D3.2 Activity classification

Project	SWELL
Project leader	Wessel Kraaij (TNO)
Work package	WP3
Deliverable number	D3.2
Authors	Suzan Verberne, Maya Sappelli
Reviewers	Jan Geert van Hall, Wessel Kraaij
Date	18 December 2013
Version	2.0
Access Rights	
Status	Final

SWELL Partners:

Almende, Noldus, Novay, Philips, TNO, Radboud Universiteit Nijmegen, Roessingh Research and Development, Sense-OS, Universiteit Twente~~,~~

COMMIT/ A public- private research community

This publication was supported by the Dutch national program COMMIT.

Summary

For this deliverable (D3.2 Activity classification), we implemented two functionalities from the “Time and stress monitor” (see M1 in D3.1b) that will eventually be integrated in the Happy Worker App:

1. Give insight in the distribution of work time over task activities
2. Give insight in the distribution of work time over projects

In this deliverable we describe the implementation of these two functionalities in the Happy Worker App software.

A short demonstration video of the software can be found at https://www.dropbox.com/s/x544e6fiey6dd81/D3.2_movie_with_text.mp4

Contents

1 Introduction

In our connected workplaces it can be hard to work calm and focused. Ruff (2002) speaks of 'plugged in compulsion' as "the strong need to check mail and the internet to stay in touch", and 'hurry sickness' as "the belief that one must constantly rush to keep pace with time". Stress can either directly lead to illness through its physiological effects or indirectly, through maladaptive health behaviour, like smoking, poor eating habits or lack of sleep (Bakker et al., 2010). Certainly, some amount of stress is not harmful and might even be beneficial to gain concentration and focus, but extended periods of stress can be a danger to the health, e.g. causing burn-out.

In our project SWELL (smart reasoning for well-being at home and at work) we aim to improve well-being at work by supporting knowledge workers. We define knowledge workers as people who use and produce information as their main task, and frequently use computers. Well-working could be defined as "being and feeling in control", with a positive impact on work efficiency and effectiveness, work pleasure, mental and physical health status (see deliverable D3.8).

In order to validate the effectiveness of SWELL concepts, we develop SWELL@Work applications that include the following (see deliverable D3.1b update):

1. Fishualization (SWELL board, D3.10);
2. Personal feedback tool: the *Happy Worker App* (HWA);
3. E-mail organization assistant;
4. SWELL golden demo (including applications above).

The Happy Worker App provides personal feedback on working behaviour, currently based on computer logging and input on subjective experience. In this deliverable, feedback is limited to mirrorring the user's behaviour without further intervention. The feedback approach is extended to a social feedback board called Fishualization (SWELL board), which gives employees of an entire department feedback.

For this deliverable (D3.2 Activity classification), we implemented two functionalities from the "Time and stress monitor" (see M1 in D3.1b) in the Happy Worker App (forthcoming):

2. Give insight in the distribution of work time over task activities
3. Give insight in the distribution of work time over projects

In this deliverable we describe the implementation of two functionalities in the Happy Worker App software. The other two functionalities in the "Time and stress monitor" (M1) are:

1. Give insight in working patterns
4. Give insight in activities and stress levels /energy levels

WP3 is currently working on these functionalities but they are not part of this deliverable.

This document starts with a scenario description including requirements and claims for the application. In Section 2 we give a functional description of the software, followed by technical specifications and requirements in Section 3.

1.1 SWELL goals

The following scenario comes from D3.10. It provides the context for the Happy Worker App.

> *Peter is a knowledge worker. He is relatively flexible in his way of working; he chooses his own working hours and how he approaches his work. He wants to make optimal use of his*

> *time and keep his productivity high while keeping his well-being optimal. Peter spends a large portion of his time at his computer, so information about his working behaviour can easily be collected via the computer with a logging tool.*
>
> *Since some days Peter uses our SWELL Happy Worker App which is installed on his computer. A computer logger unobtrusively keeps track of his computer activities. The Happy Worker App provides feedback to him in the form of a personal view on the collected data. Using these feedback charts Peter has learned that his energy and activation levels follow typical patterns. He now knows that he works most concentrated (with few task switches) before lunch. He also learned that report writing is a task that he can best do in the morning, and that when he works on the 'Napoleon' project, he tends to do lots of tasks for other projects in between.*

The awareness of his work patterns helps him planning his work better.

1.2 Context of this deliverable

The software developed for this deliverable will be integrated in the alpha version of the SWELL golden demo, which is to be delivered in the summer of 2014. In Q4 of 2014 we will implement an update of the software, which will lead to deliverable D3.3 (Final and updated implementation of task recognition). The main difference of D3.3 compared to D3.2 is the functionality for real-time use instead of demo-use with historical data.

1.3 Demonstration of the software

Figure 3 shows a screenshot of the Dashboard with Activities Viewer.

A short demonstration video of the software can be found at
https://www.dropbox.com/s/x544e6fiey6dd81/D3.2_movie_with_text.mp4

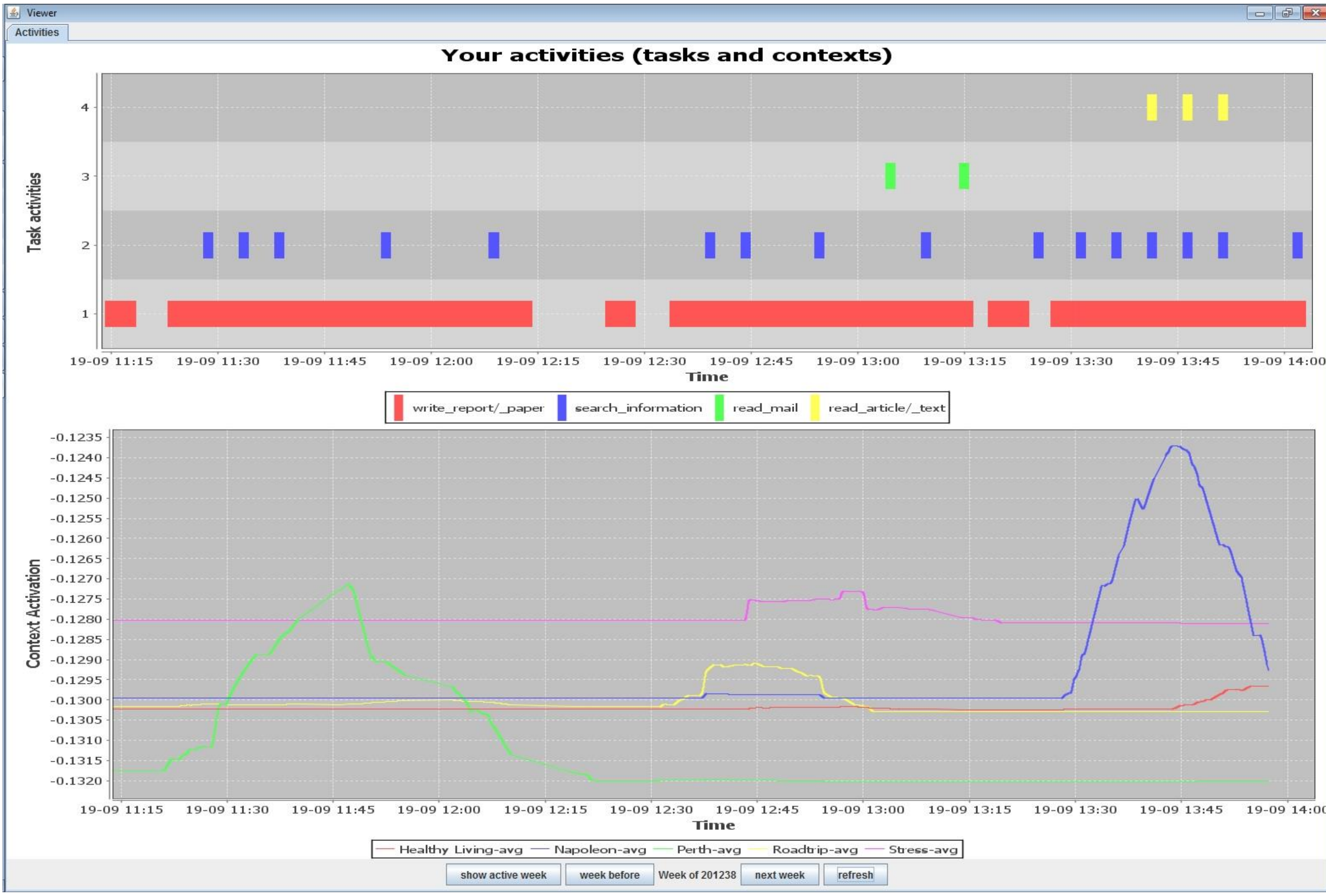

Figure 3. Screen shot of the Activities Viewer in the Dashboard. The top graph is a Gantt chart showing the dispersion of the user's task activities (write report/paper, search information, read mail, read article) over time. The bottom graph shows the Activation (see section 2.3 for a description) of the five contexts (topics) the user has worked on: Healthy Living, Napolean, Perth, Roadtrip and Stress.

2 Functional description

2.1 Main functional requirements

The main functional requirements for the Personal feedback tool *Happy Worker App* in the SWELL@work system are:

- F1: The software unobtrusively collects objective behaviour data (computer logging).
- F2: For feedback, the data is visualized in an intuitive, easily interpretable and appealing way. Our idea is that feedback that is easily interpretable and engaging is most effective for gaining insights.

For this deliverable (D3.2 Activity classification), we implemented two functionalities from the "Time and stress monitor" (see M1 in D3.1b) in the Happy Worker App:

1. Give insight in the distribution of work time over task activities. The backbone of this functionality is the task recognition classsifier in Saskia Koldijk's MSC project at TNO.
2. Give insight in the distribution of work time over projects. The backbone of this functionality is Maya Sappelli's implementation of a feed forward neural network for context recognition (see Section 2.3 for further explanation).

We have implemented these two functionalities in Java, in the SWELL package *HappyWorkerApp* (HWA). HappyWorkerApp is a continuation of the StressTool package developed previously in SWELL WP3 (no formal deliverable). The HWA has a user interface consisting of a task bar icon and a dashboard that the user opens by rightclicking the icon. The dashboard displays with multiple tabs for the visualization of user characteristics: Achievements, Application Charts and Computer Activity Charts. The user can see his (self-adjusted) energy level paired with his activities during the day. For this deliverable, the dashboard contains one tab, 'Activities'. On this tab, two charts are shown: task activities and contexts (projects/topics).

2.2 Task activity recognition

- The basis for the implementation of task activity recognition is the taskRecognition package (software implementation by Saskia Koldijk for her MSC project at TNO, see Koldijk 2011). We extended it (see technical specifications) and integrated the code in the HWA package so that the user can see the recognized tasks on the HWA DashBoard.
- The activities that we distinguish come from Koldijk (2011). For this deliverable, we adhere to this set of labels; in D3.3 we will implement the possibility to add new task activities. The current labels are: read mail, write mail, organize/archive data, plan, make presentation, create visualization, program, write report/paper, search information, read article/text, make overview, analyse data, away from keyboard, none.
- The activity classification has been implemented as a supervised learning task. The classification model was trained and evaluated in Koldijk (2011).[1] The model contains values for 10 features (e.g. average number of clicks per minute, average number of application switches per minute, most used application).
- Incoming user interaction data from uLog is classified with this model. uLog is a tool (developed by Noldus Information Technology) for automatically recording user-system interaction such as keystrokes, mouse clicks and application switches. Our software aggregates the incoming activity per 5 minutes, and determines the user's main activity during those 5 minutes. The classifier gives back a confidence score, which can be used for future active learning purposes (see Section 3.2).
- In the original taskRecognition demo, the extraction of only 4 of the model's features was implemented. We added the extraction of the other feature values for this deliverable.
- The classified task activity labels per are written to text files, that are used to generate the visualization in a Gantt chart. Figure 1 illustrates this process.

[1] For more information about the features and the reliability of the activity classification, we refer to Koldijk (2011)

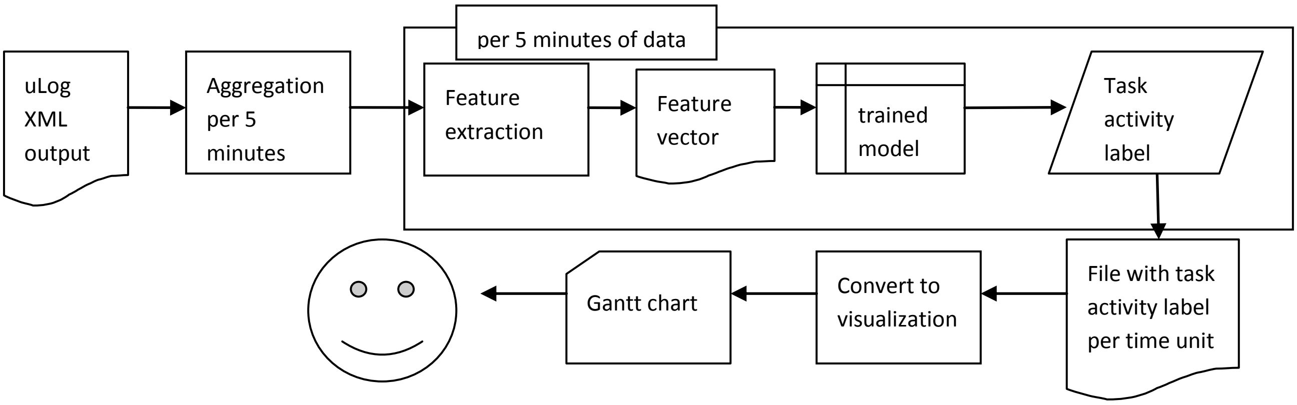

Figure 1. Process of task activity recognition

2.3 Projects/contexts recognition

- The projects that we distinguish have not been predefined in the software but differ per user. In a real working scenario, the projects could be provided by the user or harvested from directory names. In the demonstration scenario we use as projects the topics from the data collected during the SWELL@work experiment (see D3.5 and Koldijk et al. 2013): Healthy Living, Napoleon, Stress, Perth, Roadtrip and Privacy. This experiment was a controlled study in a lab setting with 25 subjects.
- The distribution of work time over projects has been implemented as an unsupervised learning task. This unsupervised model has been developed to track *user context*[2]*.* It is based on the interactive activation (IA) and competition model by McClelland and Rumelhart (1981). This is a feed forward neural network with backpropagation in which activity is spread an collected in the network. The unique pattern of activity determines the active projects. We can understand context in this fashion: events (or event blocks) stimulate relevant context elements (persons, locations, topics). Context elements can stimulate relevant information objects and they in turn stimulate other information objects and also again the context elements. This model is depicted in figure 2.
- For the context recognition, the uLog data is not aggregated over 5 minutes (as in the task activity recognition) but the uLog events are grouped in blocks, where each switch of application defines the start of a new event block. Per event block, textual elements in the event block (window title, typed keys and captions) are used to determine the topics and entities of the event. For each recognized topic and entity a node is created on the context layer, and a connection is made between the event-input-node and the context node. Connections are also made between the event-input-node and the associated location (file folder or domain) and date/time nodes. Date time nodes consist of separate nodes for day of the week, day of the month, month, year, hour and minutes rounded to 00, 15, 30 and 45. Top-down, content of documents are used to determine topics and entities of the document. A document always has a file folder as location. Date/time links of a document

[2] Context is a broader and more abstract term than ‘project’. We see the user’s context as a constantly changing state. Contexts are not static and can partly overlap, for example when some person is associated with multiple contexts (e.g. colleague at work, friend at home). An activity does not happen within a context, but the activity forms the context. In this deliverable we use Projects as concrete instantiations of contexts.

and links between documents are made based on their occurrence in the events. A special set of nodes is added to the network as a "context-identification-node or 'node-of-interest' (NOI). These nodes have connections to contextual elements such as topics and location. These connections can be made manually by the user, or automatically by analysing folder content when the NOIs come from harvested file folders – in the case of the Happy Worker App these nodes are the projects we distinguish.

- The model assigns an activation level to each project in each event block. The activation level is then plotted in a line chart where each line represents a project. The X-axis is the time and the Y-axis the activation level. In this deliverable, the activation levels are a mirror to the user, to show how he spent his time. In D3.7, the activation of contexts will be used for recommending documents and web pages that are relevant for the user's current context.

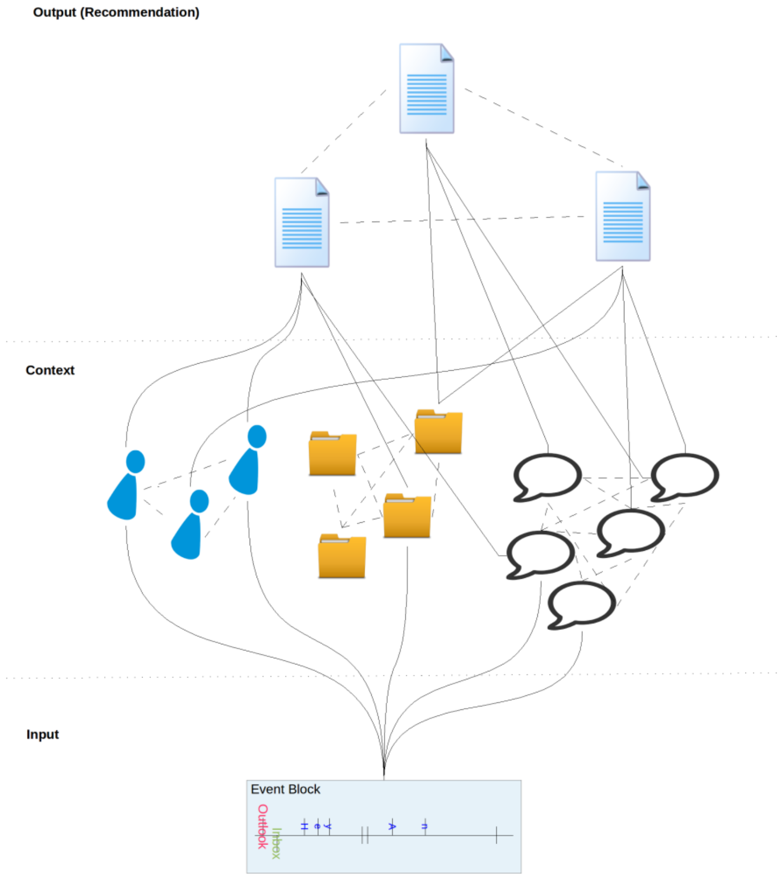

Figure 2. Interactive Activation Model for Context

3 Technical specifications

The purpose of this section is to document the (required) changes in the software, mainly as a reference for future developers.

3.1 Change log for D3.2 – Preliminary implementation

1. Give insight in the distribution of work time over task activities

- ✓ Branch StressTool (software implementation by Saskia Koldijk for SWELL WP3) to HappyWorkerApp in subversion (svn).
- ✓ Read data from local uLog files instead of RabbitMQ bus
- ✓ Let the user choose select the uLog file (or use default demo file if no file is chosen)
- ✓ Merge taskRecognition with HappyWorkerApp: output classified task activities in text files that can be read by the dashboard
- ✓ Add visualization tab 'Activities' to the dashboard (as Gantt chart)
- ✓ View the historical data in the correct week in the dashboard (Previously, all data was saved for today's date, even if the data was from another date)
- ✓ Change 'show this week'-button to 'show active week'-button (for demo purposes)
- ✓ If the user is not known in the Weka file header.arff, add the name
- ✓ Implement ULogReader as real time reader for reading a uLog file that is currently active (uLog running, XML open-ended)
- ✓ Implement and aggregate missing features in FeatureVector for taskRecognition:
    - numberSpecialKeys: in uLog data as separate event "Special key"; aggregation needed for all versions of uLog
    - numberBackspaces: in uLog data as separate event "Special key" with KeyboardValue=Backspace; aggregation needed for all versions of uLog
    - numberSpaces: in uLog data as separate event "Special key" with KeyboardValue=Space; aggregation needed for all versions of uLog
    - numberSwitches: in aggregated event "Application switches" (uLog 3.3)
    - numberScrolls: not available in uLog data
- ✓ Aggregate features for uLog 3.2 output[3]:
    - numberClicks: in aggregated event "Mouse Events Statistics" (uLog3.3)
    - numberCharacters: in aggregated event "Keyboard Events Statistics" (uLog3.3)
    - applicationPercentages: in aggregated event "Application percentages used" (uLog 3.3)

2. Give insight in the distribution of work time over projects

- ✓ Implement of Interactive Activation (IA) model
- ✓ Add IA model implementation to HappyWorkerApp package
- ✓ Merge ContextAnalyzer in HWA-code
- ✓ Merge uLog reader in HWA-code
- ✓ Add visualization of contexts over time
- ✓ Use correct week from uLog data

3.2 Requirements for D3.3 – Final and updated implementation

1. Give insight in the distribution of work time over task activities

[3] For the purpose of using the SWELL@work experiment (fall 2012) as demo data. These data were created with an older version of uLog (3.2) in which events were not yet aggregated per 5 minutes.

- Implement real-time classification. The uLogreader already works real-time: it can read from an active uLog file and monitors it for new incoming data. Every time the uLogreader gets new incoming data for a 5-minute time span, we want to classify them while the user is working, so that the dashboard always shows the most recent activities when it is opened.
- Use classifier confidence scores as threshold for user input: ask for a label when the classifier is less than X % certain. This user feedback is then taken as additional training data for the classifier.
- Retrain in real-time setting after 6x5 minutes
- Remove "username" as feature for taskRecognition; add username to filename of train and test files instead
- Implement the possibility to add new task activities
- Make the calculation of averages of events per minute more flexible. Now, custom events contain aggregated data over 5 minutes. So in setting the average number of clicks per minute the value for the custom event is divided by 5. If we would aggregate over longer or shorter time spans, this is not robust anymore.
- Remind the last active application in the previous block. If no application shift occurs in the current block of five minutes, save the last active application in the previous block with 100% use.

2. Give insight in the distribution of work time over projects

- Improve query extraction
- Test robustness
- Unit tests
- Add PDF documents to model
- Do we want to use the caption?
- Give the user the option to view the term profile of a project in the form of a term cloud, such as in Deliverable D3.6: Terminology extraction for user modelling.

General

- Compatibility with uLog 3.3
- Unit tests
- In the current version of the software, all data (the sensed interaction data and the analyzed labels) are stored locally. In the future, the software will communicate with Commonsense[4] by Sense Observation Systems.

4 Conclusions

We implemented two algorithms in a Java package called HappyWorkerApp: the recognition/classification of task activities (e.g. write report, search information) and the recognition of meaningful contexts (projects/topics). The user interface is a task bar icon and dashboard that displays two graphs: a Gantt chart for the task activities and a Activation chart for the contexts. We demonstrate the software with data from the SWELL@work experiment in the fall from 2012.

[4] https://www.commonsense-dashboard.com/#/

5 References

- Bakker, J., Holenderski, L., Kocielnik, R., Pechenizkiy, M., & Sidorova, N. (2012). Stess@Work: from measuring stress to its understanding, prediction and handling with personalized coaching. Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, IHI '12 (pp. 673–678). New York, NY, USA: ACM.
- Koldijk, S. (2011). Look what you've done! Task recognition based on PC activities. Masters' thesis. Radboud University, Nijmegen, The Netherlands.
- Koldijk, S., Sappelli, M., Neerincx, M., & Kraaij, W. (2013). Unobtrusive monitoring of knowledge workers for stress self-regulation. In: Proceedings of UMAP 2013 (Rome, Italy, 10-14 Juli 2013).
- McClelland JL, Rumelhart DE (1981) An interactive activation model of context effects in letter perception: I. an account of basic findings. Psychological review 88(5):375
- Ruff, J. (2002). Information Overload: Causes, Symptoms and Solutions. Harvard Graduate School of Education (pp. 1-13).
- http://www.noldus.com/ulog/features
- SWELL D3.1b Functional design of the reasoning components in the SWELL@work modules – update https://ecity.tno.nl/sites/COMMIT_SWELL/Documenten/20131003%20SWELL%20D3.1b_functional%20design_FINAL.pdf
- SWELL D3.5 Dataset of activity and interaction streams https://ecity.tno.nl/sites/COMMIT_SWELL/Documenten/SWELL%20D3%205_final.pdf
- SWELL D3.6 Terminology extraction for user modelling
- https://ecity.tno.nl/sites/COMMIT_SWELL/Documenten/20130405%20SWELL%20D3.6_Terminology%20extraction%20for%20user%20modelling.pdf
- SWELL D3.10 SWELL@work applications https://ecity.tno.nl/sites/COMMIT_SWELL/Documenten/D3.10%20SWELL@work%20applications%20v1.0.pdf