Self-monitoring for Computer Users

Maren Scheffel, Martin Friedrich, Marco Jahn, Uwe Kirschenmann, Katja Niemann,
Hans-Christian Schmitz, Martin Wolpers

Fraunhofer Institute for Applied Information Technology FIT
Schloss Birlinghoven
53754 Sankt Augustin, Germany
{maren.scheffel, martin.friedrich, marco.jahn, uwe.kirschenmann, katja.niemann,
hans-christian.schmitz, martin.wolpers}@fit.fraunhofer.de

Abstract: We are presenting a tool for collecting and analysing computer usage data. The observed data are locally used by the user to self-monitor and self-reflect her behaviour, decontrolling the data for personalisation of information environments only with her consent. [1]

1 Introduction

Over the years the amount of available data, applications and the number of computer and internet users has increased continuously which, on the one hand, enhances the possibility to find the information needed and to contact other people. On the other hand, it can just as well complicate these actions due to information overflow. This is why recommendation systems are needed, helping the user by presenting search results in an adequate order, suggesting objects and tools for effective data processing and pointing to other users with whom contact might be valuable. These recommendation systems have to be user and context sensitive. For a system to generate such user and context representations, data about the user and her behaviour have to be permanently collected, stored and analysed. As the collection of data must not disturb the user in any way, the observation has to run in the background. The collected data also have to be true, representing the actual behaviour of a user.

Here, a dilemma occurs: in order for a recommendation system to be successful it requires very detailed observations of user behaviour. Only then can it be user and context sensitive and, thus, truly useful. This, however, might highly discomfit the user, especially when she has no control over the observations and cannot retrace how they are used for characterisation and when she fears that things get out of hand and that her data

[1] The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 231396 (ROLE project) and from the European Community's eContent+ Programme under grant agreement ECP-2005-EDU-038098 (MACE project).

get used wherever for whatever by whomever.[2] The only way to solve this dilemma is to let the user herself control all observations.

The observation tool we are continuously constructing, therefore stores the collected data in a local data base which is only accessible to the user herself. She has full control over the tool which permanently observes her behaviour. By querying the local database the user can retrace her recorded behaviour and get it analysed. She can reconstruct what data she – maybe unknowingly – sent to others and what impression about her others might have gotten. Self-monitoring her behaviour gives the user the opportunity to reflect on her actions and adjust them accordingly. Where necessary, she can also allow controlled access to her data by external tools and services.

The rest of this article is structured as follows: first, we outline the design of our recording tool. Then we present some example components and, finally, we will conclude and list further work to be done.

2 Architecture

The tracing tool we designed is called CAMera: “CAM” because its design is based on the *Contextualized Attention Metadata* (CAM) schema for representing user actions ([Wo07], [Sc09]) and “camera” because, like a camera, it can record actions and events. The requirements mentioned above determine the architecture of the tool: metadata[3] must be collected continuously and be stored, ready for further analysis. The user must not be disturbed in her actions while being recorded, thus, the tool must not make use of obtrusive sensors. CAMera reports have to be reliable and, therefore, must not be based on defeasible interpretations of the user’s actions. The observations must also not be too fine-grained as the representations of observed and reported actions have to be meaningful to the user.[4] Finally, the collected data must be stored locally, accessible only to the user.

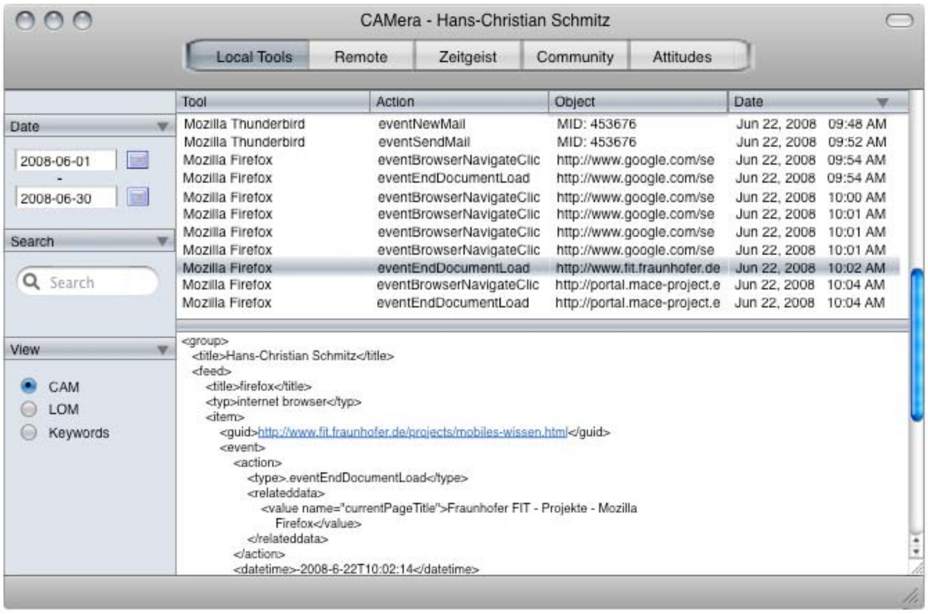
The CAMera tool is made up of different parts: it consists of a set of metadata collectors which collect usage metadata from application programs in order to then transfer these data into the CAM schema. Although these collectors sometimes have to be implemented as proper monitor components that, instead of collecting, generate metadata, it usually suffices to transfer existing log data into CAM. At present, we possess metadata collectors for the Thunderbird email-client, the Skype chat-messenger,

[2] The animadversion on data collection by Google can be taken as evidence for this fear (see f.i. [Rg08] and [Rl08]).

[3] Metadata are data about data; usage metadata are data about actions rather than data in the narrower sense. One reason to call these data *metadata* nevertheless is that they have been called metadata in the literature. (We do not see the need of changing the tradition.) Moreover, such metadata can be used to describe the actual usage of data objects (this is not in the focus of this paper, see [Sc09] instead). As such, they are data about data.

[4] Recordings of mouse gestures or single keystrokes, for example, are not immediately interpretable as they can hardly be put into context. A user’s interaction with application programs, such as opening or modifying a document, however, is immediately interpretable.

the Firefox browser, MS Outlook, the file system[5], MS Power Point, MS Word[6] and the flash meeting system ([FlMe]). Thus, though the set of collectors is still to be extended, we are already provided with quite a large amount of metadata collectors to be made use of and to be experimented with. In order to give a user full control over the collected data, she can decide which collectors are running on her system at what time. The CAMera tool also consists of a database where the generated CAM instances are stored. We are experimenting with different types of databases, relational as well as xml-native ones, particularly the eXist-database ([ExDb]).



Date

2008-06-01 - 2008-06-30

View

● CAM
○ LOM
○ Keywords

Tool	Action	Object	Date
Mozilla Thunderbird	eventNewMail	MID: 453676	Jun 22, 2008 09:48 AM
Mozilla Thunderbird	eventSendMail	MID: 453676	Jun 22, 2008 09:52 AM
Mozilla Firefox	eventBrowserNavigateClic	http://www.google.com/se	Jun 22, 2008 09:54 AM
Mozilla Firefox	eventEndDocumentLoad	http://www.google.com/se	Jun 22, 2008 09:54 AM
Mozilla Firefox	eventBrowserNavigateClic	http://www.google.com/se	Jun 22, 2008 10:00 AM
Mozilla Firefox	eventBrowserNavigateClic	http://www.google.com/se	Jun 22, 2008 10:01 AM
Mozilla Firefox	eventBrowserNavigateClic	http://www.google.com/se	Jun 22, 2008 10:01 AM
Mozilla Firefox	eventBrowserNavigateClic	http://www.google.com/se	Jun 22, 2008 10:01 AM
Mozilla Firefox	eventEndDocumentLoad	http://www.fit.fraunhofer.de	Jun 22, 2008 10:02 AM
Mozilla Firefox	eventBrowserNavigateClic	http://portal.mace-project.e	Jun 22, 2008 10:04 AM
Mozilla Firefox	eventEndDocumentLoad	http://portal.mace-project.e	Jun 22, 2008 10:04 AM

```
<group>
  <title>Hans-Christian Schmitz</title>
  <feed>
    <title>firefox</title>
    <typ>internet browser</typ>
    <item>
      <guid>http://www.fit.fraunhofer.de/projects/mobiles-wissen.html</guid>
      <event>
        <action>
          <type>.eventEndDocumentLoad</type>
          <relateddata>
            <value name="currentPageTitle">Fraunhofer FIT - Projekte - Mozilla
              Firefox</value>
          </relateddata>
        </action>
        <datetime>-2008-6-22T10:02:14</datetime>
```

Figure 1: CAMera’s interface displaying recorded CAM within a selected time interval

The database is extended with an interface for the generation of clearly arranged usage reports; users can review their activities and sort them according to various criteria. Figure 1 shows that interface. The “Local Tools” tab lists all recordings for the selected time interval, here one month. They can be sorted by tool, action, object or date. When one specific entry is highlighted, its CAM schema is displayed. Finally, CAMera consists of analysis applications for the evaluation of CAM instances, e.g. in order to detect the network of people a user communicated with or the most heavily used objects over a certain time span. To work with the different analysis applications the user can click on one of the tabs at the top of the interface, leading her to the specific analysis interface (e.g. the representation of a social network in Figure 2).

[5] For recording access to the file system we adapted the User Activity Logger developed at L3S (Leibniz Universität Hannover).

[6] These collectors are provided by the ALOCOM Framework ([Ve05], [ArAl]).

3 Analysis

In the next sub-sections we introduce example CAMera analysis components. Two different types of behaviour are analysed: communication and browsing behaviour, each again divided into two categories. One communication component exclusively monitors and analyses a user's email-exchange (the 'Community' tab), the other one records and analyses chat messages (the 'Attitudes' tab). The browsing components are *Zeitgeist* components for statistically evaluating browsing behaviour. The first of these components locally analyses browser usage (the "Zeitgeist" tab) while the second component is a remote component (the "Remote" tab) that monitors and analyses interactions with the MACE system for architectural learning ([Mace]).

3.1 Communication

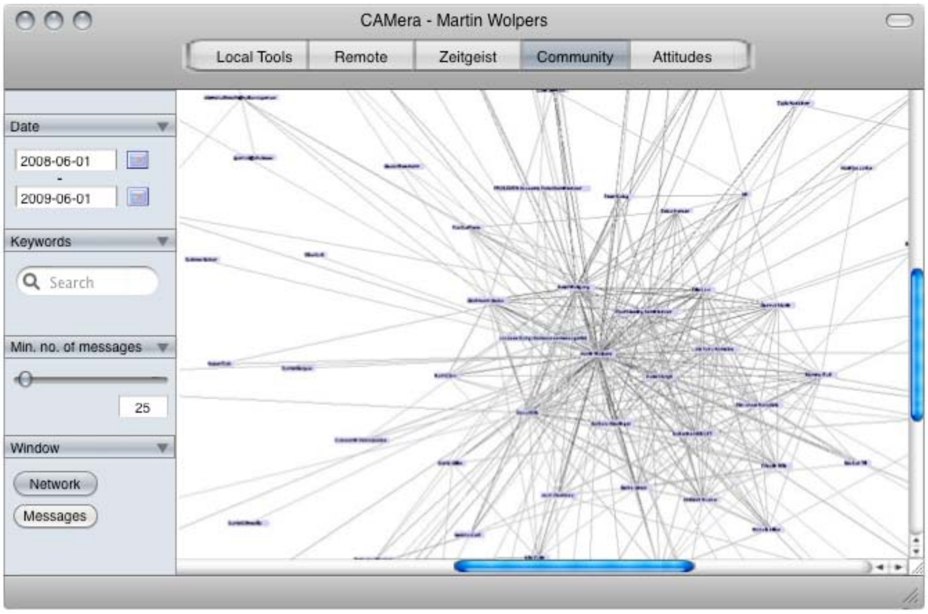
CAMera's email-component has two data collectors to record email-exchange and an analyser to generate and depict social networks. The collectors can be used together or separately. The first collector analyses email-messages stored locally on the user's computer in *mbox*-format or retrieved from an IMAP server. A CAM instance is generated and stored for each message. Sender, receivers, subject line and message body are extracted during the analysis with *Java Mail* ([JaMa]) and a shallow content representation of the message body is generated from extracted keywords. Currently, keyword extraction is carried out with the *yahoo! term extractor* ([YaTe]) and *tagthe.net* ([TagN]).[7] The user chooses whether one or both extractors are used and can decide which messages are analysed by specifying a time interval or by explicitly freeing or blocking email-folders. The second collector continuously records any interaction of the user with a Mozilla tool and is based on the plug-in *Adapted Dragontalk* ([AdDr]).[8] In our case, it records all events involving Thunderbird, such as creating a new folder, opening a message or moving it to another folder. We adapted the original plug-in, which generated usage metadata and wrote them into simple text files, so that for each event a CAM instance is generated and then stored in a database (*adapted Adapted Dragontalk*).

Email-related CAM instances are evaluated by the email-analyser to display a user's social network. Every person occurring as sender or recipient of a message is represented by a node within the network. Iff two persons are involved in the same message, their nodes are connected. The more messages two persons are jointly involved in, the stronger the connection between their respective nodes is. Figure 2 shows the representation of a user's social network within the CAMera tool. The displayed network only depicts the connections to people with whom the user has exchanged at least 25 emails within the time frame chosen by the user. Via the interface the user can browse and manipulate the network: marking a person's node results in a list of all messages that

[7] Ideally, the email-analysis only runs locally on the user's computer. The usage of the *yahoo! term extractor* and the *tagthe.net*-service for keyword-extraction demands data transfer to external services. This can only be a preliminary solution.

[8] *Adapted Dragontalk* (L3S, Leibniz Universität Hannover) is a further development of *Dragontalk* which was developed at DFKI Kaiserslautern ([Epos]).

person was involved in being displayed together with keywords of these messages. It is also possible to only look at specific time intervals or messages connected to specific keywords. With the email-analyser a user can gain insight into the structure of her social network as it gives an account on a specific type of communication behaviour and supports the user in reflecting on it.



2008-06-01 - 2009-06-01

25

Figure 2: Representation of a user's social network with the CAMera tool

CAMera's chat-component records chat conversations that are conducted with Skype, creating a searchable and analysable archive of chat conversations. We are interested in exploring which hypotheses can be deduced from this data about the emotional and cognitive states of conversation partners, their relationship to each other and the communication situation including current psychological theories of communication. According to Walther's Social Information Processing-theory ([Wa92]), for instance, people use technological media and means to construct and develop personal relationships. One thing we are evaluating so far is the use of emoticons. We calculate their occurrences and frequencies in conversations, display change in emoticon usage over time and relate this usage to different communication partners. There appear to be significant dependencies between chat partners, points of time and emoticon frequencies. Additionally, further non-verbal characteristic data, such as *chronemics*[9], are collected and put into CAM instances. Whereas we are, at the moment, not able to deduce reliable

[9] *Chronemics* is a paraphrase for length and duration of conversations and latency of answers that can be informative when describing the dimensions of intimacy/sympathy and dominance/submissiveness between communication partners ([Wa95]).

hypotheses about the user and her social relationships from the assessed statistics, the CAMera user herself can. She can explain differences in emoticon usage from her knowledge of the situation and be inspired by the retrospective report to reflect her own behaviour and, if she thinks it necessary, to adjust her behaviour.

We are working on the integrated evaluation of email-, chat- and other communication-data for generating social networks enriched with information on discourse topics and emotions in communication.

3.2 Browsing

The CAMera tool can also record the user's behaviour when interacting with the Firefox internet browser. All of her browsing actions are observed and the data is then locally stored for later access and analysis. Via CAMera's interface the user can retrace all of her steps and see which web pages she accessed, which data she sent (e.g. to search engines), and so on. She can reflect on her behaviour and, if necessary, adjust or change it to improve her desired outcomes. The component includes a *Zeitgeist* application for statistically evaluating browsing activities and detecting individual trends in web usage. Based on *Zeitgeist* analyses, the tool is not only able to report past behaviour but also to recommend certain actions or certain objects to the user, referring to past actions taken or objects used, making it easier for the user not to get lost within her own system.

The second browsing component is another *Zeitgeist* application, implemented as part of the MACE system ([Mace], [St07]). MACE sets up a federation of architectural learning repositories with large amounts of architectural contents enriched with various types of metadata. Interactions with this system are recorded and evaluated, therefore users are given the possibility to reconstruct their paths of action by retracing which resources they accessed, how they found them and which topics have been of interest to them, thus fostering reflection on their activities. As a result, the user can access analyses of her interaction behaviour. In addition to the analyses provided by her local browser component, the MACE *Zeitgeist* application can cumulate and analyse usage metadata of different MACE users, thus presenting an overview of all MACE-related activities as well as general trends in MACE usage. A user can thereby compare her usage to that of the mass of MACE users, follow trends or refrain from them and find new ways of exploring contents. With the possibility of comparing users on the basis of their usage profile, the *Zeitgeist* component can determine and point to similar users which might be good cooperation partners for her. The MACE *Zeitgeist* application provides a web-based interface but can also be accessed as a remote service from the locally installed CAMera tool.

4 Conclusion

We have explained the dilemma that on the one hand, data of usage behaviour have to be collected and evaluated in order to guarantee the usability of a steadily expanding information space, but that due to this, on the other hand, the alleged or actual risk of a big brother arises. Our solution to this problem is not to abstain from collecting usage behaviour data but to make the collection and evaluation of such data transparent and put it under the control of the individual user. She can then decide what information about her and her usage will be disclosed.

We presented the CAMera tool as a means to self-monitoring and self-reflection. Due to the tool's design, it is always work in progress as it has to be successively extended with further data collectors and analysers, for local as well as remote applications (e.g. other chat clients, online blogging or community tools etc). The main challenge at the moment, however, is to evaluate the already observed data of different sources (i.e. from different application programs) in reference to each other and to thus arrive at a more comprehensive analysis of user behaviour, such as context, task or workflow detection that could then be used for automatic application starts or further system recommendations.

References

[AdDr] Adapted Dragontalk, http://www.l3s.de/~chernov/pas/Documentation/Dragontalk/thunderbird_documentation

[ArAl] Ariadne ALOCOM Tools, http://www.ariadne-eu.org/index.php?option=com_content&task=view&id=65&Itemid=96

[Epos] Epos – Evolving Personal to Organizational Knowledge Spaces, http://www3.dfki.uni-kl.de/epos

[ExDb] eXist Open Source Native XML Database, http://exist.sourceforge.net

[FlMe] The Flashmeeting Project, http://flashmeeting.open.ac.uk

[JaMa] JavaMail API, http://java.sun.com/products/javamail

[Mace] MACE – Metadata for Architectural Contents in Europe, http://portal.mace-project.eu

[Rg08] Reischl, G.: Die Google-Falle. Die unkontrollierte Weltmacht im Internet. Ueberreuter, Wien, 2008.

[Rl08] Reppesgaard, L.: Das Google-Imperium. Murmann, Hamburg, 2008.

[Sc09] Schmitz, H.-C., Kirschenmann, U., Niemann, K., Wolpers, M.: Contextualized Attention Metadata. In (Roda, C., ed.): Human Attention in Digital Environments. CUP, Cambridge, 2009.

[St07] Stefaner, M., Dalla Vecchia, E., Condotta, M., Wolpers, M., Specht, M., Apelt, S., Duval, E.: MACE – Enriching Architectural Learning Objects for Experience Multiplication. In (Duval, E., Klamma, R., Wolpers, M., eds.): Creating New Learning Experiences on a Global Scale. Proceedings of the Second European Conference on Technology Enhanced Learning. Berlin, Heidelberg, 2007; pp. 322-336

[TagN] tagthe.net, http://tagthe.net

[Ve05] Verbert, K., Jovanovic, J., Gasevic, D., Duval, E.: Repurposing Learning Object Components. In: Proceedings of OTM 2005 Workshop on Ontologies, Semantics and E-Learning. Agia Napa, Cyprus, 2005; pp. 1169-1178

[Wa92] Walther, J. B.: Interpersonal Effects. In: Computer-Mediated Interaction: A Relational Perspective. Communication Research 19(1), 1992; pp. 52-90

[Wa95] Walther, J. B.: Nonverbal Cues in Computer-Mediated Communication, and the Effect of Chronemics on Relational Communication. In: Journal of Organizational Computing 5(4), 1995; pp. 355-378

[Wo07] Wolpers, M., Najjar, J., Verbert, K., Duval, E.: Tracking Actual Usage: the Attention Metadata Approach. In: Educational Technology & Society 10 (3), 2007; pp. 106-121

[YaTe] Yahoo! Developer Network: Term Extraction Documentation, http://developer.yahoo.com/search/content/V1/termExtraction.html