ANALYSING CONTEXTUALIZED ATTENTION METADATA FOR SELF-REGULATED LEARNING
A Supporting Framework for Self-Monitoring and Self-Reflection

Maren Scheffel, Frank Beer and Martin Wolpers

Fraunhofer Institute for Applied Information Technology FIT, Schloss Birlinghoven, 53754 Sankt Augustin, Germany

Keywords: Usage metadata, Attention analysis, Self-regulated learning, Self-reflection, Self-monitoring, Software framework, Personalised learning software, Personal learning environment.

Abstract: In order to successfully learn in a self-regulated way, self-monitoring of the learner and reflections of learning behaviour is required. We therefore introduce a framework that collects usage metadata from application programs and stores them as Contextualized Attention Metadata (CAM). We also present three approaches on how we exploit the collected CAM for further analysis such as object recommendation or learning activity classification in order to help the learner become aware of her learning behaviour, to self-reflect and to support her during her learning processes.

1 INTRODUCTION

Self-regulated learning demands self-monitoring from a learner. To improve her learning outcomes a learner reflects on her previous learning activities and adjust her learning behaviour accordingly. For a learner to look back on her actions, the learning activites need to be recorded. In this position paper we present our ongoing work with an extensible framework that accumulates usage metadata from different applications as Contextualized Attention Metadata (CAM) to support the user in her self-regulated learning. After outlining the framework's functionalities we give an overview of aspects of CAM analysis currently being developed by us. First evaluations support our research and show that we are on the right track. Finally we conclude with further extensions, experiments and evaluations still to be conducted.

2 REFLECTION-BASED LEARNING

Self-monitoring and self-reflecting one's own behaviour is key to self-regulated learning. To successfully achieve learning goals, whether they are set for solo or collaborative learning processes, self-regulated learning is an auspicious method. For computer-based self-regulated learning the self-monitoring of a learner can be supported by recording her interaction with the computer. Later she can then analyse and evaluate her learning processes by self-reflection.

The term self-regulated learning denotes a learning process where the learner becomes aware of her learning and consciously takes care of it. This does not exclude support or guidance by a teacher or tutor as long as the learner's autonomy is not questioned. A self-regulated learner has the ability to meta-cognitively assess, strategically plan, monitor, self-reflect and evaluate her learning activities.

In their report of current and future directions of self-regulated learning, (Torrano and Gonzalez, 2004) say that characteristically self-regulated learners are people actively participating in learning on a behavioural, motiviational and meta-cognitive level. Self-motivation and the employment of strategies to achieve desired outcomes are also typical attributes of self-regulated learners. In his learner model Pintrich divides the process of self-regulation into four phases - planning, self-monitoring, control and evaluation - which in turn are composed of four areas - cognition, motivation/affection, behaviour and con-

text (Pintrich, 2000). Another important aspect of self-regulated learning is the need for feedback, especially the self-oriented type (Zimmerman, 1990). A detailed account on feedback and self-regulated learning can be found in (Butler and Winne, 1995). Forethought, performance and self-reflection are the three aspects that make up Zimmerman's loop of self-regulated learning (Kitsantas, 1997). A learner can adjust and change her learning bahaviour and activities meta-cognitively assessing, analysing and evaluating her learning processes. Self-reflection is therefore not only a useful but even a very essential part of self-regulated learning that can support the successful acquisition of academic (Nota et al., 2004) as well as non-academic (Kitsantas, 1997) skills. Other very relevant publications on self-regulated learning are (Zimmerman and Schunk, 1989), (Schunk and Zimmerman, 1994), (Schunk and Zimmerman, 1998), (Boekaerts et al., 2000), (Zimmerman and Schunk, 2001) and (Madrell, 2008).

To help learners become aware of their actions and to adjust their behaviour, their learning activities need to be monitored. The two important criteria when recording and self-monitoring one's behaviour are regularity and proximity. Only if these criteria are fulfilled, does a learner get a sound impression of her actions. As tracing the features of learning has most use when done during the action of learning (Winne and Jamieson-Noel, 2002), the use of computers to trace the learner's behaviour is very convenient and the learner is free in her actions as opposed to other recording measures, such as thinkalouds which might disturb or interfere with the learning process or post-test questionnaires where she can forget to mention things. Offering real-time analysis of everything a learner does on her computer (e.g. chats, reading and writing documents, file system interaction) makes it easier for her to self-reflect and understand her learning processes whether they are solo or collaborative ones (Gress et al., ip).

3 THE CAM SCHEMA

The schema we use to describe a learner's computer-related activities is the Contextualized Attention Metadata (CAM) schema, an extension of Attention.XML which allows to model a user's handling of digital content (e.g. web sites, text documents, pictures, etc.) across system boundaries (Najjar et al., 2006). The core elements of the CAM schema are depicted in figure 1. A full description can be found in (Wolpers et al., 2007).

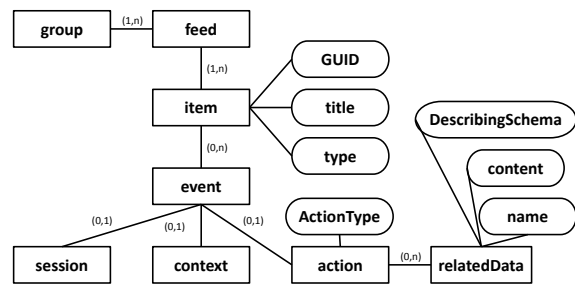Observations in CAM are focused on the user and

Figure 1: Core Elements of the CAM schema.

are thus recorded for individual users. Each user is represented by one element called *group*. Each *group* can contain one or more *feeds* which represent the applications used by the user such as browsers, email clients or word processors. Each *feed* can contain one or more *items* which are described by properties such as *title*, *type* (e.g. text, image, audio, video, etc.) and *GUID* (globally unique identifier). A browser *feed* for example can contain a website as an *item* and its *title* as a property. An email client *feed* could contain an email or an address book entry as *item*. The *item* element can contain one or more *event* elements which in turn comprise one *action* and one *session* element. *Actions* are associated with an *ActionType* - for an email *item* the *action* described could be "read", "forward" or "move to another folder" - and a timestamp (*dateTime*). *Actions* can be enriched with *relatedData* such as sender and receiver(s) of an email or keywords for a shallow content representation. The *session* element stores the session id and information about the user. CAM is developed to describe as many types of attention metadata as possible. Therefore, CAM records of a user do not merely describe the user's foci of attention but rather her entire computer usage behaviour.

The CAM schema fulfils three important requirements that are crucial for the effective collection, storage, presentation and analysis of usage metadata: firstly, the user need not be disturbed in her computer-related activities, since the collection of Contextualized Attention Metadata can run in the background. Secondly, CAM recordings can contain extensive observations taking all the tools and applications into account that are actually being used. Thirdly, CAM are not too finely grained in order for the user to actually deduce something from the recordings of her behaviour. The actions are recorded and the recordings are presented in such a way that the user can instantly understand them and recapitulate her course of actions. The opening of a document for example is an action that can be recapitulated and put into the context of other actions by a user, the recordings of a single keystroke, however, will not give the user in-

sightful information on her activities.

4 THE CAM FRAMEWORK

In order to collect CAM records of a learner's interactions with the different applications, specialised application wrappers that make use of the CAM schema as the transferring protocol to general observation repositories, are combined in the CAM framework. Thus, possible recommendations or reflections, supported by the framework, are not limited to a snap-shot analysis but to an extensive assistance for learners and teachers in learning scenarios. The functionality of the whole framework is delivered through several active and passive components. The active elements are those that capture and utilize the attention of the learner in her working environment while the passive members ensure functionality used by the active components.

The *passive elements* of the framework are those parts that provide callable or event-driven functionality. The most important parts are the observation repositories, the service-oriented software library and the configuration windows. To persist or retrieve recent observations, the framework provides a local and a global data repository. They can be accessed through the software library that manages the dynamic adressing for each framework instance. The settings of the framework, i.e. database ajustments and framework redirections are also controlled by the library. With that, the library provides some type of middleware system including the handling of concurrency, i.e. a unit of work (Fowler, 2002). In the sense of usability, the configuration files can be setup via an intuitive graphical user interface (GUI). The privacy level of sensitive data can therfore be individually controlled by each user at any time. That includes the possibility for personal analysis, taking only local observations or even the shared utilization of attention with the user's learning community into account.

The *active elements* of the framework can be seperated into two reasonable actors: wrappers and analysis tools. They are self-contained applications that collect and evaluate the activities done by the user in his daily working and learning environment. Thereby, each wrapper instance captures locally appropriate information of its domain, e.g. time spent with the favourite chat client or modifying documents on the file system. If permitted, analysis tools retrieve these collected information in a statistical way to utilize personal feedback or to publish recommendations to each user individually.

While the active components can be viewed as independent systems that make use of the functionality delivered by the software library, further extensions with wrappers and analysis tools do not seem to be problematic, so far. From this point of view it can be inferred that the technical potentials of the framework mostly depend on the expandability and the scalability of the software library. With this, we emphasize a simple but reliable design that can be extracted from figure 2 and the following considerations:

In a service-oriented architecture as originally described in (Schulte and Natis, 1996a; Schulte and Natis, 1996b), we assume in a more technical form a loose coupling dependency between consumer and service through a formally specified service interface (contract). In this context, a client or consumer is an instance that is relying on a functionality (service) of the software library, i.e. a wrapper, an analysis tool or the GUI. This combination allows flexibility, pointing to service updates or substitutions because the client does not have sufficient information on the service at compile time but on the contract. In a loose coupling environment this issue makes it necessary to either inject the implementation of the interface to the consumer in a specified way or to introduce a fourth component, i.e. a service locator (Fowler, 2004). Here we make use of a service locator that works as a singleton registry in the library of each framework instance. It has appropriate details of the intended service and requests a service implementation via the factory package that holds reflection-oriented service factories (Gamma et al., 1995). Once a service is registered at the service locator and returned as a reference, the client has direct access to the service and can operate on it via its concluded contract (Hack and Lindemann, 2007).

To this point, we have equipped the framework with two service implementations that can handle requests to the native XML dababase systems eXist[1] and Tamino[2]. Another component that is currently in progress is the interaction with the object-relational database system PostgreSQL[3]. This includes a transformation of the existing CAM schema to a relational model. The metadata collecting wrappers we posses so far are for the Thunderbird email-client, the Skype chat-messenger, the Firefox browser and MS Outlook. In order to record accesses to the file system, we have adapted the User Activity Logger developed at L3S of the Leibniz Universität Hannover (L3S, 2007). Usage metadata collectors for MS Powerpoint and MS Word are provided by the ALOCOM

---

[1] http://exist.sourceforge.net/

[2] http://www.softwareag.com/corporate/products/wm/tamino/default.asp
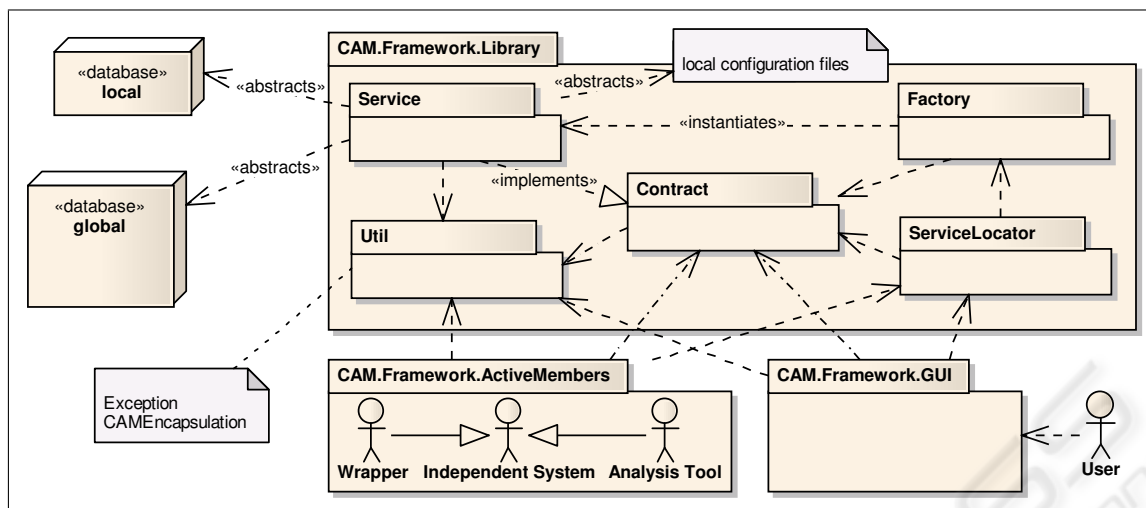
[3] http://www.postgresql.org/

Figure 2: Architecture of the CAM framework in UML notation.

Framework (Verbert et al., 2005; Ariadne, 2006). Although the set of collectors still needs to be extended, we are provided with some metadata collectors that we can make use of and experiment with.

5 CAM ANALYSIS

Several aspects of how to exploit the collected metadata are looked into by us at the moment. For one, we are developing a tool called CAMera (Scheffel et al., 2009; Schmitz et al., 2009) - "CAM" because its design is based on the Contextualized Attention Metadata schema and "camera" because, like a camera, it is basically a recording tool - that builds on the CAM framework for collecting and storing the accumulated metadata and also offers analysis applications to the learner. As the collected CAM are stored locally, the learner herself has full control over the tool and data. Several components can be accessed via CAMera's GUI. One component simply displays the colleced attention metadata records for a chosen time span. The displayed items can be ordered according to different features, i.e. tool, action, object or date. This enables the learner to analyse her computer-related behaviour. Another component analyses a learner's email-exchange to generate and depict a social network. Every person occurring as sender or recipient of a message is represented by a node within the network. Nodes are connected if the corresponding persons have been involved in the same message. The more messages two persons are jointly involved in, the stronger the connection between their respective nodes is. This helps the learner to become aware of her interactions with others. With the CAMera tool's chat component we are interested in exploring which

hypotheses can be deduced from recorded metadata about the emotional and cognitive states of conversation partners, their relationship to each other and the communication situation including current psychological theories of communication. The tool's browsing component can help the learner to retrace her steps while interacting with the Firefox internet browser or her computer's file system. The component includes a *Zeitgeist* application for statistically evaluating browsing activities and detecting individual trends in usage. Based on this, certain actions or objects can be recommended to the learner. Self-monitoring her behaviour with the CAMera tool gives the learner the opportunity to reflect on her actions and adjust them accordingly. She can, for example, reconstruct how she progressed and what data she (maybe unknowingly) sent to others.

Another aspect we are looking into is basing object recommendation on usage context (Friedrich et al., 2009). We claim the hypothesis that usage similarity gives rise to content similarity and can thus be used for recommendations. We therefore defined the notion of a usage context profile (UCP) for data objects. The UCP of an object o can be derived from a set of usage histories; it contains the objects that were used before and after o was accessed. We then introduced a similarity measure for UCPs. The approach we take is an item-based collaborative filtering one where for two objects to be deemed similar, their usage contexts have to be similar. We can therefore recommend object o_2 to a user who previously used o_1 based on the fact that o_1 and o_2 have similar usage contexts which not necessarily entails that they have both been used by the same users. First results support our hypothesis that usage context similarity is an indication of content similarity. From 100 object

pairs whose UCPs had a similarity of 55% and above, 92% showed content similarities during our manual comparison. This content similarity was only weakly indicated when comparing the objects metadata (e.g. title, description and tags) automatically. Focusing on objects' usage contexts therefore seems a promising base for recommendations and will be further investigated by us.

We are also currently working on utilizing the rough set theory to classify user activities. The rough set theory (Pawlak, 1982; Pawlak, 1991), as an extension of the classical set theory, is a mathematical framework to analyse data under uncertainty. It provides some methods for data reduction and the approximation of concepts, e.g. the indiscernibility relation, the reduct generation and the lower and upper approximation respectively. These tools could be suitable to support self-monitoring and self-reflection as suggested in the following way: the indiscernibility relation is a parameterising equivalence relation with respect to the used attribute set. Thus, it can be applied to extract and represent behaviour patterns of a group of learners or an individual one. Furthermore, it can be used as an initial partition to develop an agglomerative hierarchical clustering algorithm with a reasonable distance measure or metric respectively for CAM. Some expedient measures have been reviewed in (Grimmer and Mucha, 1998; Maimon and Rokach, 2005), but we are not limited to the listed ones. So, on the one hand our framework can be enriched with an unsupervised learning process for CAM observations. On the other hand the learned classification of user activities can be evaluated with the given concept approximation of the rough set theory which could push the qualitiy of CAM analysis for our research. Our first results on the rough set theory in combination with object-relational databases have been documented in (Beer, 2009). They show that the automatically classified activities indeed correspond to users' manually recorded activities during the day.

6 CONCLUSIONS

We have presented a framework that records usage beahviour and stores a learner's computer-based activities as Contextualized Attention Metadata which can then be exploited by analysis applications. First results have shown that the collected CAM can be presented to a learner in such a way that supports self-regulated learning. The recorded attention metadata can also be used as a basis for recommendations or learning activity classification. We now want to ex-

tend our findings and explore further possibilities of CAM analysis. Weighing of context when basing recommendations on the usage context is one aspect, for example. We also want to detect actions, e.g. whether a learner is reading a document, and combine these findings with pedagogical models. Evaluation of all our current results with test beds is another very important aspect we are looking into at the moment.

REFERENCES

Ariadne (2006). Alocom tools. Retrieved January 19, 2010, from http://www.ariadne-eu.org/index.php?option=com_content&task=view&id=65&Itemid=9.

Beer, F. (2009). Objektrelationale Datenbanken und Rough Sets für die Analyse von Contextualized Attention Metadata. Master's thesis, University of Applied Sciences Bonn-Rhein-Sieg (Department of Computer Science).

Boekaerts, M., Pintrich, P., and Zeidner, M., editors (2000). *Handbook of self-regulation*. Academic Press, San Diego.

Butler, D. and Winne, P. (1995). Feedback and self-regulated learning: A theoretical synthesis. *Review of Educational Research*, 65(3):245–281.

Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.

Fowler, M. (2004). Inversion of control containers and the dependency injection pattern. Retrieved January 19, 2010, from http://martinfowler.com/articles/injection.html.

Friedrich, M., Niemann, K., Scheffel, M., Schmitz, H.-C., and Wolpers, M. (2009). Object Recommendation based on Usage Context. Workshop about Context-aware Recommendation for Learning at the STELLAR Alpine Rendez-Vous 2009.

Gamma, E., Helm, R., and Johnson, R. E. (1995). *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Gress, C., Fior, M., Hadwin, A., and Winne, P. (i.p.). Measurement and assessment in computer-supported collaborative learning. *Computers in Human Behavior*, in press(corrected proof).

Grimmer, U. and Mucha, H.-J. (1998). Datensegmentierung mittels Clusteranalyse. In Nakhaeizadeh, G., editor, *Data Mining - Theoretische Aspekte und Anwendungen*, pages 109–141. Physica Verlag.

Hack, S. and Lindemann, M. (2007). *Enterprise SOA einführen*. Galileo Press.

Kitsantas, A. (1997). Self-monitoring and attribution influences on self-regulated learning of motoric skills. *Paper presented at the annual meeting of the American Educational Research Association*.

L3S (2007). User activity logger documentation. Retrieved January 19, 2010, from http://pas.kbs.uni-hannover.de/Documentation/UAL/Index.html.

Madrell, J. (2008). Literature review of self-regulated learning. Retrieved January 19, 2010, from http://designedtoinspire.com/drupal/node/600.

Maimon, O. and Rokach, L. (2005). Clustering methods. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, pages 321–352. Springer Verlag.

Najjar, J., Wolpers, M., and Duval, E. (2006). Attention metadata: Collection and management. In *WWW2006 Workshop on Logging Traces of Web Activity: The Mechanics of Data*.

Nota, L., Soresi, S., and Zimmerman, B. (2004). Self-regulation and academic achievement and resilience: A longitudinal study. *International Journal of Educational Research*, 41(3):198–215.

Pawlak, Z. (1982). Rough sets. *Inernational Journal of Computer and Information Sciences*, 11(341-356).

Pawlak, Z. (1991). *Rough Sets - Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers.

Pintrich, P. (2000). The role of goal orientation in self-regulated learning. In Boekaerts, M., Pintrich, P., and Zeidner, M., editors, *Handbook of self-regulation*, pages 451–502, San Diego. Academic Press.

Scheffel, M., Friedrich, M., Jahn, M., Kirschenmann, U., Niemann, K., Schmitz, H.-C., and Wolpers, M. (2009). Self-monitoring for Computer Users. In Fischer, S., Maehle, E., and Reischuk, R., editors, *Informatik 2009 Im Focus das Leben*, Lecture Notes in Informatics, Lübeck. Gesellschaft für Informatik.

Schmitz, H.-C., Scheffel, M., Friedrich, M., Jahn, M., Niemann, K., and Wolpers, M. (2009). CAMera for PLE. In Cress, U., Dimitrova, V., and Specht, M., editors, *Learning in the Synergy of Multiple Disciplines, Proceedings of the EC-TEL 2009*, volume 5794 of *Lecture Notes in Computer Science*, pages 507–520, Berlin/Heidelberg. Springer.

Schulte, R. and Natis, Y. (1996a). Service-Oriented Architectures. In *Part 1, SPA-401-068*. Gartner Group.

Schulte, R. and Natis, Y. (1996b). Service-Oriented Architectures. In *Part 2, SPA-401-069*. Gartner Group.

Schunk, D. and Zimmerman, B. (1994). *Self-regulation of learning and performance: Issues and educational applications*. Erlbaum, Hillsdale.

Schunk, D. and Zimmerman, B. (1998). *Self-regulated learning: From teaching to self-reflective practice*. Guilford, New York.

Torrano, F. and Gonzalez, M. (2004). Self-regulated learning: current and future directions. *Electronic Journal of Research in Educational Psychology*, 2(1):1–34.

Verbert, K., Jovanovic, J., Gasevic, D., and Duval, E. (2005). Repurposing learning object components. In *Proceedings of OTM 2005 Workshop on Ontologies, Semantics and E-Learning*, Cyprus. Agia Napa.

Winne, P. and Jamieson-Noel, D. (2002). Exploring students calibration of self reports about study tactics and achievement. *Contemporary Educational Psychology*, 27(4):551–572.

Wolpers, M., Najjar, J., Verbert, K., and Duval, E. (2007). Tracking Actual Usage: the Attention Metadata Approach. *Educational Technology and Society*, 10(3):106–121.

Zimmerman, B. (1990). Self-regulated learning: current and future directions. *Educational Psychologist*, 25(1):3–17.

Zimmerman, B. and Schunk, D., editors (1989). *Self-regulated learning and academic achievement: Theory, research and practice*. Springer, New York.

Zimmerman, B. and Schunk, D., editors (2001). *Self-regulated learning and academic achievement: Theoretical perspectives*. Erlbaum, Hillsdale.