

Self-monitoring for Computer Users

Maren Scheffell, 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.scheffell, martin.friedrich, marco.jahn, uwe.kirschenmann, katja.niemann, hans-
christian.schmitz, martin.wolpers}@fit.fraunhofer.de

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.

The amount of available data and the number of users has increased continuously which, on the one hand, enhances the possibility to find the information needed. On the other hand, it can just as well complicate these actions due to information overflow. This is why user and context sensitive recommendation systems are needed. 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. This is where a dilemma occurs: 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.

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 and “camera” because, like a camera, it can record actions and events. 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, a database where the generated CAM instances are stored and a set of analysers that evaluate the usage metadata and present them to the user.

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 other one records and analyses chat messages. The browsing components are *Zeitgeist* components for statistically evaluating browsing behaviour. The first of these components locally analyses browser usage while the second component is a remote component that monitors and analyses interactions with the MACE system for architectural learning.