as attention metadata.

1 For capturing activities indicating micro- and macro-level attention see Atterer 2006.
Records of attention metadata – for short: attention records – need not represent everything that is observed. For example, it might be that it is observed that a user opens, changes, saves and closes a document but only recorded that she attends to the document. In this case, the attention record does not entail the user’s actions. However, it can also be that attention records are more than pure attention representations: beside a user’s attention they can represent her actions, action contexts, and so on. (vii) Only in exceptional cases, an attention record will be a complete representation of an attentional state or a sequence of attentional states. Mostly, attention records will contain the objects that have been in the focus of attention but neither the particular aspects or properties that have been attended to nor the available alternatives which the user did not attend to. Thus, attention records are in this respect incomplete. This, however, need not be a crucial problem for the design of an attention-aware system.

From Attention.XML to Contextualized Attention Metadata (CAM)

Attention.XML (Attention.XML 2004, Çelik 2005) is an early approach to capturing and storing attention metadata, that is, to represent attention. Its conception is based on three premises: first, attention metadata are recorded for single users. Secondly, attention records are bags of data objects that have been in the user’s focus (contrary to sets, bags can contain the same element twice). These objects can be ordered according to the time when they have been in the focus of attention. Thirdly, users receive data objects through diverse channels; the objects are to be sorted according to these channels. For instance, when a user receives messages through a news feed channel and accesses web pages with her browser, then her attention record comprises two bags of objects, one for the news feed and one for the browser.

Attention.XML records are stored as XML-files. The root element group comprises the respective user’s name (title) and a set of channels (blog, feed, site). The channel elements contain item elements that refer to the data objects that have been in the user’s focus. Items have several sub-elements for specifying properties like the respective item’s title, type and GUID and information on its usage like lastRead, duration and followedLinks.

Attention.XML has been criticised as not being able to record a user’s attention in sufficient detail: first, Attention.XML only records data objects but not what the user does with these objects. This is a crucial drawback when complex interactions – for instance, updating a text or manipulating a spreadsheet – are to be recorded or when joint activities of multiple users are to be analysed. Secondly, Attention.XML does not describe use contexts. It neither specifies the sets of objects from which the user selects her foci nor further circumstances of her selection. Therefore, attention metadata cannot be evaluated with respect to specific contexts.

As a consequence, the CAM schema (Contextualized Attention Metadata schema) has been defined as an extension of Attention.XML (Wolpers et al. 2007). The most important extensions focus CAM on actions that occur on data objects. To this end, the following elements are added to a slightly modified version of Attention.XML: each item (that is, each data object) may be involved in several events. Events are associated with a timestamp (datetime) and a description, among others. An event can be associated with an action of a certain type including action related data. For instance, when an email-message is sent, the message is an item (data object) involved in an event with a send-action. Events occur in contexts, and they are part of technical sessions, e.g. a browser session. The CAM-schema is only vaguely specified for context-elements: context-elements can contain arbitrary value- and value type-elements. Within the session-element a session-ID, the IP-address of the user’s computer and further information on the involved users are collected. A complete description
of the CAM schema is given by Wolpers and colleagues (2007). The core elements are depicted in Figure 1, which is of course not a representation of the whole schema:

![Figure 1: core elements of the CAM schema](image)

CAM is developed to describe as many types of attention metadata as possible. CAM follows the Attention.XML approach that attention records contain bags in which items represent data objects (actually, CAM records contain sets instead of bags of items); for each data object, action, context and session related data via events are added. CAM records of a user therefore not only describe the user’s foci of attention but rather her entire computer usage behaviour. The CAM schema is essentially a unified schema for usage metadata: all metadata related to usage behaviour are stored within one structure.

Collections of CAM records can be exploited for generating diverse kinds of profiles like user profiles and object profiles (item profiles). CAM records represent a user’s computer related foci of attention and actions. As such, they instantly constitute profiles of individual users’ computer usage behaviour. These user action profiles can be augmented with other information on the users which is, for instance, extracted from a learning management system as in our example above. Moreover, CAM records of different users can be exploited for generating attention and usage based object profiles. Object profiles make content relationships, usage relationships and social relationships explicit, taking into account advanced social information like information on the role a user has when using the object, with whom the user is collaborating on the object, etc. User and object profiles are entailed by CAM collections; they can be derived by simple data transformations.

The CAM schema is a very rich attention metadata schema and provides powerful means for describing, storing, merging and processing streams of user observation. However, it is still under continuous development. The following issues lead to a revision: first, tasks are not explicitly specified within the original CAM schema. Let it be given that we want to transform CAM records of several users into object profiles: every object that occurs within the CAM records is to be annotated with usage related information about which users used this object in what kinds of circumstances. Object profiles are generated by mere transformations of CAM datasets. It would be an advantage if we were able to relate the objects to the tasks that the users were carrying out when they used the objects and thus to augment the object profiles with task related information. Moreover, it would be an advantage if we were able to transform CAM records into generic task profiles, as suggested in the example given in the introductory section. To this end, we need information on the particular tasks. Either CAM records contain this information directly or they contain pointers to
external task representations. Currently, the CAM schema does neither provide an element for a direct task representation nor for a pointer to such a representation. One way to solve the problem is to introduce a task-element as a further subelement of event. The task-element has to contain a title-, an ID-, a description- and a type-element by which a task can be named, identified, described and categorised with respect to a task-ontology. Note that at this stage it does not matter from where the task-related information is retrieved. Task-related information might be determined by analyzing CAM records; it is then added as a supplement to already existing records. Alternatively, the information might be captured from tools like TaskTracer (Dragunov et al. 2005) which allow users to specify the tasks they are currently working on themselves; it is then inserted into CAM records before further analysis is carried out.

The second issue in the revision of CAM is the semi-structuredness of elements like context, session and action related data. These elements serve as containers for diverse kinds of data. The related data subelement of action can, for instance, contain content data (like keywords) or lists of email-recipients, among others. On the one hand, this is an advantage, since it makes CAM flexible and allows the integration of different kinds of metadata. On the other hand, it forms an obstacle for data exchange and automatic evaluation. A possible solution is to import different metadata schemata for structuring different kinds of metadata. Contents of elements like context, session and action related data are then provided with links to their format definitions.

The third issue is the redundancy of sets of CAM records. Sets of CAM records are redundant, for instance, when semantic information on data objects (like keyword lists) are stored as action related data. When a data object is involved in several events, the semantic information is stored for each event even if the event does not affect the object’s semantic properties. Example: let it be given that Katja opens an email-message and then moves it to a particular folder. The keyword-list of the email-message is stored twice, namely within the related data-subelement of the open-action and within the related data-subelement of the move-action. A solution is to separate event descriptions and object descriptions so that different event descriptions can be related to the same object descriptions without replicating the object descriptions.

A tentative approach to address the above mentioned problems is to redefine the CAM schema as a distributed metadata schema with a flat hierarchy. Core CAM instances are defined as pairs consisting of an ID and a triple \( \langle s,p,o \rangle \), where \( \langle s,p,o \rangle \)-triples describe events like ‘user x opens file y’. That is, \( s \) is a user who performs an action \( p \) on a data object \( o \). The elements \( s \), \( p \) and \( o \) point to other metadata repositories that contain information on the user, the action and the object, respectively. The subject \( s \), for instance, can be a pointer to an FOAF-document (www.foaf-project.org), the predicate \( p \) can point to metadata denoting the application by which the action was carried out, the time when the action was carried out etc., and the object \( o \) can be a pointer to a Dublin Core record of that object (www.dublincore.org), among others. According to this tentative approach, the problem of redundancy is solved, since semantic information on objects etc. is stored independently of event-descriptions. The problems of the semi-structuredness of some elements and the introduction of a new task-element are moved: the definition of the metadata contained in these elements is separated from the CAM core; we can refer to already existing metadata standards.

A General Framework for Capturing, Storing and Analysing CAM

In the previous section we described the CAM schema as a general metadata format for merging, storing and processing user observations. In this section, we outline a general framework and infrastructure for collecting and processing CAM records. Such a framework has to meet the following objectives: attention and usage metadata are generated from as
many applications as possible. Together, these metadata represent a user’s actual computer usage. They are generated continuously as long as the user operates her computer. The data have to be integrated and transformed to a unified representation for which we propose the CAM schema as an adequate format. CAM records are to be stored locally and, possibly, on remote servers or in peer-to-peer networks. Storage must be reliable in order to ensure a most accurate analysis. Furthermore, the metadata represent highly personal data. Therefore, storage and provision of contextualized attention metadata must ensure privacy and security; access should be restricted to parties licensed by the owner (who is the observed user).

It can neither be foreseen which software will be used to store contextualized attention metadata in the long run, nor which network infrastructure will be used – the choice is between client-server and peer-to-peer infrastructures. Moreover, it cannot be foreseen which application programs will be used. Therefore, it must be possible to integrate new applications that generate CAM records into the CAM framework. The CAM framework needs to be extensible in terms of metadata providing applications as well as storage and analysis software. The underlying infrastructure must enable client-server features in parallel to peer-to-peer features; it has to be set up as a hybrid infrastructure. For sustainability – that is, for rendering the adaptation to new use cases, tools and protocols possible – the hybrid infrastructure will make use of standardized protocols that enable the easy extension while reducing limits on newly added software as much as possible. By abstracting the actual storage and transport of metadata away from the metadata wrappers, third parties are enabled to easily provide software that captures attention metadata. The abstraction also enables the development of analysis tools that are unaware of the underlying infrastructure, and therefore simplifies the development and adaptation of such tools.

The CAM framework is depicted in Figure 2. The framework provides the ability to collect, transfer, provide and store observation based attention metadata. Metadata are collected and transferred into the CAM schema by wrappers for application programs that run on the computer (observation providers). The metadata are stored locally on the client side and/or – depending on the user’s approval – remotely in databases of various types (observation consumers and providers). Using a hybrid network infrastructure, attention metadata are accessed from respective analysis tools (observation consumers) that run either locally or
remotely. The following paragraphs will briefly explain the conceptual structure of the framework by outlining its structure and composition.

Observations: Observation data are generated by all applications that the user works with on her computer. Most application programs provide some sort of usage history, either directly as log files or stored in databases. The communication software Skype (www.skype.com), for instance, uses an internal database to store all chat conversations. In order to enable the domain and application independent processing of observation data, these data have to be transferred into a unifying schema, namely the CAM schema. To this end, we use existing wrappers for the file system and application programs – e.g., the ALCOM framework (Verbert et al. 2005) and the User Activity Logger developed at L3S (pas.kbs.uni-hannover.de, Chernov et al. 2007) –, and we develop new wrappers for diverse applications that are able to provide observation data. Wrappers already exist for the file system, the Firefox browser, the Thunderbird email client, the Skype communication tool, Microsoft Office, the Winamp music player, etc.

Storage and provision: All wrappers running on a computer deliver CAM records to a peer application. The peer abstracts the underlying storage and network infrastructure away from the wrappers. It therefore provides an open and extensible framework for the development of wrappers. The peer is responsible for transportation and storage of the CAM streams. It stores all CAM records in a local database. Based on the access rights provided by the user, it can also store observation data on a remote server or make them accessible within a peer-to-peer infrastructure.

A peer is also able to receive observation data from other peers in order to store them in a local database. The security of transferring data is ensured by the user of simple encryption mechanisms like PGP within the network. In order to ensure privacy, and in addition to the respective access restriction mechanisms, observation data are anonymized (unless otherwise explicitly stated by the user) using mechanisms like K-anonymization (Sweeney 2002).

Analysis tools can access the observation data depending on where and by whom they are run. Tools that are run locally by the observed user have full access to all observation data of this user. Furthermore, using the peer application and respective access rights, they can access all observation data stored remotely on a server or within the respective peer-to-peer network.

Architecture: The underlying architecture of the CAM framework is based on a simple set of interface specifications which are all implemented within the peer application. In order to enable a transparent storage mechanism, we rely on the Simple Publishing Interface (SPI, Ternier et al. 2008) for the storage of observations. Using SPI, the peer receives observations from the wrappers and stores them in a local database. Furthermore, using SPI wrapped into a client-server protocol (e.g. SOA-driven using WSDL and SOAP), the peer forwards the observation data to respective remote storage peers. In order to query for observation data, analysis tools use the Simple Query Interface (SQI, Simon et al. 2005) of the peer to send their query to respective storage locations within the network. SQI and SPI provide the advantage of being unaware of the used query and publishing language, thus they enable the easy extension of the network with additional observation providers and consumers.

**Exploitation of CAM**

A CAM database describes in detail the computer related behaviour of one or several users. The instances of the database contain – or, at least, can be related to – additional information on the users (age, profession, ...), the data objects being used (their semantic properties, modalities, ...) and on the contexts of action (time, location, working time/leisure time, ...). Therefore, by querying a CAM database precise behaviour-oriented user profiles can be
generated: what did a particular user do under specific contextual conditions, which kinds of data objects did she use? Usage-based object profiles can be generated as well: by whom has a particular object been used, in what kinds of contexts has the object been used, what has been done with the object? A CAM database gives rise to diverse user- and object-classifications: which users performed certain actions with an over-average frequency, which users attended to objects with certain semantic properties, which objects have been in the focus of a certain user group? Finally, since also communication behaviour can be observed it is possible to deduce propositions about social relationships: who has been in contact with whom about what? A CAM database is a dynamic representation of computer-usage. Therefore, user and object profiles and classifications have a temporal dimension and reflect the evolutions of usages and attentions.

Research is carried out in the further evaluation and interpretation of contextualized attention metadata: first, by classifying the objects a user (repeatedly) refers to, her general preferences regarding contents, modalities etc. can be inferred. A simple, but quite plausible presumption for such a defeasible inference is that a user prefers those kinds of objects that she uses with a high frequency. For instance, from the fact that she often attends to learning videos when also texts are available we infer that she prefers video presentations over plain texts. Since CAM records contain the contexts of attention, preferences can be relativized with respect to specific contexts – the video preference need not be true for all contexts. Inferences on preferences are being improved and made more reliable when explicit information – like object recommendations by the respective user – are taken into account. Since recommending-actions can be recorded, this information can be entailed in a CAM database.

Secondly, CAM records can be used for the detection of competencies. Let it be given that the learning objects of an e-learning system are annotated with information regarding their complexity. These annotations give rise to the previous knowledge that is required to use and understand the learning objects. Thus, the user’s attention to objects gives rise to her (previous) knowledge. Moreover, knowledge and skills are not only proven by the ability to give answers but also by asking the right questions (Ram 1991). Thus, a user’s information search behaviour – which search queries does she pose in which contexts? – seems to be a promising clue to her actual competencies (Hölscher, Strube 2000).

Thirdly, research is going on in the area of cognitive and emotional state recognition. Research so far concentrates on the analysis of speech acts. Systems have been implemented following Weintraub’s (1964) studies in psychological states expressed via language (Shaw 2008) and the Linguistic Category Model (Fiedler 2008, Semin 2008). The systems basically depend on keyword vectors for their analysis: word tokens of different categories are counted, from the word frequencies conclusions regarding the author’s cognitive and emotional state are drawn. The analysis is to be extended to non-verbal symbols like emoticons and, furthermore, from speech acts to other kinds of acts in order to detect significant frequencies of attention shifts and repetitions, among others. Results from the analysis of email and chat messages are used to enrich social network data, thereby to generate fully-fledged diachronic sociograms and to use these sociograms for socially aware systems (Pentland 2005).

Fourthly, we derive action patterns from CAM records which are used for the automatic recognition of users’ tasks, goals and intentions both in single user- and multi users-environments. For task recognition, approaches from algorithmic learning of formal languages are being applied. Atomic actions are treated as symbols over an alphabet, tasks are considered to be sequences of actions. Therefore, the aim is to construct a task grammar that generates tasks as sequences of actions from a given action-alphabet. For the detection of

---

goals and intentions, outcome states and their evaluations are to be taken into consideration as well.

Research in the interpretation of CAM records can be carried out by using the CAM framework as a research instrument. The framework provides the means for observing users in controlled settings and for analysing observation data. We can apply the framework as a tool for investigating the correlations of users’ actions with their preferences, competencies, tasks, and so on. However, the CAM framework is not a research instrument in the first place. Analysis tools are to be extended to real application programs for task and learning support and efficient information retrieval, among others. These programs are designed not only for the controlled, experimental environment but, first and foremost, for real-world application.

One application that makes the individual installation of the CAM framework attractive is a reporting tool that summarizes the user’s actions and gives her an overview on what she did and which data objects she worked with during the day, the previous week or month. Taken the results of analyses into account, she can assess her preferences, competencies and completed as well as ongoing tasks. She can gain an overview on which data she sent to whom (maybe without being aware of it) and conclude what others might know about her. For example, she can record which data were sent to Google as search queries, gmail-messages etc. and thereby appraise her Google-profile. This can be regarded as a way of pre-defence regarding privacy: becoming aware of personal data distribution might lead to a more cautious behaviour in web-based environments.

As a prototype, we implemented a tool for observing, analysing and reporting on a user’s email- and chat-communication: we analyze e-mails that are stored locally in mbox format (which is a file format used e.g. by Thunderbird to store e-mails) or remotely on an imap server. We extract the sender, the receiver(s), the sent date, the subject and keywords from the email message. Keyword lists serve as shallow content representations; they are generated by the use of the yahoo! term extractor\(^3\) and tagthe.net (www.tagthe.net). We use a plug-in for Thunderbird, namely Adapted Dragontalk\(^4\), to generate information about the usage of the e-mail tool, that is, to observe when (and how often) a user opens a particular e-mail and when an e-mail is forwarded, responded, moved or deleted. Moreover, we collect chat data from the Skype communication tool. The communication partners, times and keywords of conversations are extracted as metadata. All metadata are transformed into the CAM format and stored in a local, native XML eXist database (www.exist.org). We analyze both email contacts and chat conversations for creating and visualizing an egocentric social network of the user. The tool allows a user to explore her e-mail and chat archive in a new way: she generates an overview on who talked about what to whom and when, so that she can, for instance, recognize that a specific topic was discussed by different groups of her contacts, maybe at different times. She can evaluate her communication behaviour and recognize whose e-mails she read most often and answered quickly. Furthermore, the use of emoticons is analyzed and depicted as a tentative clue to the evolution of social contacts. The user is provided with reports on the dynamics of her social relationships.

In the introduction of this paper, we described a scenario involving an e-learning system in which usage and attention metadata are not only evaluated locally on the observed user’s computer but also remotely on the server running the e-learning system. In this scenario, CAM records of all students using an online-course are collected, stored and analyzed for the aim of evaluating the learning system and supporting individual learning strategies and

---


\(^4\) The Dragontalk-plugin was developed by DFKI (German Research Center for Artificial Intelligence): dragontalk.opendfki.de (retrieved Nov 24, 2008). The plugin was developed further by the L3S Research Center: pas.kbs.uni-hannover.de/download.html (retrieved Nov 24, 2008).
collaborative learning processes. In such a scenario, CAM records can come from two sources, namely from the individual users’ computers and from server log files. The metadata coming from the different sources are integrated into a large CAM dataset. A first test bed for the collection and exploitation of many users’ contextualized attention metadata has been implemented within the MACE project (portal.mace-project.eu). The aim of MACE (Stefaner et al. 2007) is to improve the access to digital architectural learning resources by setting up a federation of architectural learning repositories: large amounts of architectural contents from distributed sources are integrated and made accessible to architects, architecture students and lecturers. Applying an extension to LOM⁵ (learning objects metadata), the metadata descriptions of architectural learning resources are harvested from a large number of European repositories into a central metadata repository. The harvested metadata are enriched with various types of additional metadata, including content metadata, competence⁶ and learning process metadata (Koper, Tattersall 2005) and contextualized attention metadata. Within MACE, contextualized attention metadata are composed of individual usage related metadata as described above and of metadata required through social interaction – like recommendations by peer users and blog entries. Social interaction of MACE users relies on the ALOE system (aloe-project.de, Memmel et al. 2008) which renders it possible to capture, store and allocate metadata on interactions like joint tagging of learning resources, exchange of bookmarks, object ratings and recommendations. Using the rich set of metadata, very expressive object profiles can be generated which make it possible to offer multiple perspectives on the architectural contents and diverse navigation paths through the contents. Users can find resources by simple keyword search but also with visual navigation tools for browsing through the different classifications of the MACE resources. In addition, MACE offers statistical data that are exploited, among others, for listing the most popular learning resources and for summarizing trend features within a Zeitgeist⁷ application.

A prototype for the use of contextualized attention metadata within MACE has been set up. It is based on two major components, namely a usage metadata repository and a set of usage metadata services. The usage metadata repository stores CAM records captured from different sources. It uses the XML-enabled database IBM DB 2 system so that CAM instances are stored natively without pre-processing. For communication with the outside world, the usage metadata repository offers three interfaces: (i) the Simple Publishing Interface (SPI) for inserting CAM instances into the database – SPI is used by CAM providing sources like the MACE portal, the MACE infrastructure services and the ALOE system –, (ii) the Simple Query Interface (SQI) for querying the repository – SQI is used by the analytical services described below –, and (iii) the Open Archives Initiative Protocol (OAI-PMH) to expose CAM records to a harvester in order to enable processing off site by other parties. Currently, the usage metadata repository stores metadata on the following types of events captured from the MACE infrastructure services:

1. A user requests the metadata of a learning resource.
2. A user searches for a learning resource using keyword search.
3. A user searches for a learning resource using the full text search feature of all repositories integrated into MACE.
4. A user searches for a learning resource using the ‘browse classification’ functionality.
5. A user updates a metadata instance within the MACE store.

6. A user logs into the MACE system via the MACE portal.
7. A user requests a listing of all the users registered on the MACE system.
8. A user searches for a specific user.
9. A user requests information on a specific user.
10. A new user ID is created within the MACE system.
11. A user account is activated after a verification process.
12. A user account is deactivated.

The events are described according to the CAM schema. For each event, at least the user, the involved learning resources, the time and date of the event and the location of the user are recorded.

The prototypical MACE usage metadata service provides statistical analyses for ranking search queries and learning objects: which search queries have been posed most often, which learning objects have been requested most often? Ordered lists of learning resources are generated on demand, the user is enabled to define her own ranking criteria. She can ask for an ordered list of the objects that have been requested most often in general or of the objects that have been requested most often by herself, for example. So far, the following ranking metrics have been implemented:

1. **Number of metadata instance views**: a list of the top-k objects ranked according to the number of views on a defined period of time (e.g. day, week, month, year, since recorded history) is returned. The ranking metric generates two types of lists: one global list integrating the views of all users and learning resources and one that integrates only the learning resources of a particular user.

2. **Number of metadata updates**: a list of the top-k objects ranked according to the number of updates on a defined period of time (e.g. day, week, month, year, since recorded history) is returned. Again, the ranking metric generates two types of lists: one global list integrating the views of all users and learning resources and one that integrates only the learning resources per user.

3. **Timeline of metadata instance usage**: usage timelines are returned. This first *Zeitgeist* implementation shows when which learning resources have been specifically popular. Usage timelines can be used for ranking objects regarding particular time-periods.

The ranking service (as part of the usage metadata service) can be used by any authorized client application. For performance reasons, the ranking service internally uses two databases: a normal (non-embedded) database – in this case PostgreSQL – and a database which is embedded in the particular ranking application – for this implementation we use HSQLDB. The non-embedded database stores the same information on user activities as the usage metadata repository, but according to the relational paradigm. This database is also responsible for the support of statistical services that do not require long calculation times. The usage metadata service has an internal job scheduling system that manages the update of the non-embedded database. By using the OAI protocol, this component can be configured to automatically harvest new CAM instances from the usage metadata repository and insert them into the non-embedded database.

The embedded database is used to store pre-calculated (complex) ranking metrics supported as service features. The database is pre-populated during the web application loading, by referring to the non-embedded database in order to obtain the necessary data for calculating ranking metrics. All calculations from the non-embedded database for a single ranking metric are stored in a single database table. If a ranking service is required, the usage metadata
service uses this table and is thus able to respond requests quickly. Finally, to keep the embedded database up to date, all rankings metrics are re-calculated after an automatic harvest has been done.

The application of CAM within the MACE project has two important outcomes: first, since the usage and attention of users is evaluated and visualized, individual users can reflect to which objects they have attended and to which objects significant numbers of other users have attended. Personal usage histories serve as reminders: users remember the objects that they have attended and that might become relevant again. Statistical evaluations serve as recommendations: users are pointed to objects to which a significant number of other users attended. This makes it possible to recognize trends, to follow those trends or, contrary, to resist those trends and look for objects that have not been in the focus of the general public so far. Secondly, we annotate data objects with usage and attention based metadata. Objects are classified and associated according to their actual usages. These associations are used for improving information retrieval. (Google’s original PageRank algorithm (Brin, Page 1998) only takes explicit associations – that is, hard-wired hyperlinks – into account. We extend those links with dynamic, usage-based associations.) By making explicit when and by whom a particular object has been used for what and which objects have been used together with this object, and by presuming that objects are relevant in their usage contexts, we specify the object’s potential relevance. That is, we contribute to sharpening the concept of relevance and to making it operationable for information retrieval.

Current Web2.0 approaches like Amazon, ALOE, Digg, YouTube and others already demonstrate that information access and retrieval can be personalized by exploiting usage metadata. The application of CAM within the MACE project shows how CAM is able to improve current personalization efforts for information systems. For example, the annotation with attention metadata advances the possibilities of exploratory search and the creation of (individual and community-based) associative nets beyond mere link-based document graphs as in the former HTML-based web.

We chose the MACE system as a test bed for CAM because it contains a very large data repository (needed for generating object profiles), because it has a large number of users (needed for generating user profiles) and because it is used for e-learning, namely in architectural courses at different universities. Especially in architecture, it is important to structure, associate and remember large amounts of contents. Architects use work of their colleagues as an inspirational source, for copying and extending. They need to have access to large amounts of highly diverse information ranging from pictures of buildings, project sketches, reviews to governmental regulations. They have to take diverse perspectives on this material and structure their views according to their actual individual interests. Therefore, the MACE system is ideal for testing and proving the benefits of CAM exploitation.

**Conclusion**

In this paper, we have defined our current understanding of the nature of contextualized attention metadata (CAM) as describing the attending-to behavior of agents, in particular the behavior of users while using digital information on their computer. We have introduced the CAM schema as a general schema to represent contextualized attention metadata computationally. Furthermore, we have described a framework for capturing, storing and analyzing CAM records. The framework collects instances of CAM from application programs and stores them locally and possibly (depending on access rights, among others) on remote servers. We have outlined different ways of exploiting CAM records and thus demonstrated various ways in which the analysis of CAM instances can be beneficial for users. Finally, we have introduced two demonstrators for the exploitation of CAM records,
namely a tool for observing, analyzing and reporting on a user’s email- and chat-communication and the MACE system as a test bed for using CAM in a distributed, multi-user environment. While the first application focuses on the single user, the e-learning scenario of MACE enables us to pursue research questions in multi-user CAM scenarios. Currently, a number of projects are being set up in this area as further test beds for the development and application of CAM.

At this point, we have intentionally left out one major aspect of CAM, namely privacy and security: contextualized attention metadata are personal data that should not be at everybody’s disposal. The usage and distribution of these data must be under the control of the observed user. The first method to keep the data under control is to set up the CAM framework only on a local computer and to refrain from distributing CAM records over a network. If CAM records are distributed over a network – like in an e-learning scenario – anonymization techniques have to be applied. Our simple approach, implemented in our framework, will be to let the user control who has access to her CAM records and how these records will be distributed.

Furthermore, we have not yet dealt with the issue of the controlled exchange of usage and attention metadata. One such technique might be APML (attention profiling mark-up language, www.apml.org) which is designed as a format for exchanging attention profiles. APML profiles can be automatically generated, but they can also be edited by their owners. That is, users can add information or delete information from their profiles before distribution. This open issue will be dealt with in the near future.

References


