D3.7 Context and Task sensitive information filtering and sharing agent

Project	SWELL
Project leader	Wessel Kraaij (TNO)
Work package	3
Deliverable number	3.7
Authors	Maya Sappelli, Suzan Verberne
Reviewers	Joris Janssen, Wessel Kraaij
Date	12-6-2014
Version	1.0
Access Rights	COMMIT/ public
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 the deliverable D3.7 Context and Task sensitive information filtering and sharing agent, we adapted the implementation of D3.2 Activity Classification to make the context identification (project recognition) real-time. Additionally, we adapted the implementation to make it suitable for document recommendation. This functionality will eventually be integrated in the Happy Worker App as part of M4 Search assistant in D3.1b:

1. Advice: Pro-actively suggest relevant search results at the moment of information

In this deliverable we describe the implementation of this functionality in the Happy Worker App software. A short demonstration video of the software can be found at: https://www.dropbox.com/s/mg954fp4d08isgq/RecommendationsExample.mp4

Contents

1 Introduction

In the project SWELL we aim to support knowledge workers in their daily life. One aspect is their working life, which is addressed in WP3. With the increasing amount of information knowledge workers have to handle, they can get overwhelmed easily; a phenomenon referred to as `information overload' (Bawden 2009). This can cause unnecessary stress. According to the Resources-Demands model , well-being at work can be improved by addressing both the resources that a user has available; for example by improving his motivation, or by reducing the demands that the job requires. One method for reducing the demands is by providing the user with the information he needs so that the user does not need to spend precious time on finding that information himself. Additionally this can serve as an extended memory function for the user, as information that the user has seen before, but has forgotten about, could be re-used when it is relevant for new work. The recommendation of information that is relevant to current activities is the focus of this deliverable (D3.7 Task Sensitive Information Filtering and Sharing agent).

In deliverable D3.2 we have worked on a method to give users insight in the time he/she spends on each project. This method predicts the active project based on information coming from a key logger tool on the user's computer. The method is a network-based approach which can be used for various purposes. In this deliverable we adapt the network-based approach to give real-time recommendations of information sources.

In Section 2 we give a functional description of the software, followed by technical specifications and requirements in Section 3.

1.1 Context of this deliverable

The software developed for this deliverable is an alternative version of the software delivered in D3.2. The main difference of D3.7 compared to D3.2 is that part of the functionality (project recognition) is now executed in real time. Additionally, instead of the interface that provides an overview of the day, the interface is now a screen where the user can access recommended documents and webpages in real-time and see which active project the software has detected. D3.3 will be an update of D3.2 where as an addition to real-time project recognition, real-time task recognition will be developed. Furthermore D3.2 will include possibilities to provide feedback to the machine learning algorithms.

1.2 Demonstration of the software

Figures 1-3 show screenshots of the Recommendation Center. Figure 1 displays the settings screen, where the user can add and delete the projects that he or she is interested in, and the corresponding directories. Figure 2 shows the main recommendation application, where the list of suggested documents is presented, together with the project that is recognized as currently being active. Finally, Figure 3 shows the graph activation, that shows how the activation of the various has developed over time. This can be used by the user to determine how many hours he spend on each project. A short demonstration video of the software can be found at:
https://www.dropbox.com/s/mg954fp4d08isgq/RecommendationsExample.mp4

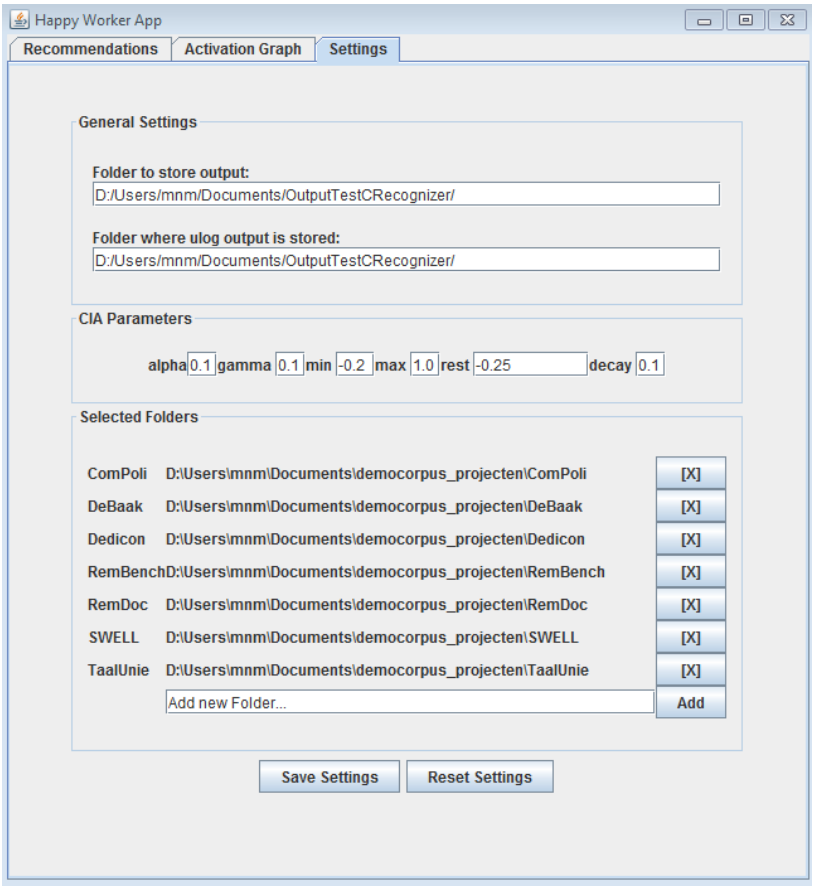

Figure 1: Settings Screen.

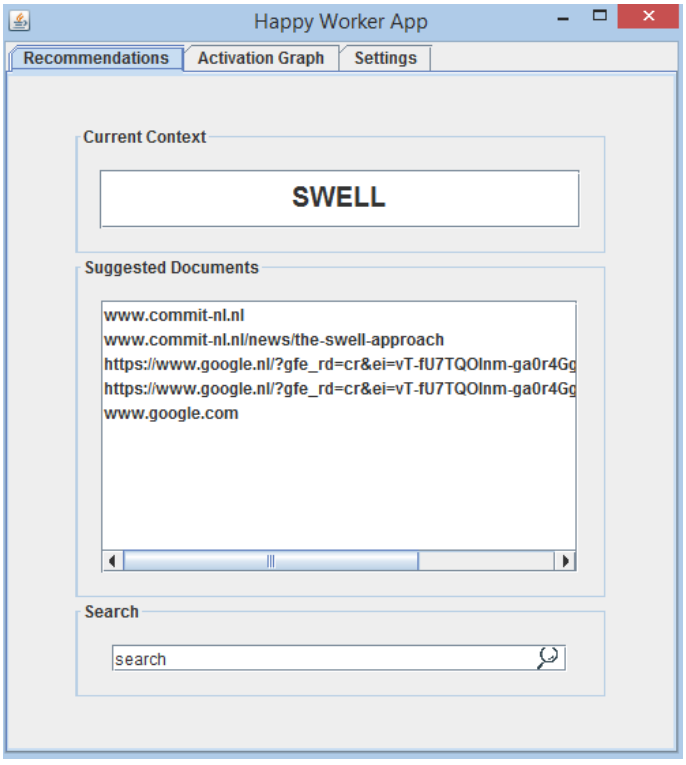

Figure 2: Recommended Documents for Project SWELL.

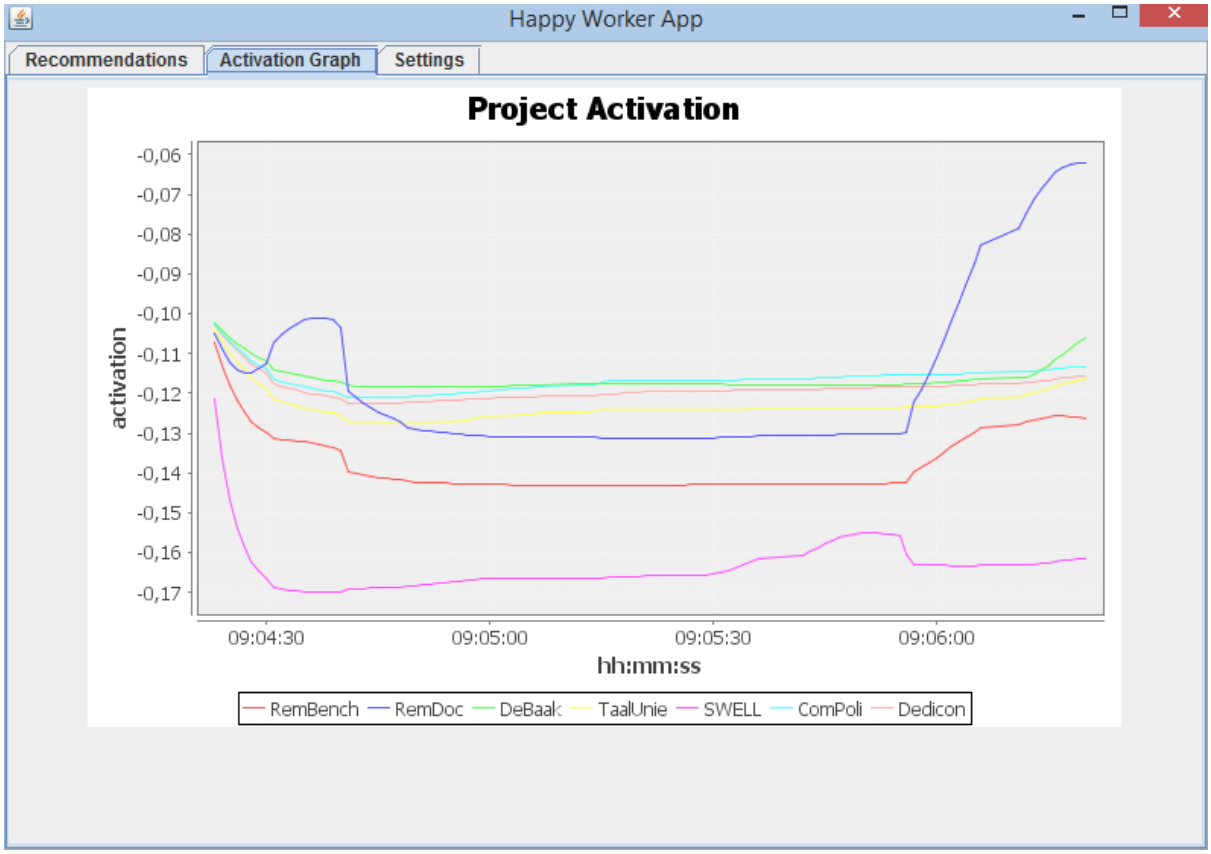

Figure 3: Overview of activated projects over time.

2 Functional Description

2.1 Main functional requirements

The main functional requirements for the recommendation of documents and webpages are:

- F1: The software unobtrusively collects objective behavior data (computer logging).
- F2: The software provides real-time access to documents and webpages relevant to the current activities. Our idea is that it is better to let the user decide to access the recommendations rather than to push recommendations.

For this deliverable (D3.7), we implemented one functionality from the “Search Assistant” (see M5 in D3.1b) in the Happy Worker App: Pro-actively suggest relevant search results at the moment of information need.

The backbone of this functionality is Maya Sappelli’s implementation of a feed forward network for context recognition (see D3.2). The functionality is implemented in Java and will later be included in the SWELL package *HappyWorkerApp* (HWA). HappyWorkerApp is a continuation of the StressTool package developed previously in SWELL WP3 (no formal deliverable).

The recommendation functionality will have a separate interface consisting of a task bar icon and a pop up screen that opens when the icon is clicked. The screen displays the currently detected project and a list of recommended documents and webpages. Both the list and detected project is

updated regularly. The list is updated with a delay as real-time update would most likely change the list of recommended items too fast, making it difficult to select recommendations in time.

2.2 Information Recommendation

For recommending relevant documents and webpages, first we need to have a basic idea of which projects a user works on. 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 projects of one co-worker and mimic some work as if she was working on the RemDoc project (a project about art).

The recommendation of the documents is done using an unsupervised model that has been developed to track *user context.* It is based on the interactive activation (IA) and competition model by McClelland and Rumelhart (1981). This is a feed forward network with back propagation 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. The project directories of step one provide an initial set of documents that can be suggested when no events have been seen yet.

For the context recognition, the uLog events are grouped in blocks, where each switch of application defines the start of a new event block. This merging of events is executed real-time. 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 if it does not exist yet. Furthermore, 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 and links between documents are made based on their occurrence in the events.

The model assigns an activation level to each node in each event block. To determine which documents are relevant to recommend, we look at the activation in the document level. The 5 document nodes that are most highly activated are the nodes that end up in the list of recommendations. The number of documents that can be suggested can be adapted.

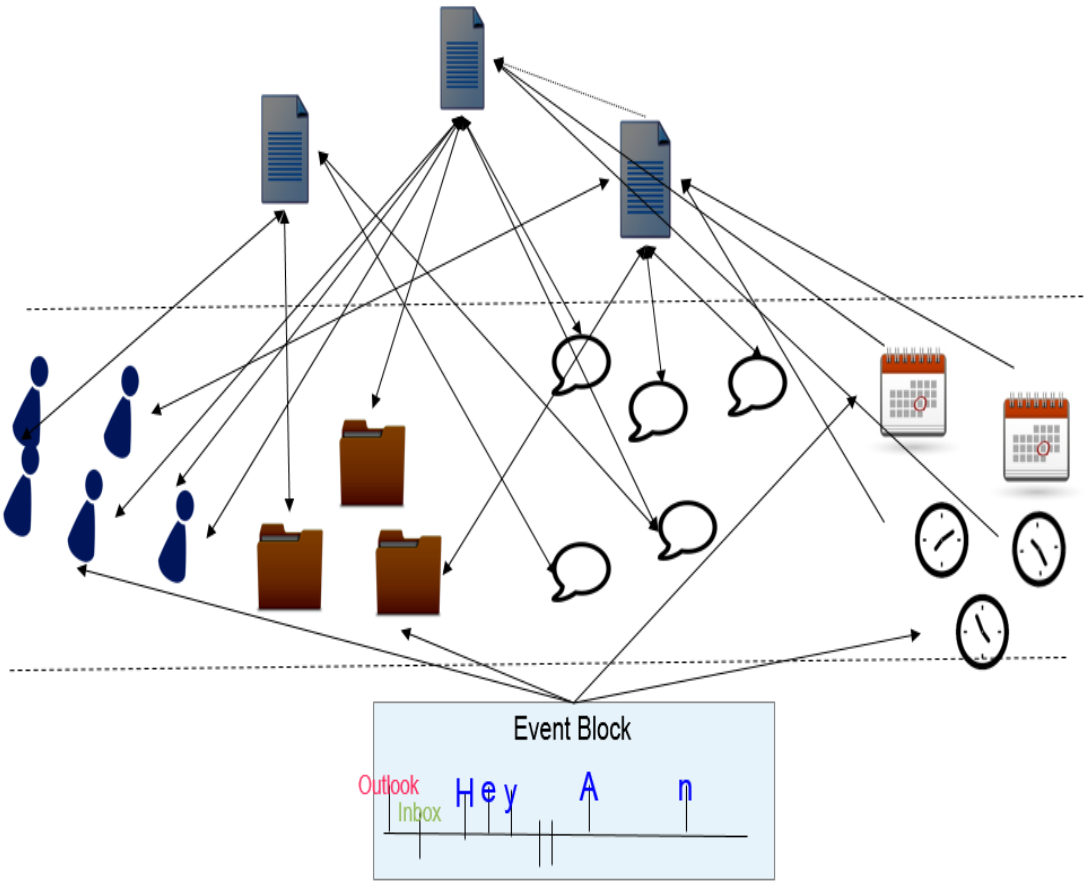

Figure 4: Schematic overview of network underlying the recommendation software

3 Technical Specification

3.1 Change Log for D3.7

- General improvements to the network method by Maya Sappelli are made (bug fixes)
  - Dealing with multiple links in one event block
  - Stability in results is increased by improving LDA recognition
- Context Detection is made real-time
  - Real-time merging of events
- Functionality to read activation of document nodes in the network is added
- Interface to present an ordered list of recommended documents is added
  - Including interface to adapt settings and interface to view context recognition over time

4 Discussion

During the development of the application we noticed a couple of problems that will need to be dealt with in the future. First and foremost, the nature of personal information management is complex. Many people save multiple versions of a document on their file system, such as a PDF and a .doc version of the same file, or multiple versions of the same file that is adapted over time (v1, v2 etc.). As of yet our recommendation system is not capable of handling these duplicates adequately.

A possibility is to add some serendipity factor to determine which documents are selected. This would make the suggested document list more diverse.

Another point that has proven complex is when to update the list of recommendations. Currently the recommendation list will be generated when a user switches to another application or tab. This means that when a user works on a document for a longer period of time without switching, no updates will be generated. However, this will only be a problem in the unlikely case that the recommendation screen is constantly visible and that the user does not switch in the meantime.

In some settings it is imaginable that it is difficult to obtain a stream of events from a user. This can be because installing a tracker is prohibited, or simply because the user is too often away from his keyboard. Even when a user is away from his keyboard, for example because he is in a meeting, it can be interesting to recommend him documents related to the meeting when he gets back. However, how can we use the proposed method when there is no stream of events? The basic purpose of the method is not to go from a set of events to document recommendations. Instead the goal is to divide an input into its parts to use the individual elements rather than the complete input to get a suitable recommendation. Therefore, any input that can be divided into a set of elements relevant for documents (e.g. social, topical, locational and time aspects) can be used in this approach.

Although the activation function in the model makes it well suited for streaming data, it is not a requirement. Theoretically the input to the model could just as well be a stand-alone message (Sappelli et al. 2014) instead of an event-block. This e-mail message could be used as input to find documents relevant to that message. Another possibility would be to create a user profile of the knowledge worker based on his written documents (Verberne et al. 2013). Ideally this profile could be separated into several facets in which for example collaborations can be distinguished as well as work-home distinctions. For each of the facets, relevant documents could be generated based on the elements that define a facet. However, these recommendations will be much more static as no real-time information is taken into account.

5 Conclusion

We improved the implementation of recognition of real-time contexts (projects/topics) of D3.2 to make the recognition real-time. Additionally we implemented functionality to suggest relevant documents to the user in real-time. The user interface is a small window with 3 tabs. One tab shows the recognized active project, together with a list of 5 document suggestions. The second tab shows the activation chart of the projects to provide the user with an overview of his day, and the last tab allows the user to adapt the projects he wants to include in the recognition as well as some other settings. We demonstrate the software with some project examples provided by Suzan Verberne and some mock up data.

6 References

- Bawden, D., & Robinson, L. (2009). The dark side of information: overload, anxiety and other paradoxes and pathologies. *Journal of information science*,*35*(2), 180-191.
- McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings.*Psychological review*, *88*(5), 375. Sappelli

- Sappelli, M., Verberne, S., & Kraaij, W. (2014). E-mail categorization using partially related training examples. To appear in: *Proceedings of the 5th Information Interaction in Context Symposium*.
- Verberne, S., Sappelli, M., & Kraaij, W. (2013). Term Extraction for User Profiling: Evaluation by the User. In *Proceedings of the 21th International Conference on User Modeling, Adaptation and Personalization*
- SWELL D3.1b Goal recognition and activity classification – updated algorithm design https://ecity.tno.nl/sites/COMMIT_SWELL/Documenten/20130624%20SWELL%20D3.1b_functional%20design.pdf
- SWELL D3.2 Activity Classification https://ecity.tno.nl/sites/COMMIT_SWELL/Documenten/20131218%20SWELL%20D3.2_Activity%20classification_FINAL.pdf