CAMera for PLE

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

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

Abstract. Successful self-regulated learning in a personalized learning environment (PLE) requires self-monitoring of the learner and reflection of learning behaviour. We introduce a tool called CAMera for monitoring and reporting on learning behaviour and thus for supporting learning reflection. The tool collects usage metadata from diverse application programs, stores these metadata as Contextualized Attention Metadata (CAM) and makes them accessible to the learner for recapitulating her learning activities. Usage metadata can be captured both locally on the user's computer and remotely from a server. We introduce two ways of exploiting CAM, namely the analysis of email-messages stored locally on a user's computer and the derivation of patterns and trends in the usage of the MACE system for architectural learning.[1]

Keywords: self-regulated learning, personalized learning environments, monitoring, usage metadata, learning reflection, social networks, *Zeitgeist.*

1 Introduction

The core idea of this paper is that self-regulated learning is especially promising regarding positive learning-outcomes, that it demands self-monitoring by the learner and that therefore the learner is to be supported in monitoring and reflecting her learning activities. We present a tool called CAMera for monitoring and reporting on user actions, thus fostering learning process reflection.

The outline of the paper is as follows: in section 2, we argue that self-monitoring is an integral part of self-regulated learning. In computer-based learning environments, self-monitoring can be supported by a monitoring tool that records a user's actions within the learning environment, generates reports on her computer-related activities and helps to recapitulate her learning paths. In section 3, we outline the design of a the CAMera monitoring-tool and present two example components; one for the observation and analysis of email-exchange and one for the observation and analysis of interactions with the MACE system.

[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).

U. Cress, V. Dimitrova, and M. Specht (Eds.): EC-TEL 2009, LNCS 5794, pp. 507–520, 2009.

2 Self-monitoring of Learning Activities

Self-regulated learning is a promising way to successfully achieve learning goals and thus highly eligible with both solo and collaborative learning processes. It will be shown that self-monitoring plays an essential role when successfully learning in a self-regulated way and that computer-based self-regulated learning demands personal learning environments. Within such personalized learning environments self-monitoring has to be supported by recording the interaction of a learner with the actually used tools and services in order for her to later analyze and evaluate her learning processes.

2.1 Self-regulated Learning and Monitoring

The term *self-regulated learning* denotes a learning process where the learner herself decides on what to learn, when and how. Self-regulatedness is a gradable feature that does not exclude guidance by a teacher as long as this guidance does not question the autonomy of the learner. Self-regulated learners are able to meta-cognitively assess and strategically plan, monitor and evaluate their learning activities.

Over the years, self-regulated learning has been a focus for research within educational psychology. Torrano and González ([1])[2] give an account of current and future directions of self-regulated learning: the self-regulated learner can be characterized as a person who actively participates in learning – on a meta-cognitive, motivational and behavioural level –, motivates herself and makes use of strategies in order to achieve the desired results. According to Pintrich's learner model, the process of self-regulation comprises four phases: planning, self-monitoring, control and evaluation ([8]). These phases, in turn, are composed of a cognitive, a motivational/affective, a behavioural and a contextual area. Zimmerman ([9]) also accentuates the need for feedback, especially the self-oriented type, when learning in a self-regulated way.[3] His loop of self-regulated learning comprises forethought, performance and self-reflection ([11]). By meta-cognitively assessing, analyzing and evaluating her behaviour, a learner can adjust her learning processes and consequently achieve better results. Thus, self-reflection is not only useful but an essential part of self-regulated learning.

In general, research agrees that self-regulated learning including self-monitoring and self-evaluation supports successful acquisition of academic (e.g. [12]) and non-academic skills (e.g. [11]) and a better understanding of the things studied. This is covered by experimental evidence: when girls learning how to throw darts were split into groups of students with present or absent self-evaluation (among other groups), those students who did not self-evaluate showed a tendency of attributing poor training outcomes to a lack of ability or insufficient effort ([11]). Those students, however, who did self-evaluate attributed poor outcomes to improper strategy use and practice. These results imply that self-evaluation and -monitoring lead to higher levels of self-efficacy and motivation to learn.

[2] According to them, the five most relevant publications on self-regulated learning are: [2], [3], [4], [5] and [6]. See also [7].

[3] For a detailed and extensive account on feedback and self-regulated learning see [10].

Other experiments were concerned with primary school children who had to solve the Tower of Hanoi problem with three and four discs ([13]). After setting up different training conditions (children watching themselves trying to solve the problem, children watching another child solving the problem inefficiently and children watching another child solving the problem efficiently) results show that those children watching themselves perform better than the others. With four other training conditions (video of self solving the problem inefficiently, video of self solving the problem inefficiently in a prescribed way, video of self solving the problem efficiently in a prescribed way and video of another child solving the problem in the most efficient way), the results, again, show that watching oneself produces better results than watching others. It also turns out to be much more efficient to watch non-prescribed, spontaneous, actual performance as that group achieves the best results.

Self-monitoring is also a motivator ([14]). When dividing students into different groups, either self-evaluating their learning processes or being externally evaluated while acquiring mathematical skills, the prediction is that self-monitoring is more effective than external monitoring. This hypothesis is supported by the results of Schunk ([14]). It can be concluded that self-monitoring fosters the students' motivation to learn. Monitoring their behaviour helps them to become aware of their actions and regulate them accordingly. There are two criteria that are important when recording and self-monitoring one's behaviour: regularity and proximity. Only if these criteria are fulfilled, do students get a thorough impression of their actions and are able to react according to the goal they set.

Tracing features of studying thus has most use as it is done during the action ([15]): by making use of computers to trace the learner's behaviour, the learner is free in her actions and not disturbed or interfered with by other recording measures, such as think-alouds, nor can she forget to mention things as might be possible in post-test questionnaires. Gress et al. show that with the possibility of real-time analysis of everything a learner does and produces while studying (chats, documents, file system interaction) it is easier for her to understand learning processes ([16]). They also stress that this holds true for solo as well as collaborative processes. Making real-time feedback available to the learner enhances the learning process as she can recapitulate what she has done.

To conclude: self-monitoring is an essential part of self-regulated learning. Environments for self-regulated learning have to provide the learner with means to monitor and evaluate her learning activities whether they are connected to solo or collaborative learning processes. Such environments therefore have to record the learner's activities and make the recordings accessible for analysis and recapitulation.

2.2 Personalized Learning Environments

Within a personalized learning environment (PLE)[4] the learner can control all learning processes. She can choose from a vast amount of services and use them how she thinks is best for her thus facilitating the process of self-regulated learning. Apart from the personal space provided by a PLE, the social context of learning is also covered by enabling the connection between several personal spaces, thus supporting collaborative learning ([19]). A PLE is a prerequisite for computer-based

[4] For definitions and characterizations of PLEs see [17] and [18].

self-regulated learning as it is able to record the learner's activities within the environment and to analyse these activities according to the individual learner's needs and choices regarding used tools as well as learning strategies. (We presume that from observations of PLE-interactions conclusions on actual *learning* behaviour can be drawn, at least by the learner herself who has sufficient background knowledge to retrospectively interpret her computer related actions in the light of her learning goals and activities.)

An important aspect of computer-based PLEs is the option for a learner to not only choose from a given number of services but to bring her own tools and services into the environment and even share them with other users collaboratively. A learner should be able to use those tools and services she is familiar with from her every-day interaction with the computer making the PLE a resource for studying and learning as well as working and using it in her spare time. With the PLE thus being both a task and information environment it is suited for academic and non-academic lifelong learning.

The question about what tools and services are actually being used in PLEs is not easily answered. For browser statistics, several sources can be looked at: according to w3schools.com ([20]) the usage of browsers depends significantly on the user group. Technical affine people tend to not use the Microsoft Internet Explorer, but alternative browsers like Mozilla Firefox. A second source are the access statistics of Wikipedia ([21]): here, the most often used browser is the Internet Explorer (66.82%), followed by Mozilla Firefox (22.05%), Safari (8.23%), Google Chrome (1.23%) and Opera (0.70%). Fingerprint ([22]) offers usage statistics for e-mail clients: it shows that most often Outlook is used (36%), followed by Mozilla Thunderbird (2.4%). Additionally it is pointed out that many users do not install e-mail clients on their computers but use web interfaces like the ones from Hotmail (33%) or Yahoo! Mail (14%).

It is relatively simple to capture information about the usage of browsers and e-mail clients, but not so easy to gather information about the usage of other tools a user is executing. The statistics from Wakoopa provide such information ([23]). Wakoopa is a software application that runs locally on a user's computer and tracks the usage of all other applications and sends this information to a server. Even if only specific user groups install such a tracker, they still offer an overview over used tools. The most used Instant Messaging Applications under Windows are the Windows Live Messenger, Skype and the Yahoo! Messenger. The most used office applications under Windows are Microsoft Office Word, Microsoft Office Excel, Microsoft Office PowerPoint, Adobe Reader and OpenOffice.org ([23]). Other frequently used software applications tracked by Wakoopa comprise e.g. World of Warcraft, Adobe Photoshop, iTunes, Microsoft Visual Studio, Windows Media Player and VCL media player.

The Centre for Learning & Performance Technologies annually compiles lists of the most used learning tools and services: one for learners and one for learning professionals ([24]). Contribution is open to anyone via an online spreadsheet. 54 learners and 35 learning professionals have participated until 19 April 2009. The top five tools for learners are Google Search, YouTube, Firefox, Twitter and, sharing the fifth position, Wikipedia and Delicious. For learning professionals Twitter, Delicious, Google Reader, Skype and PowerPoint make the top five. Most of those are accessible via the internet and available for free and also commonly used when working or leisurely using the computer. It can thus be concluded that easy access and familiarity are important requirements for tools to be used frequently while learning.

Such statistics do not necessarily reflect the actual usage of tools and services while learning though they do give an impression about what learners are generally interested in. For a thorough behaviour analysis, however, tracking of learning processes needs to be done in a personalized learning environment as not only the fact that a tool has been used should be recorded but also the actions conducted with that tool.

At this point, in addition to the self-regulated learner's choice of tools and services, the next critical aspect of personalization comes into play: the adaptation of the PLE's behaviour to a learner's preferences, goals, background, knowledge, experience, etc ([25]). For a system to appropriately react to the learner's needs, the observation and collection of user behaviour data is required. This is where the aforementioned self-recording and self-monitoring ties in to personal learning environments. When the learner's behaviour is being recorded, it can later be reproduced and analyzed so that the learner can adjust her further learning processes accordingly. As this is only possible within a personalized environment, it is evident that self-regulation demands personalization.

2.3 Stock-Taking

Self-regulated learning takes place when the learner by strategically planning and meta-cognitively assessing is in control of the actions involved in the learning process. A personalized learning environment (PLE) supports the needs of a self-regulated learner. For a PLE to be able to react to a learner's preferences and goals and for a self-regulated learner to successfully achieve these goals it is – if not essential, then at least – highly desirable to get feedback about her behaviour. This feedback can be obtained by self-monitoring and self-evaluating while learning. As the two significant features of self-monitoring are regularity and proximity, a software tool can trace the required data while learner and PLE interact.

The tracing tool has to comply with several requirements so as to be of avail to the learner: Firstly and – maybe – most important of all, the learner may not be disturbed in her studying process thus forcing the tool to run in the background, silently recording and analyzing all actions. Secondly, the recordings have to contain extensive observations taking all such tools and applications into account that are actually being used by the learner. Thirdly, the data being recorded must not be too finely grained in order for the learner to actually deduce something from the recordings of her behaviour. The actions have to be recorded and the recordings have to be presented in such a way that the learner can understand them and recapitulate her course of actions. To this end, the recorded actions must be meaningful regarding the learning process. As an example, the opening of a document is an action that can be recapitulated and put into the context of other actions by a learner. The recording of a single keystroke or a sequence of keystrokes, at contrary, will not give the learner insightful information on her learning activities.

The following section presents the realization and implementation of a tracing tool in compliance with these requirements.

3 CAMera and Usage Metadata Analysis

3.1 CAMera

We call the tracing tool for supporting self-monitoring in personalized learning environments CAMera – "CAM" because its design is based on the *Contextualized Attention Metadata* (CAM) schema for representing user actions ([26], [27]); "camera" because, like a camera, it is basically a recording tool. The architecture of the tool is largely determined by the requirements mentioned in the previous section: the tool must continuously collect usage metadata[5], transfer these metadata into a well-defined format, store them and hold them ready for further analysis and the on-demand generation of usage reports. The tool must not disturb the user from doing her actual work and, thus, it must not make use of obtrusive sensors like eye-trackers etc. The usage reports given by CAMera have to be reliable. Therefore, the reports must not be based on defeasible interpretations of the user's actions. Suppose, for example, that a user opens an email-message and replies to it. It is highly probable that she read the message or at least parts of it. It is, however, also possible that she opened the message and replied to it by accident without reading. Therefore, the CAMera tool may not store that the user read the message. Instead, the actions recorded by the tool have to be on a level of lower granularity that does not demand defeasible interpretation. Nevertheless, the observations must not be too fine-grained as representations of actions observed and reported by the tool have to be meaningful to the user. The solution to this problem is to record the interaction of users with application programs and the file system: the tool records when text documents are opened, modified and stored with word processors, when data objects are moved or deleted, when emails are sent, chat-messages are uttered, queries are posted to search engines, and so on. Such actions can be tracked without further interpretation. At the same time representations of such actions are immediately interpretable.

Since usage metadata are captured from application programs, the CAMera tool firstly consists of a set of metadata collectors, that collect usage metadata from application programs and transfers these metadata into the CAM-schema. In most cases, it suffices to transfer existing log data into CAM; in other cases a metadata collector has to be implemented as a proper monitor component that generates usage metadata instead of just collecting them from existing log-files. At present, we possess metadata collectors for the Thunderbird email-client, the Skype chat-messenger, the Firefox browser and Microsoft Outlook. Furthermore, we have adapted the User Activity Logger developed at L3S (Leibniz Universität Hannover) for recording accesses to the file system. The ALOCOM Framework provides us with usage metadata collectors for MS Power Point and MS Word ([28], [29]). Finally, we exploit recordings of Flash Meetings ([30]). Hence, we are provided with some metadata collectors that we can make use of and experiment with although the set of collectors is still to be extended. We aim at providing collectors for all of the most-used tools mentioned in the previous

[5] 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 [27] instead). As such, they are data about data.

section. Also, it will be necessary to adapt existing collectors to new versions of their source applications. It will be possible for the user to switch data collectors on and off and thus to control from which applications usage data are being collected.

The metadata collectors mentioned capture usage metadata from application programs that run locally on a user's computer. Usage metadata can also be captured from remote servers with which the user interacts. In section 3.3, we will give an example on the collection of CAM-instances from server data.

Secondly, the CAMera tool consists of a database in which the CAM-instances are stored. We are performing experiments with different kinds of databases, both relational databases and xml-native databases, in particular the eXist-database ([31]). The databases provide us with interfaces for generating usage reports.

Thirdly, CAMera consists of analysis applications for the evaluation of CAM-instances, for instance in order to detect the network of people a user communicated with or the most heavily used objects over a certain time span. The aim of the CAMera tool is to support self-monitoring for improving outcomes both of solo and collaborative learning processes. In the next two sub-sections we introduce two applications that do or can serve as CAMera components and, as we assume, support this goal. The first application exclusively monitors and analyzes a user's email-exchange. Email usage metadata and the analyses of these data can be accessed via the CAMera interface. The component runs locally on the user's computer, hence, the metadata generated are under her control. The second application is a *Zeitgeist* application that monitors and analyzes the interaction of several users with the MACE system for architectural learning ([32]). It consists of a set of web services; usage metadata are generated and stored remotely within a network so metadata of different users can be cumulated and processed together.

3.2 Email Analyzer

The email-component of the CAMera tool consists of two collectors for recording email-exchange, which can be applied together or separately, and an analyzer for generating and representing social networks.

The first collector analyzes email-messages that are stored locally on the user's computer in *mbox*-format or that can be retrieved from an IMAP server. For each message, the collector generates and stores a CAM-instance. In the course of analysis – with *Java Mail* ([33]) – it extracts the sender, the receivers, the subject line and the message body. From the body, keywords are extracted which then serve as a shallow content representation. Currently, keyword extraction is carried out with the yahoo! term extractor ([34]) and tagthe.net ([35]).[6] The user chooses one or both keyword extractors and, if she uses both extractors together, specifies whether the intersection or the union of the two sets of keywords is stored. The tool for recording the email-exchange does not need to be permanently active as it can run either continuously or on demand for the analysis of previously stored messages. Moreover, the user can decide which messages are analyzed by specifying a time interval or by explicitly freeing or blocking email-folders.

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

The second collector is based on the plug-in *Adapted Dragontalk* ([36]) that permanently records the interaction of a user with a Mozilla tool like the Thunderbird email-client or the Firefox browser.[7] In our case, it records all events involving Thunderbird, for instance the opening of an email-message, the creation of a new folder or the moving of a message to a particular folder. The original *Adapted Dragontalk* plug-in generates usage metadata and writes these data into simple text files. We adapted the plug-in so that a CAM-instance is generated for each event and then stored in a database (*adapted Adapted Dragontalk*).

Before a user's social network can be created, the e-mail analyzer has to deal with the fact that computer users can have more than one email-address and more than one alias for these addresses. (An example for an alias-address-pair is "Jane Q. Public <jqpublic@example.org>".) A user's alias-address-pairs are to be assigned unambiguously to this individual user or her ID, respectively. To this end, we adapt the approach of Bird et al. ([38]) for computing the similarity of two alias-address-pairs according to the Levenshtein distance ([39]). If the distance is below a certain threshold, we assume that the two alias-address-pairs belong to the same user. Email-messages that have been sent to or from different email-addresses can now be assigned to the same person.

The email-analyzer evaluates email-related CAM-instances for representing a social network. Every person that occurs as sender or recipient of a message is represented by a node within the network. Two nodes are connected by an edge iff the respective persons are involved in the same message (as sender or recipient). The more email-messages two persons are jointly involved in, the stronger the edge between the respective nodes is. Figure 2 shows the CAMera tool displaying a user's social network. The network represents connections to those persons with whom the user has exchanged at least ten messages within a selected time interval.

The email-analyzer provides the user with an interface for browsing and manipulating the network: by marking a person's node, a list of email-messages in which the particular person has been involved is generated and displayed together with the keywords of these messages. Furthermore, time intervals can be specified on a time line; thereby the keywords are weighted regarding their frequencies within these intervals and thus displayed larger or smaller. By clicking on a keyword, the list of messages is reduced to the messages that contain the selected keyword.

The network itself can be manipulated in three different ways: firstly, by naming keywords one can highlight the nodes and edges that have been established due to messages that contain the keywords. Secondly, a user can specify a time interval and reduce the network according to the messages exchanged within this interval. This makes it possible to follow the dynamics of the network in the course of time. Thirdly, a user can set the minimal number of messages that must have been exchanged so that a person and an edge to this person appear in the network. By standard, a person appears in the network, if she was involved in at least one message. By setting a higher minimal number, sporadic contacts can be filtered out in order to make the network representation more concise.

The email-analyzer gives a user an insight into the structure of her social network. It depicts the persons with whom the user has been in contact and the issues of her email-exchange. Therefore, it gives an account on a specific type of communication

[7] *Adapted Dragontalk* (L3S, Leibniz Universität Hannover) is a further development of *Dragontalk* which was developed at DFKI Kaiserslautern ([37]).

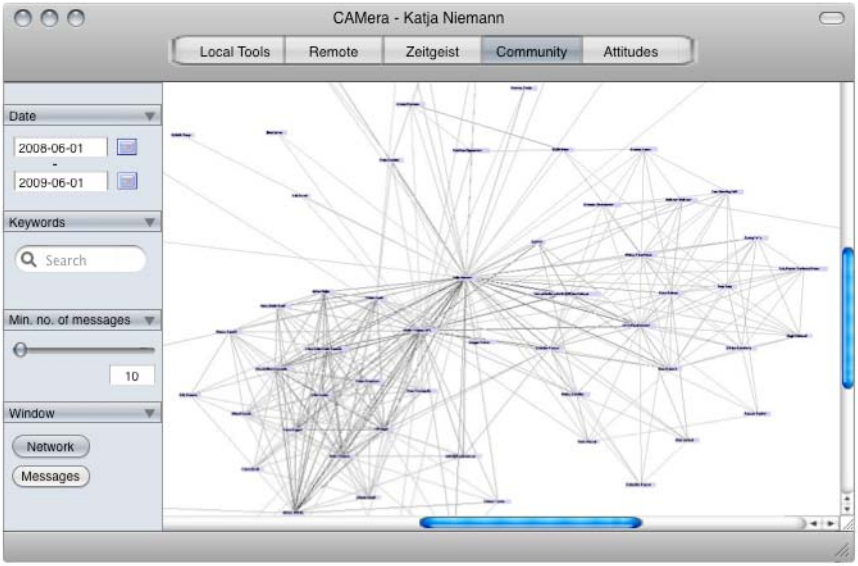

Fig. 2. Representation of a social network within the CAMera tool

behaviour and it supports the user in reflecting her communications. According to Viégas et al. ([40]), users are fascinated by the possibility of evaluating the social networks that are entailed in their email-conversations. (Thus, the email-analyzer arouses interest even without serving an immediate purpose.) Since communication is an integral part of collaborative learning (v.s., section 2), we assume that monitoring communication behaviour also contributes to the reflection of collaborative learning processes.

3.3 MACE *Zeitgeist*

The second application we introduce here is a *Zeitgeist* application that is implemented as part of the MACE system. The MACE system ([32], [41]) sets up a federation of architectural learning repositories: large amounts of architectural contents from distributed sources are integrated and made accessible for architects, architecture students and lecturers. The contents are enriched with various types of metadata, among them Learning Object Metadata (LOM) and CAM. Examples of the user actions that are captured within the system are search actions (with the respective search keywords as *related data*), downloads of metadata on learning resources and downloads of the learning resources themselves, modifications of metadata, etc.[8]

[8] The MACE system is intertwined with the ALOE system ([42]). ALOE is a web-based social media sharing platform that allows for contributing, sharing and accessing arbitrary types of digital resources. Users can up- and download resources; they can tag, rate, and comment on resources; they can create collections and add arbitrary kinds of metadata; and they can join and initiate groups, among other actions. The ALOE system provides observations of user activities related to the MACE system which are stored in the MACE usage metadata store. (See [43] and [44] for more information about ALOE and the system architecture.)

The *Zeitgeist* application is a set of web services that together provide an overview on activities within the MACE system. It gives users the possibility to reconstruct their learning paths by retracing which learning resources they accessed, how they found them and which topics have been of interest to them. This information can be used to make learning paths more explicit and to foster the learners' reflection on their activities.

Figure 3 shows the *Zeitgeist* dashboard that is used to give an overview on a user's MACE-related activities: the *Usage Summary* (top box) shows the user activities for January 2009 when she viewed the metadata of 136, downloaded 84, bookmarked 60 and tagged 34 learning resources. Further details on the objects that have been accessed can be viewed by following the links called "Details". The *Usage History* (middle box) shows the activities of the user per week, indicating when she viewed metadata, downloaded resources and tagged and bookmarked them. By simple statistics like these, the user recapitulates when she was looking for learning resources and when she found suitable ones. According to the graph, she constantly viewed resources during the week. Downloading and tagging, however, significantly increased after Thursday. Presumably, she started by searching for relevant data in the beginning of the week. By Thursday, she had found what she was looking for. Therefore, she downloaded these objects and tagged them. The *Daily Content History* (bottom box) lists the resources the user accessed most recently. According to the example given with Figure 3, the user viewed the metadata of "Villa dall'Ava" at 13:04:08 and downloaded the learning resource "Notre Dame du Haut" at 13:03:47. The respective titles of these data objects are linked to the objects themselves.

The *Zeitgeist* dashboard depicted in Figure 3 is a web-based interface. The *Zeitgeist* data, however, can also be requested by the local CAMera-tool and thus – although this is not yet implemented – be presented through the actual CAMera-interface. That is, MACE *Zeitgeist* can become a remote component of CAMera. It

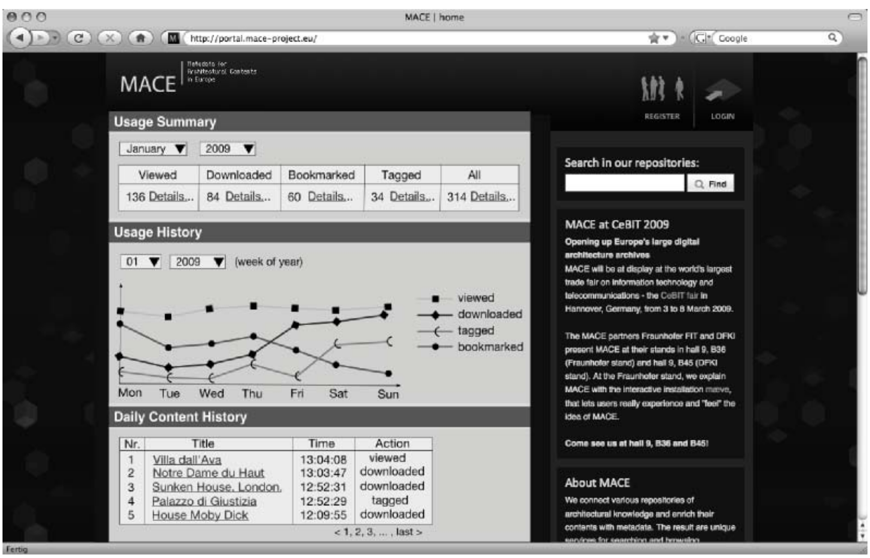

Fig. 3. MACE *Zeitgeist* dashboard

provides an individual user with an overview on her MACE-related activities. It can also cumulate and analyze usage metadata of different MACE users and thus present an overview on all MACE-related activities and on general trends in MACE usage. This gives the individual learner the opportunity to compare her usage with the behaviour of the mass of MACE users. She can follow trends or, at contrary, refrain from trends and find new ways of exploring contents. An additional advantage of collecting metadata from different users is that now users can be compared regarding their usage profiles. A very simple usage profile can be defined as the set of objects that have been accessed in a certain time interval; the similarity of user profiles correlates with the cardinality of their intersection. Therefore, the *Zeitgeist* component not only provides data for reflecting one's own learning behaviour but can also determine and point to similar learners which might be good cooperation partners. This is a clear advantage over a locally running component that observes and analyzes only a single user's browsing behaviour. (With the *Adapted Dragontalk* plug-in, we are already provided with a respective metadata collector.) The local component can collect CAM-instances about the individual user's interaction with the MACE system. However, it cannot easily integrate other kinds of metadata that are provided with MACE (LOM, e.g.), nor can it account for activities of other MACE users.

The *Zeitgeist* component provides the learner with an overview on her MACE-related learning paths. It lets her remember how she came to the engagement in her current issues. It helps her to maintain an overview on her activities and the development of her interests. Thus, the *Zeitgeist* component supports her self-monitoring.

4 Conclusions

We have argued that self-regulated learning is especially promising regarding learning outcomes and therefore should be supported. Self-monitoring is an essential part of self-regulated learning; the support of self-regulative learning can consist in the support of self-monitoring. We introduced CAMera as a tool for monitoring and reporting on a learner's computer-related activities. In particular, we introduced an email-analyzer as an internal CAMera-component and a *Zeitgeist* application that can become a remote component.

The CAMera tool is still under development, and – even further – it is necessarily continuous work in progress. It monitors user interaction with application programs and remote services. As application programs change and new tools and services are being developed, metadata collectors have to be exchanged and added. In addition, new tools most probably require new usage metadata analyses. As a consequence, new analysis components have to be implemented. We therefore designed CAMera as an open system to which new components can be easily added. CAMera can also function without observing a learner's *entire* computer usage behaviour; it may just monitor the interaction with a few selected applications.

So far, we only informally evaluated CAMera and its components by making them accessible to colleagues. The feedback we received was very good: the colleagues like to play with the components; they are interested in the reports and analyses; they state that they understand their own behaviour better. We still have to prove by a formal evaluation that the usage of the CAMera tool in fact, not only in principle, supports self-regulated learning effectively. Optimally, we have to show that the usage of this tool leads to better learning outcomes. To this end, test-beds with large groups of self-regulated learners are needed. We are currently designing such test-beds within the European research project ROLE (*Responsive Open Learning Environments*, [45]). Evaluations within these test-beds will be performed in the near future.

References

1. Torrano, F., González, M.C.: Self-regulated learning: current and future directions. Electronic Journal of Research in Educational Psychology 2(1), 1–34 (2004), http://www.investigacion-psicopedagogica.org/revista/articulos/3/english/Art_3_27.pdf (accessed: April 19, 2009)
2. Zimmerman, B.J., Schunk, D.H. (eds.): Self-regulated learning and academic achievement: Theory, research and practice. Springer, New York (1989)
3. Schunk, D.H., Zimmerman, B.J.: Self-regulation of learning and performance: Issues and educational applications. Erlbaum, Hillsdale (1994)
4. Schunk, D.H., Zimmerman, B.J.: Self-regulated learning: From teaching to self-reflective practice. Guilford, New York (1998)
5. Boekaerts, M., Pintrich, P.R., Zeidner, M.: Handbook of self-regulation. Academic Press, San Diego (2000)
6. Zimmerman, B.J., Schunk, D.H. (eds.): Self-regulated learning and academic achievement: Theoretical perspectives. Erlbaum, Hillsdale (2001)
7. Madrell, J.: Literature Review of Self-Regulated Learning (2008), http://designedtoinspire.com/drupal/node/600 (accessed: April 19, 2009)
8. Pintrich, P.R.: The role of goal orientation in self-regulated learning. In: Boekaerts, M., Pintrich, P.R., Zeidner, M. (eds.) Handbook of self-regulation, pp. 451–502. Academic Press, San Diego (2000)
9. Zimmerman, B.J.: Self-regulated learning and academic achievement: An overview. Educational Psychologist 25(1), 3–17 (1990)
10. Butler, D.L., Winne, P.H.: Feedback and self-regulated learning: A theoretical synthesis. Review of Educational Research 65(3), 245–281 (1995)
11. Kitsantas, A.: Self-monitoring and attribution influences on self-regulated learning of motoric skills. Paper presented at the annual meeting of the American Educational Research Association (1997)
12. Nota, L., Soresi, S., Zimmerman, B.J.: Self-regulation and academic achievement and resilience: A longitudinal study. International Journal of Educational Research 41(3), 198–215 (2004)
13. Fireman, G., Kose, G., Solomon, M.J.: Self-observation and learning: The effect of watching oneself on problem solving performance. Cognitive Development 18(3), 339–354 (2003)
14. Schunk, D.H.: Self-Monitoring as a motivator during instruction with elementary school students. Paper presented at the annual meeting of the American Educational Research Association (1997)
15. Winne, P.H., Jamieson-Noel, D.: Exploring students' calibration of self reports about study tactics and achievement. Contemporary Educational Psychology 27(4), 551–572 (2002)
16. Gress, C.L., Fior, M., Hadwin, A.F., Winne, P.H.: Measurement and assessment in computer-supported collaborative learning. Computers in Human Behavior (in press) (Corrected Proof)

17. Wilson, S., Liber, O., Beauvoir, P., Milligan, C., Johnson, M., Sharples, P.: Personal Learning Environments: Challenging the dominant design of educational systems. In: Proceedings of the first Joint International Workshop on Professional Learning, Competence Development and Knowledge Management (LOKMOL 2006 and L3NCD 2006), pp. 67–76 (2006)
18. Chatti, M.A.: Requirements of a PLE Framework. Blogspot (2008), http://mohamedaminechatti.blogspot.com/2008/02/requirements-of-ple-framework.html (accessed: April 19, 2009)
19. Chatti, M.A., Jarke, M., Frosch-Wilke, D.: The future of e-learning: a shift to knowledge networking and social software. International Journal of Knowledge and Learning 3(4/5), 404–420 (2007)
20. w3schools.com – Browser statistics, http://www.w3schools.com/browsers/browsers_stats.asp (accessed: April 19, 2009)
21. Wikipedia – Usage share of web browsers, http://en.wikipedia.org/wiki/Usage_share_of_web_browsers (accessed: April 19, 2009)
22. Fingerprint – Email client statistics, http://fingerprintapp.com/email-client-stats (accessed: April 19, 2009)
23. Wakoopa, http://wakoopa.com/ (accessed: April 19, 2009)
24. Centre for Learning and Performance Technologies: Top Tools for Learning (2009), http://www.c4lpt.co.uk/recommended/ (accessed: April 19, 2009)
25. Brusilovsky, P.: Adaptive Hypermedia. User Modeling and User-Adapted Interaction 11(1-2), 87–110 (2001)
26. Wolpers, M., Najjar, J., Verbert, K., Duval, E.: Tracking Actual Usage: the Attention Metadata Approach. Educational Technology & Society 10(3), 106–121 (2007)
27. Schmitz, H.C., Kirschenmann, U., Niemann, K., Wolpers, M.: Contextualized Attention Metadata. In: Roda, C. (ed.) Human Attention in Digital Environments. CUP, Cambridge (2009)
28. 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)
29. Ariadne ALOCOM Tools, http://www.ariadne-eu.org/index.php?option=com_content&task=view&id=65&Itemid=96 (accessed: April 19, 2009)
30. The Flashmeeting Project, http://flashmeeting.open.ac.uk (accessed: April 19, 2009)
31. eXist Open Source Native XML Database, http://exist.sourceforge.net (accessed: April 19, 2009)
32. MACE – Metadata for Architectural Contents in Europe, http://portal.mace-project.eu (accessed: April 19, 2009)
33. JavaMail API, http://java.sun.com/products/javamail (accessed: April 19, 2009)
34. Yahoo! Developer Network: Term Extraction Documentation, http://developer.yahoo.com/search/content/V1/termExtraction.html (accessed: April 19, 2009)
35. tagthe.net, http://tagthe.net (accessed: April 19, 2009)
36. Adapted Dragontalk, http://www.l3s.de/~chernov/pas/Documentation/Dragontalk/thunderbird_documentation (accessed: April 19, 2009)
37. Epos – Evolving Personal to Organizational Knowledge Spaces,

http://www3.dfki.uni-kl.de/epos (accessed: April 19, 2009)
38. Bird, C., Gourley, A., Devanbu, P.T., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proceedings of the International. Workshop on Mining Software Repositories, Shanghai (2006)
39. Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys 33(1), 31–88 (2001)
40. Viégas, F.B., Golder, S., Donath, J.: Visualizing email content: portraying relationships from conversational histories. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, Montreal, pp. 979–988 (2006)
41. 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.) EC-TEL 2007. LNCS, vol. 4753, pp. 322–336. Springer, Heidelberg (2007)
42. ALOE-Project, http://aloe-project.de (accessed: April 19, 2009)
43. Memmel, M., Schirru, R.: Sharing digital resources and metadata for open and flexible knowledge management systems. In: Tochtermann, K., Maurer, H. (eds.) Proceedings of the 7th International Conference on Knowledge Management (I-KNOW), pp. 41–48. Journal of Universal Computer Science (2007)
44. Memmel, M., Schirru, R.: Aloe white paper. Technical report, DFKI GmbH (2008)
45. ROLE – Responsive Open Learning Environments, http://www.role-project.eu (accessed: April 19, 2009)