D3.8 uLog: Task level analysis of computer interaction data

Project	SWELL
Project leader	Wessel Kraaij (TNO)
Work package	3
Deliverable number	D3.8 (software deliverable)
Authors	Leon Wiertz (Noldus), John Schavemaker (TNO)
Reviewers	Marten van Sinderen, Martijn Vastenburg
Date	5/4/2013
Version	0.5
Access Rights	SWELL project members
Status	Reviewed

SWELL Partners:

Ericsson, NCSI, Noldus, Novay, Philips, TNO, Radboud Universiteit Nijmegen, Roessingh Research and Development, Universiteit Twente,

SUMMARY

This document describes a software implementation of a key-logging tool that captures computer user interactions patterns from input devices like keyboard and mouse and allows identification of the main user tasks by aggregation of this sensor data. The accompanying software to this document can be installed on computers with Windows OS but needs a SWELL server to fully operate.

This deliverable D3.8 is a new development (from scratch) based on the existing Noldus uLog key-logging tool. The update with respect to the previous versions of uLog includes a SWELL plugin that adds new functionality for asynchronous and synchronous communication using a message queuing implementation, and sensor-based analysis of computer interaction data.

Contents

1 INTRODUCTION

Managing work and family responsibilities is often difficult and impacts the health and well-being of employees and the workplace performance. A healthy workforce is a happier and more productive workforce. Keeping people healthy and at work has obvious benefits: protection against financial hardship, promoting a better quality of life and allowing people to make the most of their potential. Conversely, being out of work can exacerbate physical and mental health problems and increase the chance of social exclusion. The objective of the SWELL (Smart reasoning systems for WEll-being and weLL-working) is to develop user-centric sensing and reasoning techniques that help to improve physical well-being (mostly in a private context) and to improve well-working (in a work context). Of course physical well-being influences well-working and vice versa. 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.

The SWELL balance board demo is a prototype of a reasoning and feedback system for individuals and groups that could improve well-working in a typical office setting. The demo combines some of the previous work and results of WP3 into one working prototype that combine basic sensing using key logging (uLog), an communication framework (AMPQ), aggregation and analysis of sensor data, and feedback strategies by visualization of data. The SWELL balance board will be based on this deliverable D3.8. This deliverable D3.8 provides a new software implementation of an intelligent key-logging sensor that is needed as the primary sensor for the SWELL Balance Board demo.

D3.8 is a new development (from scratch) based on the existing Noldus uLog key-logging tool. The update with respect to the previous versions of uLog includes a SWELL plugin that adds new functionality for asynchronous and synchronous communication using a message queuing implementation, and sensor-based analysis of computer interaction data. The message queuing is based on the RabbitMQ implementation of the emerging AMPQ protocol.

This document describes a software implementation of a key-logging tool that captures computer user interactions patterns from input devices like keyboard and mouse and allows identification of the main user tasks by aggregation of this user sensor data. The accompanying software to this document can be installed on computers with Windows OS but needs a SWELL server to fully operate.

In this document we describe the SWELL plugin for the uLog3 key-logging tool. The outline of this document is as follows: This introduction section gives a description of the context of the software deliverable, being the uLog3 key-logging tool and the SWELL balance board. Section 2 starts with a set of requirements (from the Balance Board demo) of the SWELL plugin. Sections 3 focuses on the functionality of the uLog3 tool, with in Section 3.4 a contextual overview of the balance board and the place of uLog3 within this setup. Section 4 ends with and an outlook to the requirements for valorisation of the uLog3.

1.1 uLog3

uLog3 is the innovative tool for automatically recording user-system interaction. uLog records more than just keystrokes, scrolls, and mouse clicks. You can see which button was clicked and whether or not it was double-clicked. On top of this, uLog3 can log individual characters as well as complete text

strings so you can read exactly what your test participant has entered. uLog3 tells you which applications are active, which menus are used in the user interface, which pages are opened in your web browser, and much more (see [1]).


Figure 1: uLog automatically records user-system interaction.

A previous version of uLog has been part of the Noldus product portfolio. This version of uLog however does not run on newer Operating Systems and is limited in its functionality with respect to goals set for SWELL. This document describes the functionality of a new version, uLog3, that has been written from scratch.
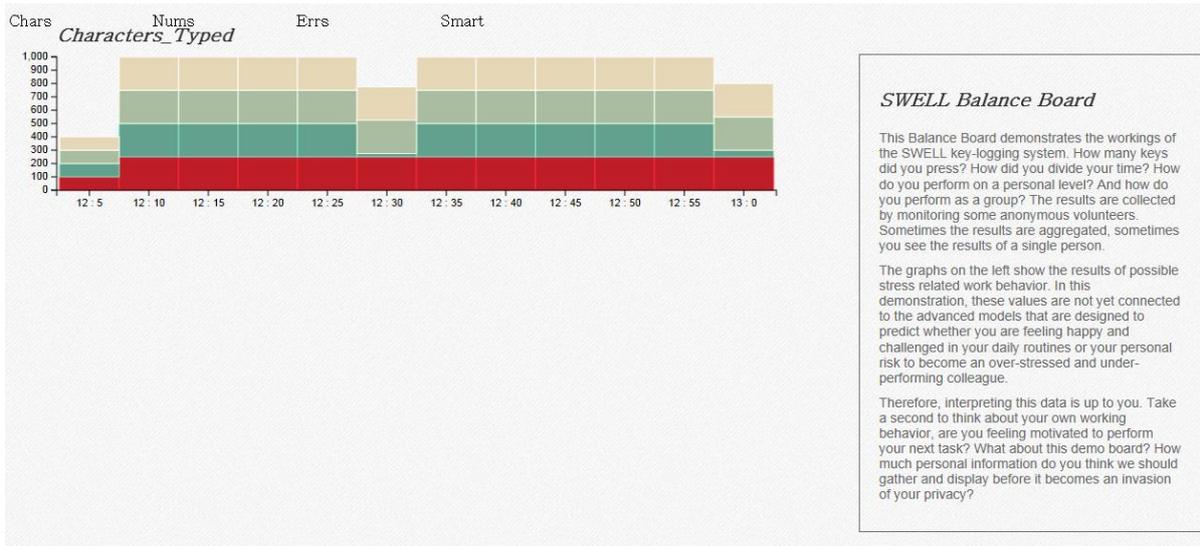
1.2 Balance board

The SWELL balance board demo combines some of the previous work and results from the SWELL project into one working demo prototype that combines basic sensing using key logging at multiple PCs of users in a group, one communication framework (RabbitMQ), central aggregation and analysis of sensor data, and feedback strategies (for individual persons and groups) by visualization of data at a central place (like the coffee corner).
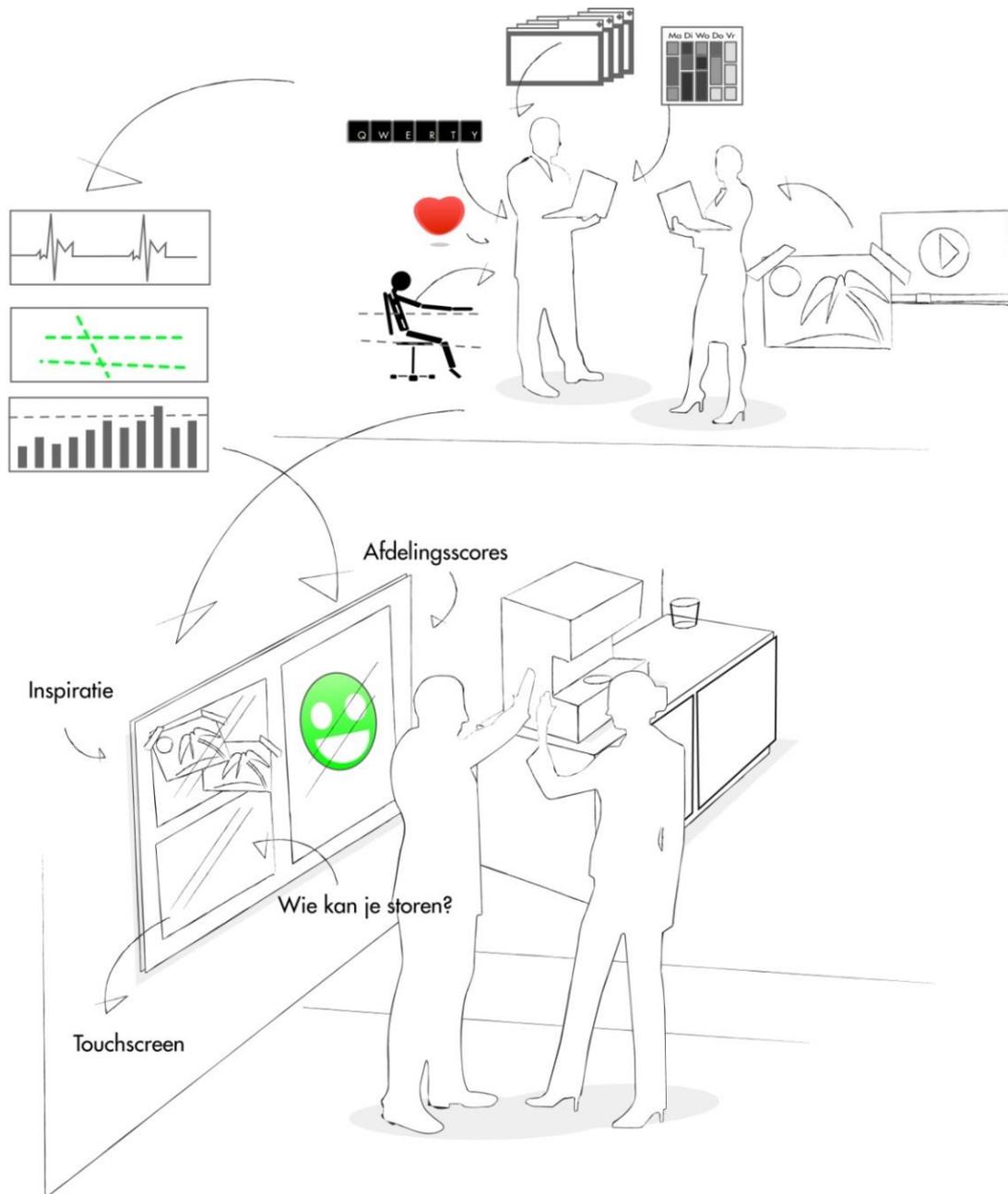
Figure 1: Visual depiction of SWELL balance board workings (in dutch).

The SWELL balance board is a large interactive screen that can be a placed at a central place at the office (for example the coffee corner). The board shows visualizations of the state or mood of the group or individual people working in that specific group. The balance board bears some similarity with the Helo.Wall from Fraunhofer [2]. The board shows graphical representations of human-computer interaction patterns based on sensors. The primary sensor is a key-logging sensor that captures key strokes, mouse movements and clicks together with information provided by the operating system: window titles, open applications, application switches, etc. The goal of the SWELL balance board is to make people aware of their computer interaction patterns by showing your sensor data in aggregated way, or by means of other feedback mechanism. To show results in a central place of a group allows comparison with working peers.

**Figure 2: An example graphical representation of characters typed on SWELL balance board.**

In order to enable multiple users, the balance board needs a communication means that is cross-platform and can connect multiple sensors from different users. The RabbitMQ[1] framework fulfils these requirements: it is a robust messaging framework for applications, that is easy to use, runs on all major operating systems, supports a huge number of developer platforms and is open source and commercially supported.
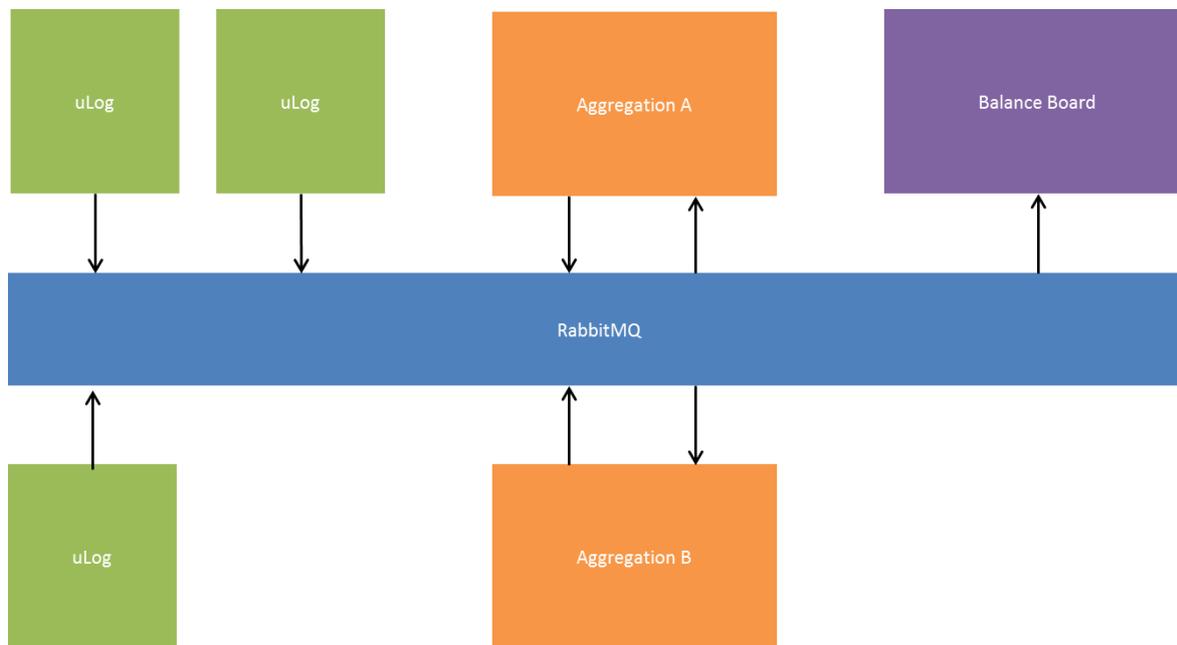
Figure 3. SWELL balance board server overview with RabbitMQ message bus.

In the balance board, different analysis modules will be implemented for individual users as well as groups that will map sensor data to user concepts via features. The current foreseen modules are: tasks & topics and stress. All modules need (aggregated) key-logging data to extract meaningful features to recognize user concepts. User concepts that are recognized can be used in the balance

board visualizations to show what users (or groups) are working, how they are experiencing that and how users are related to each other in terms of tasks, topics and workload.
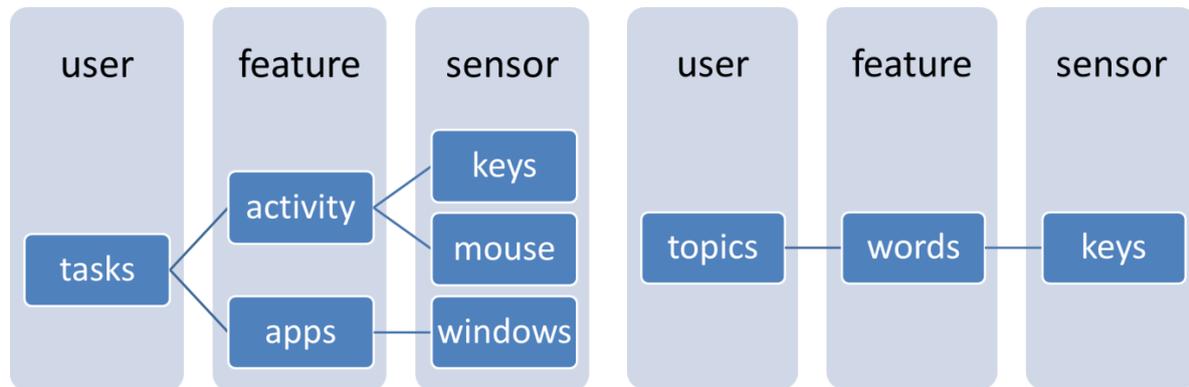
Figure 4. Balance board analysis module 'tasks & topics'.

The user-task module needs features that describe the current open applications (from open windows) and keyboard / mouse activity to identify the user's task(s). For example, when a Microsoft Word window is open and there is sufficient keyboard activity (alphanumeric and editing) the user task of 'writing a document' can be identified. At the same time relevant topics where the user is working on can be identified by means of extracting words from keyboard input and apply term-extraction methods (D3.6) to find the relevant terms that describe the user topics.

The stress analysis module relates human-interaction patterns to stress levels. From a key-logging sensors different sensor data can be used to extract features that correlate with stress. For example, increased keyboard and mouse activity with more than normal use of error keys and more frequent window switches can be signs of higher stress levels.
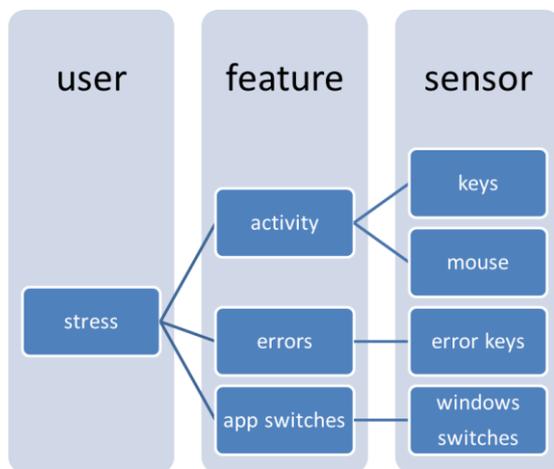
Figure 5. Balance board analysis module 'stress'.

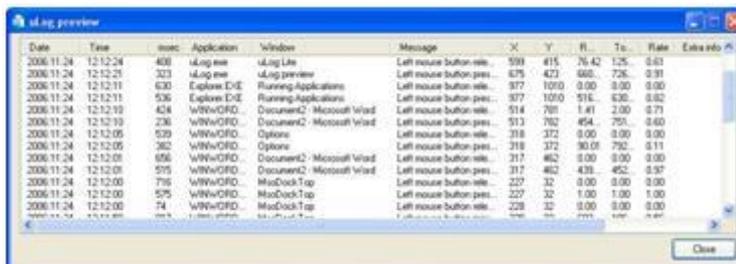# 2   FUNCTIONAL REQUIREMENTS OF ULOG3 SWELL plugin

The (functional) requirements of the uLog3 SWELL plugin stem from what the 'tasks & topics' and 'stress' modules of the SWELL balance board need in terms of sensor data and subsequent feature processing.

- SWELL plugin must be able to use RabbitMQ messaging
- SWELL plugin must be able to send raw sensor data ('events') in an asynchronous way (when events happen)
- SWELL plugin must be able to send aggregated sensor data in an synchronous way at fixed time intervals
- SWELL plugin must aggregate sensor data for a pre-defined interval of time
- SWELL plugin must send aggregated sensor data that contains:
  - amount of key presses
  - differentiation of keys (alphanumeric, error, shortcut, arrow)
  - amount of mouse movement
  - amount of mouse clicks
  - differentiation of mouse movements
  - differentiation of mouse movements
  - amount of applications switches
  - amount of window title switches
  - applications list plus percentage used
  - windows list plus percentage used

3   FUNCTIONALITY OF ULOG3

3.1   Description uLog3

From a functional point of view uLog3 can be divided in four distinct parts: The base logger, The Event plug-ins (consisting of Filter and Builder plug-ins), The Base output and the Output plugins. This paragraph describes these four parts in detail.

Figure 6. uLog3 example output.

3.1.1   Base logger

The base of uLog3 is the logger of keyboard, mouse and application events. The following events can be logged (the user is able to select for each of the following items whether or not the events should be logged):

- Mouse – Click

- Mouse – Double Click
- Mouse – Wheel
- Mouse – Drag
- Mouse – Hover
- Keyboard – String
- Keyboard – Special Keys
- Keyboard – Key Combination
- Keyboard – Single Character
- Other – Browser Address Changed (Internet Explorer and Firefox)
- Other – Window Activation
- Other – Message Box
- Other – Application Start
- Other – Application Exit
- Other – Focus Changed

3.1.2   Event plug-ins

As described in the previous paragraph, the base logger is responsible for gathering raw events with respect to keyboard, mouse and application handling. These are the base events. uLog3 can be extended with so called Event plug-ins to extend or limit the events being generated.

There are two types of event plug-ins. The first type are the filter plug-ins. These can be used to only send a subset of all generated events to the output. A filter plug-in that is by default available is an application filter. If this plug-in is active only events belonging to a -by the user pre-specified- list of application will pass.

The other type of event plug-ins are builder plug-ins. These plug-ins can generate new (higher level) events based on one or more of the base events described in the previous paragraph. For the balance board uLog3 has been extended with a builder plug-in that calculates averages over the key- and mouse movements of a user. The next paragraph will describe this plug-in in a bit more detail.

3.1.3   SWELL aggregator builder plug-in

For the balance board a new plug-in has been written. The functionality of the plug-in however is not limited to the SWELL project. The plug-in is a generic builder plug-in that can be used by any user of uLog3 if they need the data. The plug-in collects all the events gathered by the base logger and aggregates these to send averages of the usage in a given period of time. The following averages can be obtained:

- Average key presses per minute
- Average alphabetic characters per minute (a-z, A-Z)
- Average numeric characters per minute (including +, -, =)
- Average remaining characters per minute (the ones that do not belong to alphabetic characters or numeric characters)
- Average words per minute
- Average shortcut keys per minute (Ctrl+[Key], Alt+[Key])
- Average arrow keys per minute
- Average Error keys per minute (Ctrl+Z, backspace, delete)

- Average mouse events per minute
- Average mouse clicks per minute (all, left, right, and/or middle)
- Average double mouse clicks per minute
- Average mouse scroll events per minute
- Average mouse scroll distance per minute
- Average application switches per minute
- Average window title switches per minute (to detect tabpage changes in a browser for instance)

Furthermore the user can get also the following figures:

- the Alphabetic character/Key ratio
- Error/Key ratio for the keyboard events
- percentage used per application
- percentage used per window title

3.1.4   Base output

The base output is the central place where all events from the base logger and the event plug-ins go to. In this component the user can choose what information per event should go to the output plug-ins. The following fields can be selected by the user.

- Date
- Time
- Milliseconds
- Event Type
- Event Description (the event in a self-explaining sentence)
- The name of the application
- The window text
- Mouse – Button pressed
- Mouse – Cursor X position
- Mouse – Cursor Y Position
- Mouse – Wheel turn direction
- Mouse – Wheel turn amount
- Mouse – Relative distance
- Mouse – Total distance
- Keyboard – Value (character, key combination, strings)
- Keyboard – Modifiers (Ctrl, Alt, Shift, Win ,Caps)
- Control Information – Type
- Control Information – Caption
- Control Information – Value (if applicable)
- Control Information – Password Control (true or false)
- Control Information – Control Framework
- Control Information – Control class
- Control Information – Snapshot Image (with or without mouse cursor(
- Control Information – Target Information

- Other Information – Clipboard Text
- Other Information – Browser URL
- Other Information – Custom Events (Generated by Builder plug-ins)

3.1.5 Output Plug-ins

The information mentioned in the previous paragraph is not stored anywhere by default. Instead uLog3 uses an output plug-in mechanism to enable the user to output the data in specific formats. uLog3 comes standard with three output plug-ins:

- XML File Writer – Writes the output information to a XML file for further processing by other application
- ODX File Writer – Writes the output to an ODX File that can be imported into The Observer XT (a versatile manual logging and integration software package made by Noldus).
- Preview – A Live dialog that displays all the events that are being logged.

3.1.6 SWELL AMQP Output plug-in

For the balance board another plug-in has been written. This plug-in sends the output data over an AMQP[2] bus to listening clients. This is a SWELL specific plug-in. Although the data is send to a RabbitMQ server and anyone might listen to this. The format of the message send to the RabbitMQ is defined specifically for the balance board. The plug-in enables the user to send all the uLog3 data to the balance board next to the default output options that have mentioned in the previous paragraph.

3.2 FUNCTIONAL CONSTRAINTS

uLog3 is a logger application that can be used on the Windows platform only. It is meant to log the data of one specific user at the time and it will not aggregate over multiple users. The application cannot be installed silently; uLog3 will always notify the user (at start recording and on a periodic time base) that is recording the keystrokes and the mouse moves of the user.

3.3 FUNCTIONAL RESTRICTIONS

The current version of uLog3 is still in prototype phase. The intention of Noldus is that the tool will be made available commercial when all functionality has been implemented and tested. Until that time the software can be used free of charge. The only restriction on the software is that it can only be used until a specific end date. This end date is currently set to December 31, 2013.

3.4 FUNCTIONAL CONTEXT

In the beginning of this document a short introduction has been made to the Balance Board, a measurement environment build within the SWELL project. uLog3 is a data collection tool and on this moment nothing more than that. One can regard it as a sensor that senses data rather than a standalone software tool. In context, such as within the balance board setup (Figure 7), uLog3 can be of added value and give insight into the computer behaviour of one or more office workers.
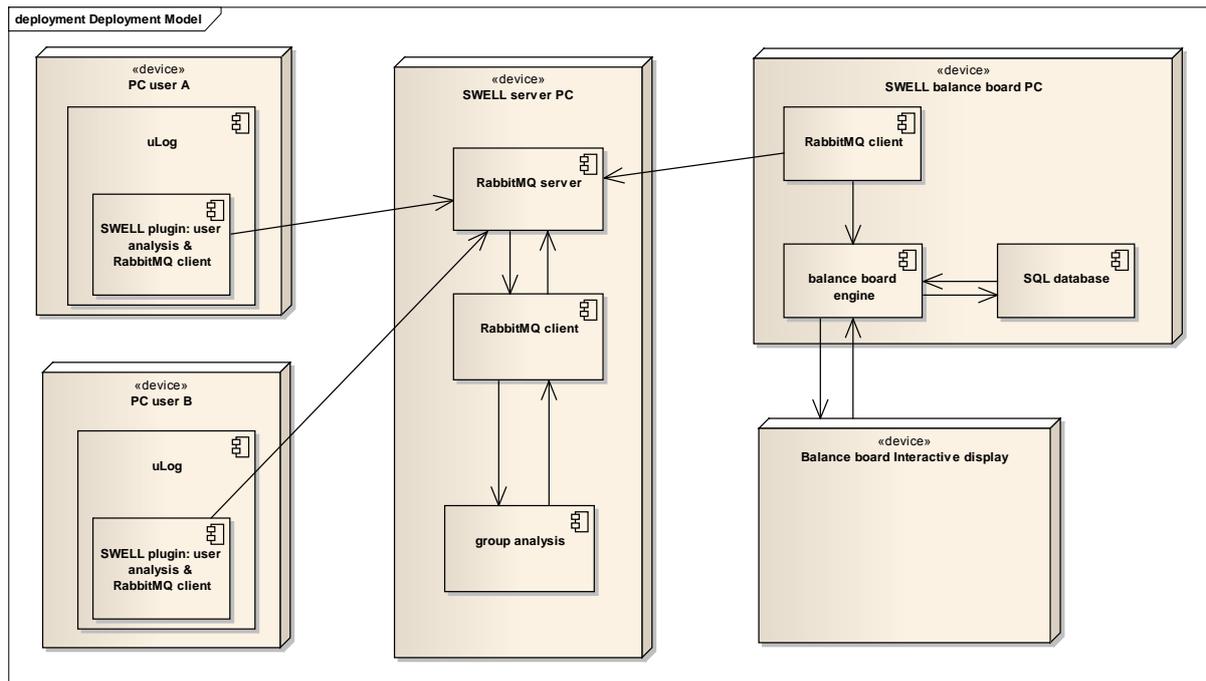
Figure 7. uLog3 with SWELL plugin in deployment model of SWELL balance board.

4 Outlook

uLog3 in its current state is already useful within the SWELL project and especially within the Balance board demo application. Noldus wants to elevate the prototype as is to a tool that can be sold on a commercial base. To achieve this goal the following requirements will be implemented:

- System Requirements - Run on Windows 7 and Windows 8
- System Requirements – Standalone setup
- System Requirements – 'Server' PC with one or more connect uLog3 Computers. Including Remote Start, Stop and configuration.
- Interaction Design - Dialog resizable + logic GUI
- Interaction Design - Proper defaults
- Logging Events - Mouse Move
- Logging Events - support latest interaction modalities (touch, speech, etc.).

5 References

[1]   http://www.noldus.com/ulog/features
[2]   Th. Prante, C. Röcker, N. A. Streitz, R. Stenzel, C. Magerkurth, D. van Alphen, D. A. Plewe: Hello.Wall – Beyond Ambient Displays. In: Peter Ljungstrand, Jason Brotherton (Eds.): Video Track and Adjunct Proceedings of the 5th Intern. Conference on Ubiquitous Computing (UBICOMP'03), Seattle, Wash., USA, Oct. 12-15, 2003

---

1 http://www.rabbitmq.com

2 http://www.amqp.org/