

# Inputlog: A Multimethod Approach Describing Cognitive Writing Processes Using Keystroke

## Logging

Mariëlle Leijten<sup>°\*</sup> and Luuk Van Waes<sup>°</sup>

<sup>°</sup> *University of Antwerp, Belgium*

<sup>\*</sup> *Flanders Research Foundation, Belgium*

Prinsstraat 13, BE 2000 Antwerp, Belgium

+ 32 3 265 50 72 | [marielle.leijten@ua.ac.be](mailto:marielle.leijten@ua.ac.be) | [www.ua.ac.be/marielle.leijten](http://www.ua.ac.be/marielle.leijten)

Manuscript - Please, do not cite

*Abstract:* In the field of writing process research keystroke logging programs have become instrumental in the identification and understanding of the cognitive processes and strategies involved in writing. In this article we describe a logging program called Inputlog (freely available via the internet to the research community). Inputlog consists of four modules: (1) a data collection module that registers digital writing processes on a very detailed level; (2) a data analysis module that offers basic and more advanced statistical analyses (e.g. text and pause analysis); (3) an integration module that allows data merging between data files from other observation tools; and (4) a playback module that replays the recorded writing session. This article opens with an overview of research methods in writing research with a particular focus on keystroke logging. Next we describe the general and technical features of Inputlog. In the second part of the article we elaborate on the research objectives that can be addressed with Inputlog, either in isolation or in combination with various other research techniques. We describe a number of research projects that have been conducted in the workplace and in an educational context. To illustrate the methodological complementarity of Inputlog, the final section discusses four different studies in which keystroke logging was combined with other observation methods (thinking aloud protocols, speech recognition logging, usability logs, and eyetracking data). We conclude the paper with a preview of plans for further developments in the domain.

*Keywords:* *Inputlog, keystroke logging, writing processes, eyetracking, writing, reading, research methods*

## Introduction

With the emergence of digital media, the importance of writing as a human activity is beyond question. There has never been a decade in which so many people – with a wide variety of backgrounds and of all ages – have produced written texts. Most of these texts are produced on a computer (e.g. word processor, email or chat application) or by using another technological device (e.g. mobile telephone or smart phone), serving different communication and social purposes. Although the general characteristics and cognitive activities that underlie writing are fundamentally the same as in earlier years, these contexts create new challenges and certainly also new opportunities for writing researchers to investigate online writing behavior.

In order to better identify and understand the strategies governing the dynamics of writing, several research methods have been developed to conduct online writing research (Alamargot, Chesnet, Dansac, & Ros, 2006; Latif, 2009; Sullivan & Lindgren, 2006; Luuk Van Waes, Leijten, & Neuwirth, 2006; A. Wengelin, et al., 2009a). These different methods enable the researcher to study the complexity of writing processes from different perspectives. To observe writing processes on a computer, keystroke logging programs have been developed (for a review, see Latif, 2009). These programs log every keystroke activity together with a time stamp and offer different types of process analyses to reconstruct and describe text production processes.

In this article we briefly describe a new writing research instrument, called Inputlog. To better frame this instrument, this article first presents a brief overview of the most widely used research methods in writing research, with a focus on keystroke logging programs. Then the

different components and characteristics of Inputlog are presented. We illustrate its use in different research contexts and for different purposes: professional writing in the workplace, writing in an educational context, writing development, and writing mechanics. Because one of the strengths of the program is its open architecture, we also focus on the integration of Inputlog data with the output of other research techniques. In doing so, we want to show the richness of combining research methods and the opportunities it creates to address more complex and interdisciplinary research questions. In conclusion, prospects for further research and further development of the research instrument are discussed.

### **A brief overview of research methods in writing research**

The shift in writing research from focusing on the product approach to the process approach in the early 1980s, also resulted in the introduction of a wide range of new research methods in the domain. Work by Flower and Hayes (L. Flower & J. R. Hayes, 1980; Hayes & Flower, 1980) and Matsushashi (1981; 1982) are considered milestones in this evolution; by asking writers to produce a thinking aloud protocol or by videotaping writers in action, they were the first to focus on the temporal aspects of written text production. At the same time, a shift from pen&paper or typewriter as the most common writing tools to the computer took place, resulting in a broader use of technological instruments and computer programs to observe writing processes.

One of the main challenges for writing researchers is to adequately collect information to describe the (cognitive) complexity of the writing activity. Writing involves an orchestration of different subprocesses, traditionally divided in planning, transcription and revision (Hayes, 1996; Hayes & Flower, 1980). These subprocesses are characterized by a high degree of non-

linearity and constantly interact with each other during writing in iterative cycles. As in oral speech production, the number, length and location of pauses are considered important resources for identifying cognitive complexity because they index of production flow interruptions. Moreover, in contrast with oral speech the physical presence of text allows writers to read, evaluate and revise the text produced so far. In computerized environments these changes to the text are not visible any more in the end product. Online research methods should enable us to carefully monitor these different aspects of the writing behavior.

In their classification of writing research methods, Janssen, Van Waes and Van den Bergh (1996) distinguish between synchronous and asynchronous observation methods and direct and indirect observation methods. Synchronous observation methods gather information about cognitive processes during the writing process itself. With asynchronous observation methods, cognitive data are gathered after the writing act. On the other axis the methods are classified along a continuum between *directness* and *indirectness*. *Directness* refers to observation methods that claim to provide relatively direct evidence about writing cognition. *Indirect* research methods are used to make inferences about human cognition from process or product characteristics.

Table 1

Classification of writing observation methods (based on Janssen, Van Waes & Van den Bergh, 1996)

As Table 1 shows, the category of direct research methods mainly represents methods in which the writers themselves are asked to actively articulate what goes through their mind either during writing, or – in the case of a retrospective or prompted protocol – after the writing activity (Greene & Higgins, 1994; Smagorinsky, 1989; Van den Haak, De Jong, &

Schellens, 2003). The advantage of protocols is that the researcher gets a direct account of the writers' thoughts which can then be coded post hoc on the basis of classification schemes (Flower & Hayes, 1985; Krings, 2001). These direct research methods are widely used in writing research, and they are very suitable for identifying controlled writing processes of which the writer is actively conscious. Of course, one should take into account the reactivity of these methods (Janssen, et al., 1996).

In the category of indirect research methods, a wide variety of observation techniques is listed. *Video observation*, either analogue (e.g. Matsushashi, 1982) or digital on screen (e.g. Degenhardt, 2006), is one way to observe writing as it develops. The advantage is that it is not really intrusive, but on the other hand, it often requires very intensive coding afterwards (depending on the research objectives). The *double task method* is mainly used to assess the cognitive cost of the different subprocesses that take place while writing (Olive & Kellogg, 2002). For instance, by asking writers to respond to an auditory prompt as quickly as possible during text production, it is possible to deduce variations in the cognitive load associated with the different tasks that are interrupted on the basis of the reaction time (Levy & Ransdell, 1994; Olive & Kellogg, 2002; Piolat, Kellogg, & Farioli, 2001; Piolat, Olive, & Kellogg, 2004). Recently, more technical observation methods have been introduced in writing research, viz. eyetracking, Evoked Visual Potentials (*EVP*) and Functional magnetic resonance imaging (*fMRI*) (Roger Johansson, Wengelin, Johansson, & Holmqvist, ; Richards, et al., 2009; Luuk Van Waes, Leijten, & Quinlan, in press; Å. Wengelin, Leijten, & Van Waes, ; A. Wengelin, et al., 2009a). Thanks to the new technical developments in eyetracking, for instance, it is now possible to collect detailed temporal data that also tell us 'what' writers are reading or looking at during pauses, and how their visual behavior relates to other processes of text production.

This article mainly focuses on keystroke logging as a way to synchronically collect fine grained writing process data (for an overview, see Latif, 2009; Sullivan & Lindgren, 2006). In the next section we will provide more detailed information about this research method. As Alamargot et al. (2006) explain, each of these methods has its advantages and limitations. For instance, verbal think aloud protocols cannot reveal processes that are too automatic, and of which the writer is only unconsciously aware. The combination of tasks in the double-task method may cause cognitive overload and is quite intrusive, therefore, influencing the main task. Eyetracking and especially EVP and fMRI require highly technical lab settings which create a context and constraints that deviate to a large extent from everyday writing settings. Keystroke logging is less intrusive than most other methods, but fails to provide any direct information about the underlying cognitive processes that push along the writing process. Therefore, it is not surprising that a lot of researchers prefer to combine different research methods to complement specific shortcomings of one specific method. In the last section of this paper (*Multimethod Approach*) we will discuss the possibility of combining specific methods in more depth.

### **Keystroke logging in writing research**

One of the earliest attempts to use keystroke logging in writing research was described by Bridwel and Duin (1985). Their article is an early illustration of how the advent of the personal computer not only changed the writing context radically, but also how it created new opportunities as a research tool, viz. to observe and study writing. From then on several tools have been developed (for a recent review, see Latif, 2009) that all had in common that they facilitate a fine grained recording of every keystroke (and sometimes also mouse-movements and clicks). These logged data are then made available for further analysis and/or enable an exact replay of the emerging text. At the moment Scriptlog (B. Andersson & et al., 2006;

Strömqvist, Holmqvist, Johansson, Karlsson, & Wengelin, 2006), Translog (Jakobsen, 2006) and Inputlog (Mariëlle Leijten & Luuk Van Waes, 2006a; Luuk Van Waes & Leijten, 2006) are the most widely used keystroke logging programs.

Keystroke logging can be a suitable research instrument in a number of contexts (for an overview see Sullivan & Lindgren (2006)). Research areas include for instance: studies on cognitive writing processes in general, description of writing strategies in professional writing or creative writing, the writing development of children - with and without writing difficulties - , spelling, first and second language writing, the writing of expert and novice writers in professional contexts and in specialist skill areas such as translation and subtitling . The research technique can also be used in educational settings: second language learning, programming skills, and typing skills. For example, Lindgren (2003) used keystroke logging to facilitate reflection on writing activities through peer-based intervention by replaying the writing session.

The main rationale behind keystroke logging is that the fluency and the flow that characterizes the evolving text, reveals traces of the underlying cognitive processes.

Therefore, in the analysis of log files researchers mainly focus on characteristics of the pauses (length, number, distribution, location etc.) and revisions (number, type, operation, embeddedness, location etc.). As in speech, pauses are seen as indexical of cognitive effort.

Therefore, researchers interpret longer pauses as a possible indication that the related process or writing activity requires more cognitive effort. Several studies (c.f. Spelman Miller, 2000) have shown, for instance, that pause length increases according to the text level (Degenhardt, 2006) at which they occur. In general pauses between letters within a word are shorter than those preceding a word; pauses between sentences are shorter than those between paragraphs.

Revisions on the other hand could be considered as critical incidents indicating a conflict on a certain level between the writers' intention and the text produced so far (M. Leijten, Ransdell,

& van Waes, 2010). They can relate either to formal or content related problems that are observed during text production (Faigley & Witte, 1981).

This article presents some examples to illustrate the potential, strengths and weaknesses of the research method. In the next two sections we briefly describe and illustrate the general and technical characteristics of a specific keystroke logging program, Inputlog. Next we describe different research settings (workplace, educational setting) in which the instrument has to date been used to address a wide variety of research questions. The final section describes the possibilities for combining keystroke logging with other research techniques and tools such as thinking aloud or eyetracking. Readers who are also interested in the technical aspects of Inputlog should read Leijten (2007a) and the fact sheets on the *WritingPro: Knowledge Center for Writing Process Research* website ([www.writingpro.eu](http://www.writingpro.eu)).

## Features of Inputlog

Inputlog<sup>1</sup> has been developed at the University of Antwerp since 2003, initially as an alternative for Macintosh based logging tools and then as a means for observing speech recognition processes. Earlier, screenvideo-based research on speech recognition users' error correction strategies [17] allowed the analysis of specific speech recognition writing processes, but the collected data lacked precision and analysis was time-consuming. Additionally, most logging-tools were either developed for a specifically designed computing environment, or not adequately adapted to the current Windows environment. As such, they could not be used for research on 'natural' writing and computer networks using commercial word processors (e.g. Microsoft Word<sup>TM</sup>). These discrepancies propelled the development of Inputlog in 2003. In this article we describe version 4.1 of Inputlog.

For the development of Inputlog we were able to fall back on the functionality of two existing programs: JEdit and Trace-it (Kollberg, 1998; Severinson Eklundh, 1994; Severinson

Eklundh & Kollberg, 1992, 1996a, 2003; Spelman Miller & Sullivan, 2006), and ScriptLog (Strömqvist, et al., 2006; Strömqvist & Karlsson, 2001). JEdit and Trace-it are designed for Macintosh computers. JEdit only logs data in an in-house developed, limited word processor. ScriptLog also mainly logs in a limited word processor that was developed for research purposes (i.c. mainly writing experiments with young children). The most important feature that Inputlog builds on is the extended interactive revision module of Trace-it. Collaboration with the Scriptlog developers convinced us of the importance of a flexible output format and of combining logging data with other output files, e.g. from eye-trackers or speech recognition systems.

In short, Inputlog is a logging tool that allows researchers to:

- **Record** (keyboard, mouse and speech) data of a writing session in Microsoft Word, I-pad and other Windows based programs.
- **Generate** data files for statistical, text, pause, mode and revision analyses.
- **Integrate** various types of data from other programs (Morae, Dragon Naturally Speaking, Eyelink II).
- **Playback** the recorded session at various levels (text process, revision data, linear representation, graphic representation).

First, we describe the features of Inputlog that distinguish Inputlog from the other available tools (i.c. Jedit/Trace-it, Scriptlog, ULog and Translog: for an overview see (Luuk Van Waes, Leijten, Wengelin, & Lindgren, 2011). The most distinguishing characteristics of Inputlog to date are its *word processor independent* functionality, the *parsing technology*, the *standard XML structure* of the output and the logging of *speech recognition*.

## **Word processor independency**

Inputlog is primarily designed to log and analyze writing data produced in Microsoft Word. However, the program also logs keyboard and mouse actions in other Windows programs (e.g. Internet Explorer, Google Chrome, MS Excel, MS PowerPoint etc.). In other words, Inputlog not only logs the text development in the word processor document itself, but also related processes like consulting websites via a browser. Furthermore, all the windows that the writers opens, are identified and logged as such (e.g. different Word documents that are opened; consulted webpage URLs). So, if a writer googles specific information when writing a report, Inputlog logs the identification of the web browser used (e.g. Google Chrome), the active URL (e.g. www.google.com), the keywords used to activate the search operation, and the resulting web page accessed subsequently (together with an epoch timestamp in ms). This enables researchers to take writers' search behavior into account when studying the writing process.

## **Parsing**

In the context of the Inputlog program, parsing refers to the process of analyzing an input sequence. Inputlog logs the input of writing sessions and generates logging files that are used as input data for further analysis. After a writing session has been recorded, Inputlog generates different analyses from the source logging file. The processing of input data depends on what has preceded, and on what follows. Keeping track of these dependencies in order to verify how to process the data is called parsing. The parsing techniques simplify the overall program by separating the input and processing components and by providing a natural, modular structure. Furthermore, by hiding the implementation details of the different analyses, a more readable program structure is generated as well as faster processing times.<sup>2.3</sup>

XML structure of output files

Inputlog generates output files in a universal XML structure (Figure 1). XML is the abbreviation of Extensible Markup Language, which allows users to define the tags (markup) that are needed to identify the data and text in XML documents.

Figure 1. Example of XML notation used in Inputlog 4.1.

The example shown in Figure 1 illustrates the XML structure of Inputlog. Each `<tag>` states the information between the tags: e.g. `<WritingMode>2</WritingMode>` is a mouse movement or click, i.e. a click on the left button of the mouse; whereas `<X>252</X>` states the location of the X-axis of the mouse click on the screen.

The open and flexible XML notation structure allows researchers to adapt the data to their needs. Research data derived from other programs can be (re)structured in a comparable way as the Inputlog output and can then be integrated easily into one another. At present, we are developing a standard for keystroke logging tools to guarantee easier data exchange with other (keystroke) logging programs<sup>2</sup>.

### **Speech recognition**

As mentioned above, Inputlog is currently the only logging tool that can integrate the input of speech recognition software by Dragon Naturally Speaking (DNS, Nuance). The specially designed logging add-on in the DNS 8.1 (in combination with a Python script) enabled us to integrate text dictated with the Nuance speech recognition program with keyboard and mouse data logged by Inputlog<sup>3</sup>. As with the general logging file generated by Inputlog, Naturally Speaking logging files rely on timestamps. Via a Python script that was developed for this purpose, we generated an XML file of a recorded speech session structured as an Inputlog data file. The merged result is a single file that can be used for further analysis of multimodal writing sessions in which speech input is combined with keyboard and mouse.

Figure 2 shows a short XML excerpt of a logged writing process in Microsoft Word in which three writing modes have been used (column 1: writing mode): keyboard (1), mouse (2), and speech (3 – dict).

Figure 2. General logging file with data of Dragon Naturally Speaking.

In the excerpt shown in Figure 2 the writer dictates the following text segments (translated from Dutch):

```
04.39 <in context>
04.43 <The general obligations of the employer related to the protection of the employees health at the working place.>
04.55 <full stop>
05.02 <fit in with the development of the prevention policy><full stop>
```

The words that are dictated with the DNS software are represented as segments. Each continuous dictation episode is considered a segment. For each segment a time stamp for the start and end times are given in a dual coding: in hours:minutes,seconds and in milliseconds. The keyboard strokes, mouse movements and mouse clicks are represented per row as their actual output or as a readable code for each action (e.g. BS refers to backspace). Additionally, the timestamps are shown for the start and end of each action: for example key press in and key press out, beginning and end of mouse movement and start and end of a dictated text segment. For mouse movements and clicks the x-and-y-values are represented.

These multimodal logging data enable us to study the hybrid character of this kind of ‘writing’ in which dictated segments alternate with keyboard based word processing. Inputlog thus analyzes mode switches between speech and keyboard, error correction in various writing modes or rates of productivity in both speech and keyboard writing (cf. section on Inputlog & Speech recognition for an elaborated example in the context of live subtitling). The implementation of speech recognition in Inputlog could stimulate research on the effect of this new technology on the writing process of writers who use the software to

speed up their text production in a professional setting (e.g. respeakers producing live subtitling, lawyers writing an appeal), and writers with learning disabilities. Moreover, we would also like to explore the logging possibilities of speech recognition to simultaneously transcribe thinking-aloud protocols and/or retrospective interviews.

## General description of Inputlog

In this section we briefly explain the basic functionality of the program and its interface. The interface of Inputlog consists of an entry screen and 4 different tab pages: record, generate, integrate and play. Inputlog also provides a help file. This is a standard html-file that can also be consulted without starting Inputlog itself. For more detailed information about the use of the program we refer to this file.

### Record a writing session

By selecting the record tab, users can start a new logging session in two main types of logging environments, viz. *Microsoft Word*, or *without any predefined program (none)*.

*Microsoft Word* is the most commonly used word processor at the moment. This facilitates research in which the natural working environment is important. When starting a logging session, it is important to decide whether post-hoc revision analyses are an essential component of the research objectives. This choice is related to the general logging options: the basic version opens the normal Word version installed on a computer, whereas the extended versions – with several options ranging from Light up to Plus (cf. infra) - opens a Word-based template with extra facilities that allow post-hoc revision analyses of the logged data. The *basic* logging mode enables logging basic data in Microsoft Word or any other Windows application. However, data logged in the ‘basic’ mode cannot be analyzed in the revision module. Alternatively, researchers can use the *Light*, *Minimal*, *Plus* and *Full*

versions. In these versions the same basic data are logged (time stamps, environment switches, keystrokes and mouse operations) together with some extra input characteristics, e.g. position of the typed character within the evolving text, actual text length (in characters) after each input event, length and content of each selected text block, changes in text formatting like bold and italic. Using these additional data characteristics, it is possible to conduct a revision analysis (cf. Inputlog's manual for a detailed overview of which events are logged in each version).

At the end of each logging session an Inputlog Data File (idf) file is stored containing all the logged data in a compressed format. When the extended Word options are activated, an extra logging file (with an iwf-extension) is stored in the same directory. These source files are used as an input to generate different logging analyses.

## Generating analyses

In this part of the program, analytical files are generated on the basis of a source file recorded during a logging session. In other words, any idf file can be opened at any time to generate data output files for specific analyses. The output files are all XML-based files. These can be converted to Excel files or exported to SPSS for further analyses (cf. Figure 1 and 2).

At the moment Inputlog offers five different data analyses:

- 1. General file:** An XML file containing basic logging representation of the writing session in which every line represents an input action (keyboard, mouse click or movement and - if present – speech, window information); for every input action the session information is stored together with an identification of the start and end time of the input (key in and key out), the preceding pause time, and – for a mouse operation - the xy-value of the screen position (see Figure 2) .

**2) Linear text:** A plain linear text in XML-format containing the complete linear production of the text (keyboard and speech) including mouse movements and pauses. The linear analysis is divided into two options: a linear output in which the writing activities are divided into periods (fixed time durations of x seconds, free to choose) or intervals (fixed number of equal timeslots in which the writing process is to be divided). In both options the threshold for the pause length can be adapted to meet the requirements of a particular study.

Example:

Figure 3. Example of linear file.

**3) Summary data:** An XML file containing basic statistical information that identifies a particular writing session on a more aggregated level. Several process characteristics are shown, such as the number of characters, words, sentences, and paragraphs produced, the average length (and standard deviation) of words, sentences and paragraphs, average pause times (based on the threshold entered in the interface) and the use of the different writing modes. It is important to note that these data are based on the online process data: e.g. a paragraph that is for example written and deleted is counted in the process data, but of course not in the final product (e.g., via word count in MS Word). This enables researchers to calculate product-process ratios.

**4) Pause analysis:** An XML file containing analyses of every non-scribal period. The threshold for the pauses can be set to 1, 2 or 5 seconds or to any user defined level larger than 1 millisecond. Pause data are generated on a more general level: number of pauses, mean and standard deviation of pause length, and on a more specific interval level in which the writing session is divided into 10 equal timeslots. Finally, pause characteristics are

summarized at each text level, viz. word, sentence and paragraph boundaries. (Again this is based on process measures).

**5) Revision analysis:** An XML file containing a basic analysis of e.g. the number, the level and the kind of revision that has taken place during the writing session. Also the level of embeddedness of each revision is characterized.

The functionalities of the first four analyses have been described elsewhere (Mariëlle Leijten, 2007b; Mariëlle Leijten & L. Van Waes, 2006; L. Van Waes, Leijten, & Van Weijen, 2009).

The revision module is still under development and needs a more elaborate explanation in this article. Revision data can be analyzed via two methods: the replay module (cf. Section 3.4) and the revision analysis.

To define revisions we have developed an algorithm and a set of rules. The revision analysis first of all defines critical process events in the writing process that can be linked to revision and then evaluates these instances by comparing the operations in the isolated writing episode to the revision rules in the algorithm. Inputlog successively analyses the beginning of the revision, the selection of the text involved in the revision, or the positioning of the cursor, the (possible) deletion of the text and the end of the revision. In Fragment [1] we describe two (technical) revision operations to change the last word of the sentence ‘Questions of science, science and progress.’ into ‘evolution’.

Questions of science, science and [progress.]<sup>1</sup><sub>1</sub> {evolution.}

[1]

The first operation is a very basic one: the writer simply deletes the last word - i.c. by backspacing the full stop and the word 'progress' at the point of utterance - and then types the new word 'evolution'. This is a rather minimal operation, because the writer does not have to move or position the cursor in the text produced so far. However, the writer could also opt for another sequence to realize this substitution: for instance, he can move the mouse to the left, position the cursor by left-clicking the mouse, use the delete key to delete the word, change the text and move to the point of utterance by using arrow keys to the right. In the revision analysis these revisions will be identified identically because the representation represents the same basic text operation; the linear and the general file however, reveal the differences of the revision actions in detail.

At the moment the revision analysis of Inputlog calculates: number of revisions, type of revisions, level of revisions, time stamps related to beginning and end of revision, number of characters before and after a revision operation, number of characters in the final text, as well as the location of the revisions in relation to the point of utterance.

### **Integrate various types of data from other programs**

Inputlog's third tab is *Integrate*. This module allows researchers to merge different XML output files from other logging and observation programs. At present, Inputlog's output can be integrated with output files generated by *Dragon Naturally Speaking*, *Morae* and *EyeLink 2/EyeWrite*. The integration results in a single XML-file or Excel-file that can be used for further analysis.

#### **Dragon Naturally Speaking 8.1**

As mentioned above Inputlog data can be combined with speech recognition output from Dragon Naturally Speaking (cf. Figure 2). On the record tab researchers can indicate which

user of Dragon Naturally Speaking needs to be selected and on the Integrate tab researchers can merge both logging files into one file. To guarantee a successful merger, both logging files count with the same time measure. More specifically, Inputlog and Dragon Naturally Speaking need to be based on an absolute epoch time, resulting in a new XML file that is used as starting point for each analysis. Therefore, when recording data on different computers, It is therefore important to carefully synchronize the time settings when recording data on different computers (e.g. via time.nist.gov).

### Morae

Inputlog can be seen as a research instrument that provides data on the micro-analytic level of writing processes. To combine these very detailed data with macro-analytic data provides new perspectives. For instance, Morae is a macro oriented observation tool developed by Techsmith ([www.techsmith.com](http://www.techsmith.com)) for usability research purposes. It also enables researchers to code processes at various levels while observing these processes (simultaneously) online. It is our intention to complement the Inputlog data with Morae logging data. This program, for instance, enables researchers to flag important segments of an observed writing sessions and code these segments in great detail. Just like Inputlog, Morae logs very detailed timestamps enabling the integration of additional data registered by Morae into the output of Inputlog. For the observation of writing processes this integration opens new perspectives for further analyses.

### Eyelink

Finally, Inputlog data can be automatically integrated with eye tracking data from the Eyelink 2 device. Eyelink is one of the head mounted eye tracking systems of SR Research. The eye tracker also logs very detailed timestamps that could form the basis for the integration of both

data sets. Of course, the amount of data will increase enormously when combining these two data sets. However, also in this situation the Inputlog data may provide more insight into the eye tracking data. For instance, pause times could be indicative of monitoring the text produced so far. However, there is no direct indication of what writers are exactly looking at. Eyetracking enables researchers to identify what writers are looking at during a writing task<sup>4</sup>. Eyetracking is ideally suited for research on reading and writing processes. In section 6.4 we describe the concept of ‘*reading-during-writing*’ based on the analyses of process data in more detail.

### **Playback the recorded session at different speeds.**

The final tab of the Inputlog interface is the play function. A recorded writing session can be replayed using, the IDF file as a source file. In the basic version of Inputlog researchers can replay the logging file as it was recorded. However, it should be noted that the screen settings should be exactly the same before and after logging. In the Light, Minimal, Plus and Full versions the text can be replayed in four main windows (1-4; see Figure 4) and a flexible toolbar (5):

#### **1. text process - top left window (1)**

The writing session can be replayed at different speeds. It can be played back exactly as it was recorded (in real time). Since this might take too long the recording can also be replayed at the default speed, during which each interkey interval is limited to 150ms.

Finally, users can also replay the session at a percentage of the real time speed or can set each pause to a fixed interval.

#### **2. revision data – top right window (2)**

The revision data is represented in an Excel-like matrix, showing the main characteristics of each revision (revision number, revision type, recursiveness, start time revision, end

time revision, nesting depth, pause time before revision, number of characters before revision, number of characters after revision). In addition to these fixed variables, researchers can also add new variables: e.g., type of error, linguistic category of error, grammatical error, etc. All variables – fixed and added – can be changed and saved. This creates great flexibility for researchers to analyze the data according to their research question.

Figure 4. Example of replay full writing process and revisions.

### 3. **linear representation** – *bottom left window (3)*

The linear representation of the text is based on the S-notation by Kollberg and Severinsson (Kollberg, 1998; Severinsson Eklundh & Kollberg, 1996b). The S-notation has been developed together with Trace-It (Severinsson Eklundh & Kollberg, 1992) and has subsequently been used to describe revision in on-line writing processes (Eva Lindgren & Sullivan, 2006; Severinsson Eklundh & Kollberg, 2003; Sullivan & Lindgren, 2006). An example of S-notation is presented in Fragment 1.

### 4. **graphic representation** – *bottom right window (4)*

The graphical representation is a visual representation of the number of characters that are produced and deleted at each moment during the writing process. The cursor position and the pauses longer than a predefined threshold value are also shown. The graphical representation is based on a combination of Perrin's progress analysis (Perrin, 2003) and Lindgren's interactive representation via a Geographical Information System (Eva Lindgren, 2002, 2007).

The x-axis represents the time (in seconds) while the y-axis indicates the number of characters that are produced *cq.* realized effectively in the text produced so far. The top red line indicates total character production including deleted characters at each point in time

(cursor end position); the bottom green line indicates the characters retained after deletions at each point in time. So, when the green line drops, some characters are deleted. The blue field shows the difference between the characters produced and the characters in the text. The dotted line on the x-axis shows all the points in time at which the writer pauses during text production and the large pauses are represented by yellow circles. The size of each yellow circle indicates the length of the pause.

The red lines on the x-axis represents the location of each revision.

Researchers can generate a graphical representation automatically, but it can also be generated manually based on the revision data. In Section 5.3 we describe a case study in which the graphic representation was generated manually to represent differences in the writing process of two writers, a novice and a professional writer.

The writing session can be replayed at different speeds. It can be played back exactly as it was recorded (in real time): this is an exact reproduction of the recording of the session. Another option is that users select a percentage of the real time speed. The final option 'pauses at a fixed value' enables researchers to assign a fixed value, for example 0.1 seconds, to every pause (non-scribal activity). This allows users to view a writing session without long interruptions or pauses.

## Technical aspects of Inputlog

In this section we describe Inputlog from a more technical perspective. Only those technical aspects that may contribute to better grasp the concepts used in Inputlog are described. For further details we refer to Leijten (Mariëlle Leijten, 2007b) (technical terminology, glossary, program structure) and [www.writingpro.eu](http://www.writingpro.eu) (requirements, installation procedure, fact sheets for analysis).

Because of the various types of output files that can be generated by the program, Inputlog is able to log and analyze writing processes from different perspectives. When logging writing processes, Inputlog captures input data at a level before they are converted to screen information:

- scan codes of keystrokes (e.g. scancode '12' refers to the letter 'e').
- mouse activities (clicks, movements, location).
- time stamps of all input 'events' or 'actions'.

These data are stored in so-called IDF-files and IWF-files that are converted to different output files afterwards, preparing the rough data for qualitative and quantitative analyses (cf. general description). Figure 5 visualizes the general process flow of the program.

Figure 5. Flow of Inputlog.

In step 1 the process data of the online writing session is logged (cf. supra). In step 2 the logging data are saved in the IDF-file. In this source file each 'event' or action of the writing process is stored separately as binary data. The source file data can be used as input for the analyses, or as input for the playback module. In step 3 the binary data of the source file are converted to text data. This conversion is necessary because of the syntax directed parsing technique that Inputlog uses (cf. supra). The logging data are parsed between step 3 and 4: the text data are parsed to a readable XML-format by combining and reducing data, for example mouse movements are reduced to a starting point and an end point. The reduced XML file is the starting point for analyzing the modus data, the pause data and the text data on a more aggregated level in step 5. In this step the data are parsed by performing more specific rules on the data, for example via a set of rules which locate every pause on a specific text level (e.g. pause within words).

The source file (IDF/IWF) is also used as input for the playback module . This enables researchers to replay the writing session exactly as it was registered, or speed up to the researcher's preference, for instance by reducing long pauses.

Inputlog is mainly programmed in Visual C++.NET and the interface is programmed in Visual Basic.NET. We opted for C++ as a programming language because of its object orientedness and processing speed. Visual Basic.NET is the standard for interface development. For the logging of the writing process Inputlog uses two Windows hooks: Journalrecord and Journalplayback. For the parsing of the analyses we used Flex (Paxson, 1995) and Bison (Donnelly & Stallman, 1995) tools. Merging with Dragon Naturally Speaking was conducted with a Python script (Python, 2007).

## **Research perspectives**

In this section we briefly discuss writing processes taken from experiments in which we have used Inputlog as a research tool to study writing processes related to different genres, in different writing settings, and with diverging research objectives. Four such studies are discussed.

### **Writing in the workplace**

As we know from the literature on bad news communication, the strategic considerations involved in making the bad news as acceptable as possible for the reader are crucial in the perception of the message (and may also determine the future interpersonal relation with the

reader) (Jansen & Janssen, 2010, 2011). Bad news letters need to be formulated in a strategic and emphatic manner in order to maintain the relation with the reader.

When conducting research in the workplace on issues like this, it is crucial that the writers are able to work in their natural environment and that the research conditions affect the writing *situation* as little as possible. Good examples of this kind of research are, for example, Perrin and Van Hout (Perrin, 2003; Van Hout, 2010; Van Hout & Macgilchrist, 2010). In their research, they incorporated logging tools into existing journalistic writing platforms in order to log the writing processes of journalists and editors. We deliberately talk about ‘situations’ because it might not only be the text editor that is of importance. In lawyer’s offices we have noticed that the flow of documents is also an important factor so as not to disturb the situation or intervene with routine writing practices (M. Leijten, Van Waes, & Janssen, 2010).

Another example to illustrate writing in the workplace, is a study in which professional writers were asked to write a (negative) response to a claim (De Smet, 2009). The claim was based on a realistic situation and was described in a form normally used by the Health insurance fund at which we conducted the study. We opted to analyze the writing process of expert writers who had at least five years of writing experience in a genre, i.c. bad news letters, because research shows that writers need a substantial period of time to become really fluent in writing a certain genre (Kellogg, 2008; Schilperoord, 1996). The writing processes of these expert writers were contrasted with those of university students who had never written bad news letters before. In the current description we will mainly focus on the expert writers.

In general the study affirms the expectation that the experts’ letters meet readers’ expectations better than those written by students. An analysis of their final products shows that – as described in the literature (see Jansen & Janssen for a review (Jansen & Janssen, 2011)) - they especially include more elements that may generate a positive perception of bad news letters (give reasons, use indirect structure for difficult news, personalize). Experts pay more

attention to providing reasons to the readers and formulate more alternatives. Therefore, the letters of the experts are also longer than the letters written by the students. Consequently, experts spend more time on writing their letters.

If we narrow the analyses down to the formulation of the bad news (also called the Face Threatening Act, (Brown & Levinson, 1987)), we see that expert writers formulate the bad news in a very script-based manner. For instance, they hardly pause during the formulation of this crucial sentence in which they formulate the bad news, and they hardly pause or revise while producing this sentence. In the example, the writer focusses on the possible temporary state of the negative response to the claim by adding *at present*.

Figure 6 Production of Face Threatening Act by expert writer.

However, quite a lot of revisions were made in the text preceding and following the FTA-sentence. For instance, one writer explicitly revised his description of the situation leading to the refusal by adding the first name of the receiver's son that was involved in the case (*Jan*).

The intended effect of this addition was articulated in the retrospective protocol. The writer commented on the prompted revision in the playback as follows: “[...] I mentioned the name of the son explicitly, because she could have more sons and it makes the letter more personalized like this.”

This example shows that process logging enables researchers to analyze (professional) writing processes in a naturalistic situation. The addition of other observation methods, like (retrospective) thinking aloud protocols, complements the acquired data.

## Writing in an educational context

Keystroke logging is also used to study writing in educational settings. Educational researchers collect writing process data in classrooms, and use them for subsequent instructional purposes, for instance to create lessons involving observational learning activities (Rijlaarsdam, et al., 2008).

An example of this kind of study is Lindgren (E. Lindgren, 2004). She used TraceIt and JEdit to record and replay a pupil's writing process (13 years old). The process was played back for the author him/herself but in the presence of a peer. The peer took on the role of the questioner who acted like a researcher doing a stimulated recall "why do you pause here?" or "what did you think when you revised this phrase?". The results indicate that the method was generally successful especially for low L1 ability writers. It also helped pupils to separate the act of writing and the act of reflection (Braaksma, Rijlaarsdam, & Janssen, 2007; Rijlaarsdam, et al., 2008). Braaksma et al. used keystroke logging (Inputlog) in one of their studies on writing hypertext in the classroom (Braaksma, et al., 2007). In their paper they propose that hypertext writing at school could have beneficial effects on the acquisition of content knowledge and the acquisition of writing skills compared to linear writing. Students (N=123) from Grades 8 and 9 performed two linearization tasks and two hierarchicalization tasks under think aloud conditions. The writing processes were logged to time and identify the writing sub processes. Results show that writing hypertexts stimulates the use of writing activities that are positively related to writing proficiency and text quality, viz. planning and analyzing activities.

To illustrate how writing research in educational settings can be supported by keystroke logging with Inputlog in more detail, we briefly present a study in which we evaluated a

module for writing Bad news letters in Calliope (Daphne Van Weijen, 2009). Calliope is the online writing centre of the University of Antwerp (Jacobs, Opdenacker, & Van Waes, 2005). The online environment contains different modules related to business communication in three languages: Dutch, English and French. Each module consists of four inter-linked components: (1) a general introduction, (2) a theoretical section, (3) a set of exercises, and (4) a case (see also L. Opdenacker & Van Waes, 2007; Liesbeth Opdenacker, Verheyen, & Van Waes, 2007). Each of these components is linked to the other three, which gives users the opportunity to access them in the order they feel is appropriate. This structure was chosen in order to make the different modules accessible and user-friendly for users with different types of writing profiles (cf. Luuk Van Waes & Schellens, 2003) or learning styles' (cf. Opdenacker & Van Waes, 2007), such as those developed by Kolb (1984): the accommodator, the diverger, the assimilator and the converger (cf. also Opdenacker & Van Waes, 2007, p. 258 – 259). The main aims of the module are to help students learn how to analyze the context of the bad news that has to be conveyed, choose a suitable structure and strategy for the letter, and write a personal letter which is perfectly suited to the context and the reader's needs. The students must go over the information in all the Calliope sections and write a bad news letter on the basis of a case within a four hour time limit. However, they are free to access the information in the different sections in any order.

The study was carried out to determine whether the bad news letter module is suitable for students with different learning profiles, and whether students with different learning styles make use of its potential flexibility. Twenty students each completed the module and wrote a bad news letter. Their work processes were recorded with Inputlog 4.1 and with Statcounter<sup>TM5</sup>. The results of the analyses show an effect of learning style, which suggests that writers with different learning styles tackle the module in different ways.

To identify the participants' different navigation paths and switches between the Calliope learning modules on the one hand, and the switches between the writing and the learning environment on the other hand, we used the 'window identification' that Inputlog provides. As described above, Inputlog also registers which Windows environment is active: either program identification, document identification or URL; see output column in Figure 7. These unique identifiers enabled us to reconstruct the path followed by the students in their learning process, the number of switches they made between the different Calliope sections and the Word document in which they produced their text, and the time spent in each of the sections/environments. Finally, we also calculated how much time each writer devoted to each section during the different phases of the writing process. This was done by dividing each writer's total time on task into three equal parts: beginning, middle, and end. For each part we calculated the percentage of time devoted to each section of the module and text production in Word. This enabled us to determine the characteristics of their writing process in detail, taking into account their interaction with the online learning environment.

Figure 7 shows a fragment from the general logging file Inputlog generated for one of the students. In the output column we see that the writer opens a page in the theory section of Calliope with the information about the structure of a bad news letter. She appears to read this page for about 25 seconds (cf. StartTime) and then moves on to the page which contains information about the Subdivisions in such a letter. After accessing this page for about 20 seconds she returned to the empty Word document to type the first word of the letter 'Betreft' [Subject]. However, she makes a typing error (transposition of two letters) which is corrected at the end of the word.

Figure 7. Fragment from the Inputlog general file from the Calliope study.

In general the study shows that on average Assimilators work longer on the case than writers with the other three learning styles, generally viewed the highest proportion of pages in Calliope, and also switched most, on average, between Calliope and Word. The results also indicate that, in the beginning, Assimilators (69.67%) and Divergers (59.35%) appear to spend most of their time in the Theory section, while Convergents split their time between the Theory section (41.21%) and the Case (43.58%). We also found an interaction effect between time and learning styles for the Case section ( $F(6, 45) = 2.78; p < .05; \eta^2 = .27$ ). Thus it appears that writers with different learning styles do interact with Calliope in different ways to a certain extent during the writing process. In other words, the module's users appear to make use of its potential flexibility instead of solely interacting with the module in a more traditional linear way.

In the next stage of this on-going research project we will relate these observations to more product related issues, e.g. by also assessing the quality of the final products. It goes without saying that this kind of logging application can also be useful in tracking the interaction with different (online) sources and application in the creation of texts. Nowadays, most writers when creating a text constantly rely on multiple online sources – either texts they have written themselves previously, or publications by other authors. In the meanwhile, especially in collaborative environments, they exchange ideas via mail or online (social) fora and integrate these in their texts. We envision the detailed process information that Inputlog provides to play an instrumental role in the study of digital writing processes.

### **Writing development**

The study of writing development is one of the most important domains in writing research. Bereiter and Scardamalia (Bereiter & Scardamalia, 1987) related writing development to writers' general cognitive development. Beginning writers tend to use a *knowledge-telling*

*writing strategy* and they plan and write their texts in a more ‘list wise’ manner. The primary focus is on ideas and on putting these ideas on paper. Text elaboration mostly concerns formal aspects, such as spelling. Later, writers develop the ability to evaluate whether the text expresses what they actually intended; they use *knowledge-transforming strategies* to revise their text so that it conveys the intended message. Knowledge-transforming strategies thus include formal meaning aspects of the text.

Kellogg (Kellogg, 2008) introduced yet another level of writing strategy, the *knowledge crafting strategy*. As writers’ experience increase, they are not only able to interact with the text, they also interact actively with the intended reader. Knowledge crafting is thus characterized by advanced writing skills which enable writers to evaluate, during writing, whether the text expresses what they intended and whether the contents and the form in which it is presented match the intended readership. Awareness of the reader during writing is considered a crucial aspect of successful writing and maturation of the working memory plays an important role in the development of reader awareness.

Against this background we set up a study (Eva Lindgren, Leijten, & Van Waes, 2008) in which various writers, representing different levels of expertise in writing, undertook the same writing tasks. In this article we present a case study showing an example of one task – rhz production of an instructive text – performed by a young girl, aged 10, and a professional writer, aged 34.

The writers were shown a video clip explaining how to plant and take care of bamboo, and were allowed to watch the video as many times as they wanted. The writing process data were collected using Inputlog 4.0 and were complemented with stimulated retrospective interviews. Analysis focused on the textual coherence strategies of both writers.

The instructive text was based on a video in which some structural elements were manipulated and the way in which the audience was addressed. The task was to transfer the

video instruction into a written text. This task enabled us to analyze how writers with different levels of writing experience deal with these audience related issues. The use of graphic representations showed to be very valuable in the interpretation of the data.

Figures 7 and 8 visualize how (1) the total text production as cumulative number of characters produced at different intervals, (2) the actual length of the document at every interval which gradually increases and sometimes decreases when text is deleted, (3) the cursor position as an indication of the (non-)linearity of the process, and finally, (4) the distribution of pauses as dots which relate to the right y-axes. The combination of these pieces of information enables us to describe the specific characteristics of each writing process and relate them to the writer's audience awareness.

Figure 8 shows that Judith's writing process is clearly linear. Her writing process lasts for about 39 minutes and her text gradually grows throughout that period, with only few revisions. During the first half hour Judith (10) only revises smaller typing errors. The distribution of the pauses (dots) indicates that she is cognitively struggling with the text. The pattern of the dots shows that she produces numerous pauses longer than five seconds during the entire writing session, probably reflecting a strong emphasis on internal text formulation and revision.

Figure 8. Graphical representation of Judith's writing process (instructional text).

In sum, Judith's writing process is characterized by limited interaction between planning and translating, with minimal reviewing. This reflects a knowledge telling strategy, with only limited transformation as compared with the source text.

In Figure 9 Dieter's writing process is shown (professional writer). A first look already indicates that the nature of the writing process is completely different as compared to Judith. The first seven minutes of Dieter's writing process are characterized by a fluent and rather

linear text production. Dieter sets off by creating a skeleton structure and distributes the information presented in the video into these content categories he himself calls ‘structural devices’. He distinguishes between two main – user centered - themes in the overall instruction: (1) how to plant bamboo, and (2) how to care for bamboo. When he has crafted his ‘text skeleton’, he elaborates the different parts of the text, while relistening to the video in the background. But, as the graph in Figure 9 shows, this is not done consecutively.

Figure 9. Graphical representation of Dieter’s writing process (instructional text).

He knows that revision of one part of the text can affect other parts of the text. After about nine minutes a first draft of the text is finished, which he then revises in four different rounds (9-15; 15-20; 20-the end; cf. dotted line representing cursor position). From the process data in combination with the retrospective interview, we learn that every round has its own focus, and the rounds gradually become shorter:

9-15	focus on the persuasiveness of the text
15-20	focus on instructive quality (e.g. numbering, parallel phrasing and separating speech acts)
20- end	focus on stylistic perspective (readability and attractiveness).

In short, Dieter spends about two thirds of his time ‘sculpting’ the content towards the reader’s needs. He frequently produces two versions of a sentence next to each other before he decides which of them is most appropriate. This also explains why his final text only contains half of the words that he produced during the writing session (cf. difference between the lines representing the total number of words produced and the actual document length).

These kinds of case studies illustrate how a graphical representation of keystroke logging data provides a clear picture of the emerging text process. In combination with retrospective interviews, the writing processes can be described in great detail and differences between writers and their strategies become apparent (see also Eva Lindgren, 2002).

### **Writing mechanics**

Inputlog data can also be used to analyze data at the micro level, for instance, interkey intervals in relation with typing speed, keyboard efficiency of touch typists and others, dyslexia and keyboard fluency, biometric verification on the basis of digraph latency etc. A nice example of such a study is Grabowski's research on the internal structure of students' keyboard skills in different writing tasks (Grabowski, 2008). He studied whether there are patterns of overall keyboard behavior and whether such patterns are stable across different (copying) tasks. Across tasks, typing speed turned out to be the most stable characteristic of a keyboard user. Another example is the work by Nottbusch and his colleagues. Focusing on linguistic aspects of interkey intervals, their research (Guido Nottbusch, 2010; G. Nottbusch, Weingarten, & Sahel, 2007; Sahel, Nottbusch, Grimm, & Weingarten, 2008; Weingarten, Nottbusch, & Will, 2004) shows that the syllable boundaries within words have an effect on the temporal keystroke succession. Syllable boundaries lead to increased interkey intervals.

In recent research we have used Inputlog data to analyze typing errors from a micro perspective (publication in preparation). As we all know, typing errors – and their correction – often interrupt the linear production flow of new text which sometimes lead to a loss of planning information stored in our short term memory. We briefly present this study here to illustrate how this kind of logging data can be used in micro analyses.

Inputlog defines every keystroke by two time stamps:

- Key-in time: time in ms when the key was pressed
- Key-out time: time in ms when the key was released

Technically, a pause time ( $\Delta$ ) can be described as the period of inactivity between two keystrokes ( $i$ ):  $i^{\wedge}i$ . More specifically, the pause time ( $P$ ) for a specific Key ( $i$ ) is defined as the time between two consecutive presses:

$$\text{Pause}_i = P_i - P_{i-1}$$

[2]

In other words, a pause is calculated as the latency between the pressing of the previous key (key-in time of key  $i-1$ ) and the actual key  $i$  (key-in time of key  $i$ ). This type of interkey interval is also sometimes referred to as *Flight time* (e.g. (Douhou & Magnus, 2009)).

Ex

Writing Mode	Output	Position	StartTime	StartClock	EndTime	EndClock	ActionTime	PauseTime	Pause Location
I	T	23	<a href="#">17500</a>	00:00:17	17547	00:00:17	47	578	4
I	h	24	<a href="#">17750</a>	00:00:17	17828	00:00:17	78	250	1
I	e	25	<a href="#">17907</a>	00:00:17	17985	00:00:17	78	157	1
I	SPACE	26	<a href="#">18032</a>	00:00:18	18094	00:00:18	62	125	2

am

ple

(ge

ner

al

file) of a typing sequence in which the word ‘The’ is typed:

or shown in a linear representation:  $\cdot\{578\}T\{250\}h\{157\}e\{125\}$ .

In this example the pause time between the letter ‘h’ and the ‘e’ is 157 ms, viz.  $17907 - 17750 = 157$ . The pause preceding the word is a bit longer (578 ms). Usually pause time increases as the text unit level increases (Spelman Miller, 2000; A. Wengelin, 2006).

The Action time (A) for a Key (*i*) is defined as the time between the key in time and the key out time of that specific key. It is the time the key is actually pressed:

$$Action_i = P_i - D_i$$

[3]

In other words, a pause is calculated as the latency between the pressing of a key (key-in time of key *i*) and the release of the same key *I* (key-out time of key *i*). This type of interkey interval is also sometimes referred to as *Dwell time*. In the example above the Acton time varies between 47 ms and 78 ms.

For the analysis of typing errors we analyzed the pausing times on the digraph level. In an experimental setting 30 participants were asked to type several (combinations of) words 25

times. Five different digraphs with different characteristics (frequency, keyboard distribution, left-right coordination) were selected to analyze inter and intra typist differences in latency distribution and the effect of typing errors on the length of the interkey interval. The results of a multilevel analysis on the data show that there is no correlation between the frequency of a digraph and the chance that a typing error occurs. Typing errors show a limited variation: pressing the adjacent key explains more than 40 % of the errors, both for touch typists and others; the chance that a typing error is made is related to the characteristics of the digraph, and the individual typing style. Finally, the median pausing time preceding a typing error tends to be longer than the median interkey transitions of the intended digraph typed correctly. We hope that further research on typing errors will lead to a better insight in this phenomenon and that it will also enable us to add an automatized routine to the program so that we can distinguish typing errors from other revisions in process data. This kind of filter would enable researchers to analyze revisions without the 'noise' of typing errors.

### **Applications on a broader level**

Inputlog can also be used as a research instrument to facilitate research methods on a broader level. For instance, Inputlog can be integrated as an additional research instrument to register complimentary research methods such as thinking aloud and retrospective interviews. In the research project described in section 6.2, for instance, we recorded the retrospective interviews of the participants via Inputlog and Dragon Naturally Speaking. As a result, the retrospective interview was automatically transcribed and timed, facilitating the tedious transcription work considerably.

This kind of extended use of the logging software certainly opens up avenues for further research on thinking aloud and retrospective interviews as research methods. In addition to

pauses in the writing process, pauses in a thinking aloud protocol can be an interesting measure for gaining insight in the cognitive load during the production of protocols.

## **Multimethod approach: Inputlog in relation with other logging tools**

All observation methods used in writing research have their strengths and weaknesses. One of the advantages of keystroke logging is its complementarity with other research techniques and tools. In this section we discuss four different studies in which keystroke logging was combined with another observation method: (1) thinking aloud protocols, (2) speech recognition logging, (3) usability logs, and (4) eyetracking recordings.

### **Inputlog & thinking aloud protocols**

To collect cognitive data during the writing process, many researchers have used 'thinking aloud protocols' (TAPs). Starting from the early work of Matsuhashi and Hayes (L. S. Flower & J. R. Hayes, 1980; Matsuhashi, 1982), TAPs have been used to address a variety of research objectives in writing research and related domains (e.g. influence of new media on writing (M. Leijten, L. Van Waes, et al., 2010); comparison of L1 and L2 writing (D Van Weijen, 2008); etc.). TAPs were developed in the field of cognitive psychology by Ericsson and Simon (1984, 1993). By having writers verbalize their thoughts when writing, the researcher collects cognitive data related to (conscious, non-automated) problem-solving or decision-making strategies (Krings, 2001; Smagorinsky, 1989). Because keystroke logging is an unobtrusive research instrument, it is possible to combine logging with TAPs. Both research methods complement each other in several ways. Of course, one has to take into account that having writers verbalize their thoughts simultaneously, may disturb the problem-

solving cognitive process and the fluency of writing as such (see (Dam-Jensen & Heine, 2009; Göpferich, 2008) for an overview of the validity and reliability issues related to TAP). The main points of criticism leveled at TAPs concern the interference with the ‘normal’ writing activities, especially related to the low level cognitive activities. Thinking aloud while writing takes up more time than writing ‘silently’. This claim has been empirically validated by Krings (Krings, 2001), Jakobsen (Jakobsen, 2003) and Janssen, Van Waes & Van den Bergh (Janssen, et al., 1996), who report average delays between 25% and 33%. The reliability of TAPs has also been questioned since it might be possible that inconsistencies occur between the mental processes involved and the verbalizations uttered. Therefore, we think that by combining simultaneous TAPs with KSL researchers create a more solid basis to infer cognitive processes from the collected data.

A study by Schrijver, Van Vaerenbergh and Van Waes (Schrijver, Van Vaerenbergh, & Van Waes, 2011) on ‘transediting strategies’ in translation processes is a nice illustration of how TAPs and KSL data may complement each other and may compensate some of the shortcomings mentioned above. To study transediting, the participants were given a translation brief. The source text was a patient information leaflet about a medicine not yet commercialized in the European Union, i.e. Geodon®. The target text had to be in accordance with European legislation and guidelines (terminology, norms, standards, structure, and lay-out). In this study the participants were prompted to produce a simultaneous TAP, and their writing activities were logged with Inputlog.

Table 2 shows a 15 second excerpt taken from the transcribed data illustrating a low-level process. In the linear keystroke logging representation we see that after about 10 seconds, the writer pauses for about 3 seconds before writing the verb ‘gebruiken’ [use]. A pause that is hard to interpret. However, in the TAP we see that during that pause the translator hesitates about the word choice (‘to use’ vs. ‘to consume’).

Table 2. Example of a combined TAP and a linear KSL representation

In Table 3 we present an example of a more high level process. The production of the first item in a bulleted list (“when you suffer from a psychosis”) is interrupted by a long pause of about 1 minute between the first word (“when”) and the rest of the sentence (see linear representation: •wanneer•{56906ms}). The TAP transcript reveals the strategic considerations during this interruption of the sentence production. The start of the bulleted list triggers the writer to represent certain items in the source text differently in the target text. She also shortly explains the reason for this reorganization, and then continues writing. Again, the TAP and the keystroke logging data complement each other. On the one hand, the keystroke logging transcript shows an exact representation of the writing process, including the exact pause length and little automated actions like the correction of typing errors. On the other hand, the TAP reveals the contents of some cognitive processes that lead to certain interruptions in the fluent text production (longer pauses).

Table 3. Example of a combined TAP and a linear keystroke logging representation

### **Inputlog & Speech recognition**

Speech technology has made it possible to use speech recognition for text production. In the late nineties speech recognition systems were first made commercially available and allowed speech to text conversion on a normal PC in a fairly reliable way, i.e. with an accuracy rate above 95% (Honeycutt, 2003). At the moment, speech recognition is mainly used both in a business context (lawyers, doctors, translators) as well as in an educational context to help young children with certain learning disabilities to write more fluently (Mariëlle Leijten &

Luuk Van Waes, 2006b; MacArthur, 2006; Quinlan, 2006). Speech recognition is also used in live subtitling, for instance by the BBC and the Flemish public television (VRT). The VRT launched speech recognition as their standard method in of (real-time) live subtitling in 2001 (Remael, 2007).

To illustrate how Inputlog enables reserarchers to log writing processes in which different text input media are combined, we would like to show an excerpt from a study in which professional subtitlers use speech technology (Dragon Naturally Speaking, Nuance) together with their keyboard and mouse to produce subtitles for programs that are broadcasted live (Luyckx, Delbeke, Van Waes, Remael, & Leijten, 2010). In this context, respeakers have to listen and speak concurrently, within the constraints of time, place, and cognition. This makes live subtitling a particularly challenging and cognitive demanding writing task. With these issues in mind, live subtitling data were collected and a detailed analysis was conducted based on Inputlog data, combing keyboard&mouse data and the data of Dragon Naturally Speaking and Softel Swift (subtitling tool). As already shown in Figure 2, both the subtitling and logging file can be combined and integrated into one general logging file, which is based on the convergence of timestamps (Mariëlle Leijten, 2007b).

Table 4. Example of a merged general log file (keyboard, mouse, speech)

Table 4 shows an excerpt taken from a live subtitling session logged for this research project. In the first part of this excerpt, the subtitle is “*He belonged to groups who did not want to join the VNV during the war.*” is produced. It takes about 16 seconds to broadcast this subtitle. The long delay<sup>7</sup> is mainly caused by the speech recognition error at the end of the first dictated phrase. The respeaker voiced the Dutch abbreviation “VNV” (Flemish National Union) as [ve: en ve:] and the speech software recognized this as “vee en vee” [“cattle and

cattle”] which resembles the phonetic input quite closely, but which does not make any sense semantically. The respeaker notices this error and decides to reposition the cursor (by mouse and keyboard, column 1). He deletes the last occurrence of the word “vee” and replaces it by “VNV”, and then deletes “vee en” by selecting the words with a shortcut (SHIFT + CTRL + LEFT) followed by the BackSpace-key. The subtitle is now completed with a short dictated clause (“during the war”) and broadcasted (using the PageDown-key as a shortcut).

This detailed representation of the merged logging data underlines the interaction between the different input media. The analysis is an illustration of the fact that the study of the production process of subtitles in this context helps to understand the possible causes for delay.

### **Inputlog & Morae**

One of the challenges in writing research in general is to explain the structural variation in writing processes within and between subjects. In general, recursivity is attributed to writing experience, proficiency, task characteristics and the writing mode or medium. This study focuses on professional writers (n=10) who use a modern writing instrument - speech recognition - as their primary tool for text production and revision. More specifically we are interested in the way this new technology affects the cognitive processes that underlie text production. In our study we focused on error correction, providing a description of the errors that professional speech recognition users need to deal with, how they deal with them and why they opt for various error correction strategies. Different converging research methods were used: (1) product, (2) process, and (3) protocol analysis.

The results show that the contrast between immediate and delayed error correction is quite decisive for the way in which writers structure their writing process. Next to this, the distinction between technical problems and revisions also plays an important role. Most writers prefer solving technical problems immediately. The same does not necessarily hold for

other – more stylistic or content related - revisions. However, the revision behavior is not random: overall results show distinct patterns or profiles of error correction.

To register the process, we opted for a combination of Inputlog, Dragon Naturally Speaking and Morae. As mentioned above, Inputlog is specifically designed for micro-analytic research on writing processes. However, these very detailed data can easily be combined with more macro-analytic research tools. Therefore, we have complemented the data of Inputlog with another observation tool: Morae (version 1.3)<sup>8</sup>. This program is mainly developed for usability testing and uses an online screen cam (Morae Recorder) to register every action on the computer screen. Next to some lower level analyses, Morae also captures changes between programs on a higher level and allows extra online coding. Just like Inputlog it also logs very detailed timestamps which enabled us again to merge the additional data registered by Morae with the Inputlog output.

In addition, Morae integrates a webcam that frames the writers' actions in combination with the text developing on the screen. In this experiment we were mainly interested in the way writers interact with the 'text produced so far' on the computer screen. Figure 10 illustrates the observational setting of the experiment.

Figure 10. Observation setting of experiment.

As Figure 10 shows, the Remote Viewer of Morae offers the possibility to observe the writing process from a different location. Observers can connect to the experimental computer, view the writers' computer screen and view the writers' face, as well as hearing what the writer is saying. Using the correct settings the observing researcher can hear the input of the speech recognition. Finally, the remote viewer enables us to 'flag' important moments of the writing session that can be used as a basis for a retrospective interview.

Figure 11. Example of merged logging data of Inputlog and Dragon Naturally Speaking.

Figure 11 shows an example of a merged output based on the output files of Inputlog, Morae and Dragon Naturally Speaking. As seen in Figure 2 Dragon Naturally Speaking data are shown per dictated segment (see previous section for more details) and the Inputlog data per event.

In this example four extra columns are added (cf. Figure 11). These extra columns are the clock time of Morae, the merge time (in milliseconds) between Morae and Inputlog, Morae's marker identification and the corresponding code used in this study.

In this excerpt Morae's marker refers to a technical error caused by the speech recognition software. The code 'O' is chosen by the researcher and stands for 'technical error, immediately solved'. The participant initially dictated the following two segments as one segment (4:43) 'The general obligations of the employer related to the protection of the employees' health at the working place' and (5:02) 'fit in with the development of the prevention policy'. In the pause related to the full stop (3.1 seconds) the participant is reading the text produced so far and notices that the final segment does not appear on the screen. After a pause of 5.6 seconds he switches into writing mode to delete the full stop with his keyboard and insert an interspace. He proceeds with speech and dictates the final segment once again. Interestingly, he does not correct the other smaller errors in the text (title of section), for example 'context' instead of 'context in'. A subsequent segment of the observation reveals that he prefers to continue with text production and to finish his paragraph.

This example shows the interaction between the macro-analytic approach of extra process coding in Morae and the micro-analytic approach of Inputlog. It enables researchers, for instance, to search through the information in a more structured way and to combine more global analyses with very detailed analyses.

## Inputlog & Eyetracking

The logdata of Inputlog can also be merged with logdata of Eyetracking devices. This enables researchers to characterize reading activities during writing processes. Using eye tracking data in reading research has a relatively long tradition (Rayner, 1975). In writing research, however, only a few projects have been conducted in this field (B. Andersson, et al., 2006; R Johansson, Johansson, Wengelin, & Holmqvist, 2009; Luuk Van Waes, Leijten, & Quinlan, 2010; A. Wengelin, et al., 2009b) . Until a few years ago, no adequate research tools were available to study the process of reading during writing. Since the emergence of keystroke logging tools, the possibility of combining both keystroke logging and eyetracking data, became more feasible and created a better basis to interpret the fixations in fluidly produced written text which is constantly changing. In this section we show some of the possibilities *reading during writing* data provide for writing process research.

The simultaneous logging of eyetracking data and keyboard-and-mouse events, enables us to analyze the interaction between both activities. The main objective is to get a better insight into the function of – different types of - reading that feed and support the distinct subprocesses of writing. The most important obstacle in these kinds of research questions is the concept of 'emerging text'. We should take into account that in eyetracking reading research small units of static text have been the main focus of research: "Eye tracking has been successfully adopted to the study of basic reading processes and to that of syntactic parsing, but there are surprisingly few studies where eye tracking is employed to examine global text processing. A likely reason for this is might be the apparent lack of consensus on the measures to be used to tap into global text processing ((Hyönä & Lorch, 2004, p. 313))."

In writing process research the most suitable measures for describing reading processes during writing in detail are still being developed. So far, research has been limited to what writers read during writing and at what moment in the writing process writers look at the screen. In this section we would like to initiate more fine-grained analyses of the various activities of reading during writing by describing the reading activities of two kinds of writers: monitor gazers and hunt-and-pecktypists.

The data are based on a short typing test (Leijten, Galbraith, Torrance & Van Waes, conducted at Staffordshire University in 2008). The writers were typing a nursery rhyme by heart (e.g. Twinkle twinkle little star...), so as to lower the cognitive load caused by content generation as much as possible.

In this case description we mainly focus on the reading activities that occur during critical discourse events, i.e. when fluent text production is interrupted either by a *revision* or a *pause*. In Figure 12 and 13 we show a visual representation of both critical discourse events, viz. a revision and a pause, for the two distinct writing profiles.

Figure 12. A revision as a critical discourse event during writing.

The timeline in Figure 12 represents the synchronicity of reading and writing related to two keystroke events. The top line shows the keystrokes with a short pause in between (on average between 50 and 250ms), indicating a – more or less - fluent typing episode. The first key is identified as a last keystroke of a series of typing activities that relate to the production of new text; the second key on the timeline is identified as the first keystroke that initiates a *revision* activity (e.g. the correction of a typing error via the backspace key). We hypothesize that the characterization of the fixations that precede this critical discourse event deviates

from the fixations that relate to fluent text production. Fixations might be related, for instance, to identifying the typing error in the text or evaluating a phrase in the text produced so far that might need revision.

Figure 13. A pause as a critical discourse event in writing.

The second timeline in Figure 13 represents the synchronicity of reading and writing related to two keystroke events that are interrupted by a significant pause (e.g. significantly deviating from the normal interkey interval in fluent writing). There is ample support in writing research that longer pauses which interrupt fluent text production indicate a cognitively complex situation for the writer (A. Wengelin, 2006). During these longer pauses usually certain ‘reading’ activities take place that can be identified by a series of fixations and saccades. These are instances where the writer interacts with the text produced so far in various ways. As Kaufer, Hayes, & Flower (1986) observed, writers visually interact with their own text for various purposes, such as ‘finding prompts to allow search and retrieval of more ideas’ and ‘evaluating the formal correctness or suitability of the text produced so far’ (Luuk Van Waes, et al., 2010).

Figure 14 shows a small excerpt of a hunt-and-peck typist. This is a very straightforward example of ‘reading during writing’. The Figure has been developed by merging the Eyetracking and Inputlog data.

Figure 14 Reading during writing behavior of hunt-and-peck typist.

The writer is typing the nursery rhyme ‘Twinkle twinkle little star’ by heart. When typing “Twinkle twinkle” (see square dots on x-axis) he apparently ‘feels’ - proprioceptive awareness - that he makes a typing mistake in the second word. He monitors the text on the

screen to identify the error in the text produced so far on the monitor (see rereading diamonds on top line indicating eye fixations in the TPSF). After correcting the typing error, he continues typing and produces seven words without looking at the screen, not even while typing. Then he rereads this part of the text, and continues. This example is typical for a hunt-and-peck typist who produces new chunks of text in quite large ‘bursts’ without significant pausing and without monitoring the text produced so far on the screen (Chenoweth & Hayes, 2001; M Leijten, Van Waes, & Ransdell, 2010).

In Figure 15 a short selection of the writing process of a monitor gazer is shown in which the concept of rereading needs to be subdivided in (at least) two categories.

Figure 15. Reading during writing behavior of monitor-gazer.

This writer is constantly monitoring the text produced so far on the screen while typing, more or less following the moving cursor. In this figure, the category ‘reading’ stands for reading during fluent typing. Fluent typing is calculated as a sequence of keystrokes that is produced with an interkey interval of less than 300 ms. The Figure shows a very rhythmic pattern of text production and monitoring of the screen. The first two errors ‘and w’ and ‘may’ are fixated almost immediately and also corrected immediately (see black squares on top line indicating rereading activities that indicate ‘awareness’ of the error in the TPSF). The final error – displayed in a black and white square on the top line – has also been corrected immediately by the writer, but she hasn’t explicitly fixated the error with her eyes. The writer probably must have felt – kind of proprioceptive awareness – the error and corrected it

without the need to really fixate the error. Apparently, the writer can focus on the text production and does not get distracted by correcting this typing error as such.

These examples show that merging keystroke logging data and eyetracking data shed a new light on the reading-writing interaction that characterizes writing processes. By integrating both types of data new perspectives are created to analyze writing data from different perspectives and to infer underlying cognitive processes on a more grounded basis.

## **Further developments & future research perspectives**

In this article we have shown some possibilities of Inputlog for describing cognitive writing processes from a multimethod approach. Inputlog is a logging tool that can be used in different settings, either as a standalone application, or in combination with other observation methods and tools (e.g., thinking aloud protocols, eyetracking, speech recognition logging). The main advantage of observation with key stroke logging is that the method is unobtrusive and that it generates very fine-grained data for further analysis.

We want to wrap up this paper by elaborating briefly on further developments and perspectives. In the near future we will mainly focus on two tracks: (1) the standardization of the XML structure for keystroke logfiles; (2) aggregation of the process data from the letter level to the word level by merging them with lexicons and natural language processing tools.

### **Standardization**

Within the framework of the European Cost Action on ‘Learning to Write Effectively’ the ‘Technological advances in writing tools’ workgroup represents the most important

developers of keystroke logging tools. One of the spearheads of this group is the standardization of keystroke logging data. At present, each logging program uses its own formats and data analysis modes. To further facilitate the exchange of data and expertise between research groups, it is important to standardize the data structure of the output files and agree on a common XML-format. This technical and methodological harmonization aims at further fostering the development of the different tools and analytical procedures.

Furthermore, to stimulate the exchange of expertise, an on-line knowledge center for Writing Process Research called WritingPro was launched. At this stage in writing process research, we think it is very important to spread and discuss various types of knowledge and expertise on keystroke logging tools and the related techniques of data analysis. The main aim of the WritingPro website is to bring together the expertise of researchers involved in keystroke logging (both developers and users), share expertise, further develop expertise, make expertise more visible, and establish 'best practices' in writing research methods (see [www.writingpro.eu](http://www.writingpro.eu)).

### **Aggregation of logged data to the word level**

In a current research project we are exploring the possibilities of aggregating the logged process data from the letter level (keystroke) to the word level by merging them with existing lexica and NLP-tools. This may create a very valuable basis for more linguistically oriented writing process research, and could establish a new paradigm in writing process research.

In a first phase we will aggregate keystroke data to the word level by identifying and filtering the typing errors made during text production. This is a necessary step in making the linear

data ‘readable’ and linking them to existing lexica (like Celex or e-Lex). These lexica consist of coded information about aspects like part of speech, word class, lemma, syntactical and orthographic status, pronunciation, morphology etc. We opt to link both to lexica generated on the basis of written and of spoken language because of the process characteristics of written discourse, especially when produced with speech recognition. Since most word characteristics (like word class or lemma) are contextually determined, we intend to link the process data existing to Natural Language Processing (NLP) tools like the Tadpole tagger-lemmatizer-dependency parser (Van den Bosch, Busser, Daelemans, & Canisius, 2007) and decomposition software (Vandeghinste, 2008).

Linking writing process data to lexica and NLP tools enables us to analyze the data on a higher and more complex level and to relate this type of writing research to ongoing research in other domains (e.g. Pragmatics, CALL, translation studies). We hope that this perspective will further enforce and enrich writing process research.

#### NOTES

<sup>1</sup> To facilitate a broad usage of Inputlog, the program is put at the disposal of the research community free of charge ([www.inputlog.net](http://www.inputlog.net)), provided that reference is made to this article. User feedback is very important for the evaluation and further development of Inputlog. For any and all questions or feedback, please feel free to contact the authors.

<sup>2</sup> Standardization of the XML-structure is conducted within the COST ‘European Research Network on Learning to write effectively’ ([www.cost-lwe.eu](http://www.cost-lwe.eu)) . More information about this project can also be found on [www.writingpro.eu](http://www.writingpro.eu).

<sup>3</sup> Most users of Dragon Naturally Speaking combine spoken input with keyboard and mouse operations. Therefore, it is important to analyze writing processes a dictation context from a multimedial perspective (see Leijten, 2007).

<sup>4</sup> For this purpose, EyeWrite (Simpson & Torrance, 2007) was developed. This program uses the Eyelink II in combination with a tailored word processor. The custom made application not only generates the X-Y coordinates of each fixation on the computer screen, but also matches them with the actual word at that location at that specific moment in time (fluid text production).

<sup>5</sup> Accommodators prefer to learn-by-doing, and are thus likely to focus on the module's case or exercises and are less likely to focus on the module's theoretical information. Assimilators, on the other hand, prefer to focus on theory and structure first, before putting this new knowledge into practice by carrying out the case. Divergers generally prefer to look at a new task from several perspectives and gather plenty of information before carrying it out. Therefore, they are likely to spend a lot of time in Calliope gathering information before carrying out the writing task. Finally, Convergents prefer to solve problems and make decisions by actively searching for a solution, which suggests that they may focus on the case initially, before examining the theory section of the module (Kolb, 1984).

<sup>6</sup> Statcounter is an invisible web tracker to develop web stats ([www.statcounter.com](http://www.statcounter.com)).

<sup>7</sup> On average the delay between the spoken comment and the subtitle in live subtitling is about 6 seconds.

<sup>8</sup> More information about Morae can be found on the Techsmith website <http://www.techsmith.com>

## ACKNOWLEDGEMENT

We would like to thank Robbe Block, Joris Roovers, Eric Van Horenbeeck, Alexander Prinssier, Wesley Cabus, Ahmed Essahli, Wim Claesssens, Mathia Van de Poel and Bart Van de Velde for their excellent work in programming Inputlog (BOF 2005-2011). We would also like to thank Nico Verlinden and the KdG-Polytechnic Antwerp for providing internships and a collective project on Inputlog (PWO 2005-2008). We would also like to thank Stijn van Even, Guido Gallopyn, Hans Geuns and Neil Grant of Nuance (previous Scansoft) for all their efforts in making the logging facility at Dragon Naturally Speaking available to us.

We would like to thank Eva Lindgren, Daphne van Weijen, Tijs Delbeke, Bieke Luyckx and Iris Schrijver for their cooperation on the described research projects. For the research projects conducted at the Flemish Broadcasting Station VRT we would like to thank Bernard Dewulf and Erik Desnerck for their enthusiasm and time. Finally, we thank Asa Wengelin, Victoria Johansson, Roger Johansson and Eva Lindgren for the many fruitful discussions on the development of logging techniques and methods.

## References

- Alamargot, D., Chesnet, D., Dansac, C., & Ros, C. (2006). Eye and Pen: A new device for studying reading during writing. *Behavior Research Methods Instruments & Computers*, 38(2), 287-299.
- Andersson, B., Dahl, J., Holmkvist, K., Holsanova, J., Johansson, V., Karlsson, H., et al. (2006). Combining keystroke-logging with eye-tracking. In L. van Waes, M. Leijten & C. Neuwirth (Eds.), *Writing and digital media* (pp. 166-172). Amsterdam: Elsevier.
- Andersson, B., & et al. (2006). Combining keystroke logging with eye-tracking. In L. Van Waes, M. Leijten & C. Neuwirth (Eds.), *Writing and Digital Media* (Vol. 17, pp. 166-172). Oxford: Elsevier.
- Bereiter, C., & Scardamalia, M. (1987). *The psychology of written composition*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Braaksma, M., Rijlaarsdam, G., & Janssen, T. (2007). Writing Hypertexts: Proposed effects on writing processes and knowledge acquisition. *L1-Educational Studies in Language and Literature*, 7(4).
- Bridwell, L. S., & Duin, A. H. (1985). Looking in-depth at writers: Computers as writing medium and research tool. In J. L. Collins & E. A. Sommers (Eds.), *Writing On-Line* (pp. 115-121). Upper Montclair, NJ: Boynton/Cook.
- Brown, P., & Levinson, S. (1987). *Politeness: Some universals in language usage*: Cambridge Univ Pr.
- Chenoweth, N., & Hayes, J. R. (2001). Fluency in writing. *Written Communication*, 18(1), 80-98.
- Dam-Jensen, H., & Heine, C. (2009). Process Research Methods and Their Application in the Didactics of Text Production and Translation: Shedding Light on the Use of Research Methods in the University Classroom. *trans-kom. Zeitschrift für Translationswissenschaft und Fachkommunikation*, 2(1), 1-25.
- De Smet, M. (2009). Schriftelijke slechtnieuwscommunicatie. De invloed van kennis en ervaring op het schrijven (Master Thesis Master in Multilingual Professional Communication ed.). Antwerp: University of Antwerp.
- Degenhardt, M. (2006). Camtasia and Catmovie: Two digital tools for observing, documenting and analysing writing processes of university students. In L. Van Waes, M. Leijten & C. Neuwirth (Eds.), *Writing and Digital Media* (Vol. 17, pp. 180-188). Oxford: Elsevier.
- Douhou, S., & Magnus, J. (2009). The reliability of user authentication through keystroke dynamics. *Statistica Neerlandica*, 63(4), 432-449.

- Faigley, L., & Witte, S. (1981). Analyzing revision. *College Composition and Communication*, 32, 400-414.
- Flower, L., & Hayes, J. R. (1980). The dynamics of composing: Making plans and juggling constraints. In L. Gregg & E. Steinberg (Eds.), *Cognitive processes in writing* (pp. 31-50). Hillsdale, NJ: L. Erlbaum.
- Flower, L., & Hayes, J. R. (1985). 'Talking about protocols'. *College Composition and Communication*, 37, 16-54.
- Flower, L. S., & Hayes, J. R. (1980). The dynamics of composing: Making plans and juggling constraints. In L. W. Gregg & E. R. Steinberg (Eds.), *Cognitive Processes in Writing*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Göpferich, S. (2008). *Translationsprozessforschung: Stand, Methoden, Perspektiven*. BoD—Books on Demand.
- Grabowski, J. (2008). The internal structure of university students' keyboard skills. *Journal of Writing Research*, 1(1), 27-52.
- Greene, S., & Higgins, L. (1994). 'Once upon a time': the use of retrospective accounts in building theory in composition. In P. Smagorinsky (Ed.), *Speaking about Writing* (pp. 115-140). London: Sage.
- Hayes, J. R. (1996). A new model of cognition and affect in writing. In C. M. Levy & S. Ransdell (Eds.), *The science of writing* (pp. 1-30). Hillsdale, NJ: Erlbaum.
- Hayes, J. R., & Flower, L. S. (1980). Identifying the organization of writing processes. In L. W. Gregg & E. R. Steinberg (Eds.), *Cognitive Processes in writing* (pp. 3-30). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Honeycutt, L. (2003). Researching the use of voice recognition writing software. *Computers and Composition*, 20, 77-95.
- Hyönä, J., & Lorch, R. (2004). Effects of topic headings on text processing: evidence from adult readers' eye fixation patterns. *Learning and Instruction*, 14(2), 131-152.
- Jacobs, G., Opdenacker, L., & Van Waes, L. (2005). A multilanguage online writing center for professional communication: Development and testing. *Business Communication Quarterly*, 68(1), 8-22.
- Jakobsen, A. L. (2003). Effects of think aloud on translation speed, revision and segmentation. *Triangulating Translation. Perspectives in process oriented research*, 69-95.
- Jakobsen, A. L. (2006). Research methods in Translation - Translog. In K. P. H. Sullivan & E. Lindgren (Eds.), *Computer keystroke-logging and writing* (Vol. 18, pp. 95-105). Amsterdam: Elsevier.
- Jansen, F., & Janssen, D. (2010). Effects of positive politeness strategies in business letters. *Journal of Pragmatics*.
- Jansen, F., & Janssen, D. (2011). Explanations First: A Case for Presenting Explanations Before the Decision in Dutch Bad-News Messages. *Journal of Business and Technical Communication*, 25(1), 36.
- Janssen, D., Van Waes, L., & Van den Bergh, H. (1996). Effects of thinking aloud on writing processes. In C. M. Levy & S. Ransdell (Eds.), *The science of writing: Theories, individual differences, and applications* (pp. 233-250). Mahwah, NJ: Lawrence Erlbaum Associates.
- Johansson, R., Johansson, V., Wengelin, Å., & Holmqvist, K. (2009). Reading during writing: Four different groups of writers. *Lund Working Papers in Linguistics*, 53(0), 43-59.
- Johansson, R., Wengelin, Å., Johansson, V., & Holmqvist, K. Looking at the keyboard or the monitor: relationship with text production processes. *Reading and Writing*.

- Kaufert, D. S., Hayes, J. R., & Flower, L. (1986). Composing written sentences. *Research in the Teaching of English*, 20(2), 121-140.
- Kellogg, R. T. (2008). Training writing skills: A cognitive developmental perspective. *Journal of writing research*, 1(1), 1-26.
- Kolb, D. A. (1984). *Experiential Learning*. Englewood Cliffs, New Jersey: Prentice Hall Inc.
- Kollberg, P. (1998). *S-Notation - a computer based method for studying and representing text composition*. Lic. Thesis, Stockholm University, Stockholm.
- Krings, H. (2001). *Repairing Texts: Empirical Investigations of Machine Translation Post-Editing Processes* (G. S. Koby, Trans.). Kent: Kent State University Press.
- Latif, M. M. A. (2009). A state-of-the-art review of the real-time computer-aided study of the writing process. *International Journal of English Studies*, 8(1), 29-50.
- Leijten, M. (2007a). How do writers adapt to speech recognition software? The influence of learning styles on writing processes in speech technology environments. In M. Torrance, L. Van Waes & D. Galbraith (Eds.), *Writing and Cognition: Research and Applications* (Vol. 20, pp. 279-292). Oxford: Elsevier.
- Leijten, M. (2007b). *Writing and speech recognition: observing error correction strategies of professional writers* (Vol. 160). Utrecht: LOT.
- Leijten, M., Ransdell, S., & van Waes, L. (2010). Correcting Text Production Errors: Isolating the Effects of Writing Mode From Error Span, Input Mode, and Lexicality. *Written Communication*, 27(2), 189-227. doi: DOI: 10.1177/0741088309359139
- Leijten, M., & Van Waes, L. (2006). Inputlog: New perspectives on the logging of on-line writing processes in a Windows environment. In K. P. H. Sullivan & E. Lindgren (Eds.), *Computer Key-Stroke Logging and Writing: Methods and Applications*. (Vol. 18, pp. 73-94). Oxford: Elsevier.
- Leijten, M., & Van Waes, L. (2006a). Inputlog: New Perspectives on the Logging of On-Line Writing. In K. P. H. Sullivan & E. Lindgren (Eds.), *Computer Key-Stroke Logging and Writing: Methods and Applications* (Vol. 18, pp. 73-94). Oxford: Elsevier.
- Leijten, M., & Van Waes, L. (2006b). Repair strategies in writing with speech recognition: The effect of experience with classical dictating. In L. Van Waes, M. Leijten & C. Neuwirth (Eds.), *Writing and Digital Media* (Vol. 17, pp. 31-46). Oxford: Elsevier.
- Leijten, M., Van Waes, L., & Janssen, D. (2010). Error correction strategies of professional speech recognition users: three profiles. *Computers in Human Behaviour*, 26, 964-975. doi: 10.1016/j.chb.2010.02.010
- Leijten, M., Van Waes, L., & Ransdell, S. (2010). Correcting Text Production Errors: Isolating the Effects of Writing Mode From Error Span, Input Mode, and Lexicality. *Written communication*, 27(2), 189.
- Levy, C. M., & Ransdell, S. (1994). Computer-aided protocol analysis of writing processes. *Behavior Research Methods, Instruments and Computers*, 26(2), 219-223.
- Lindgren, E. (2002). The LS graph: A methodology for visualising writing revision. *Language Learning*, 52(3), 565-595.
- Lindgren, E. (2004). The uptake of peer-based intervention in the writing classroom. In G. Rijlaarsdam, H. Van den Bergh & M. Couzijn (Eds.), *Effective learning and teaching of writing* (Vol. 14, pp. 259-274). Dordrecht: Kluwer Academic Publishers.
- Lindgren, E. (2007). GIS for writing: Applying geographic information system techniques to data-mine writing's cognitive processes. In M. Torrance, L. Van Waes & D. Galbraith (Eds.), *Writing and Cognition: Methods and Applications* (Vol. 20, pp. xx-xx). Oxford: Elsevier.
- Lindgren, E., Leijten, M., & Van Waes, L. (2008). *Writing development & Knowledge crafting*. Paper presented at the Writing across Borders II, St. Barbara.

- Lindgren, E., & Sullivan, K. P. H. (2006). Analyzing on-line revision. In K. P. H. Sullivan & E. Lindgren (Eds.), *Computer Key-Stroke Logging: Methods and Applications* (Vol. 18, pp. 157-188). Oxford: Elsevier.
- Luyckx, B., Delbeke, T., Van Waes, L., Remael, A., & Leijten, M. (2010). Live subtitling with speech recognition: Causes and consequences of text reduction. In K. Peeters (Ed.), *Artesis working papers in translation studies* (Vol. 1, pp. 39). Antwerp: Artesis.
- MacArthur, C. A. (2006). Assistive technology for writing: Tools for struggling writers. In L. Van Waes, M. Leijten & C. Neuwirth (Eds.), *Writing and Digital Media* (Vol. 17, pp. 11-20). Oxford: Elsevier.
- Matsuhashi, A. (1981). Pausing and planning: The tempo of written discourse production. *Research in the Teaching of English*, 15(2), 113-134.
- Matsuhashi, A. (1982). Explorations in real-time production of written discourse. In M. Nystrand (Ed.), *What writers know. The language, process, and structure of written discourse* (pp. 269-290). New York: Academic Press.
- Nottbusch, G. (2010). Grammatical planning, execution, and control in written sentence production. *Reading and Writing*, 23(7), 777-801.
- Nottbusch, G., Weingarten, R., & Sahel, S. (2007). From written word to written sentence production. In M. Torrance, L. Van Waes & D. Galbraith (Eds.), *Writing and Cognition: Research and applications* (Vol. 20, pp. 31-54). Amsterdam: Elsevier.
- Olive, T., & Kellogg, R. T. (2002). Concurrent activation of high- and low-level production processes in written composition. *Memory and Cognition*, 30, 594-600.
- Opdenacker, L., & Van Waes, L. (2007). Implementing an open process approach to a multilingual online writing center; The case of Calliope. *Computers and Composition*, 24(3), 247-265.
- Opdenacker, L., Verheyen, A., & Van Waes, L. (2007). Implementing an online writing center in career management services: Six perspectives on quality assessment. In K. Pelsmaekers (Ed.), *Economically speaking* (pp. 199-210). Antwerpen: Garant.
- Perrin, D. (2003). Progression analysis: Investigating writing strategies at the workplace. *Journal of Pragmatics*, 35(6), 907-921.
- Piolat, A., Kellogg, R. T., & Farioli, F. (2001). The triple task technique for studying writing processes: on which task is attention focused? (Current psychology letters). *Brain, Behavior and Cognition*, 4, 67-83.
- Piolat, A., Olive, T., & Kellogg, R. T. (2004). Cognitive effort during notetaking. *Applied Cognitive Psychology*, 18(1), 1-22.
- Quinlan, T. (2006). Young Writers and Digital Scribes. In L. Van Waes, M. Leijten & C. Neuwirth (Eds.), *Writing and Digital Media* (Vol. 17, pp. 21-29). Oxford: Elsevier.
- Rayner, K. (1975). The perceptual span and peripheral cues in reading. *Cognitive Psychology*, 7, 65-81.
- Remael, A. (2007). Sampling subtitling for the deaf and the hard-of-hearing in Europe. In J. Díaz Cintas, A. Remael & P. Orero (Eds.), *Media for All* (pp. 23-52). Amsterdam: Rodopi.
- Richards, T., Berninger, V., Winn, W., Swanson, H. L., Stock, P., Lianga, O., et al. (2009). Differences in fMRI activation between children with and without spelling disability on 2-back/0-back working memory contrast. *Journal of Writing Research*, 1(2), 93-123.
- Rijlaarsdam, G., Braaksma, M., Couzijn, M., Janssen, T., Raedts, M., Van Steendam, E., et al. (2008). Observation of peers in learning to write: Practice and research. *Journal of Writing Research*, 1(1), 53-83.

- Sahel, S., Nottbusch, G., Grimm, A., & Weingarten, R. (2008). Written production of German compounds: Effects of lexical frequency and semantic transparency. *Written Language and Literacy*, 11(2), 211-228.
- Schilperoord, J. (1996). *It's about time: Temporal aspects of cognitive processes in text production*. Amsterdam/Atlanta: Rodopi.
- Schrijver, I., Van Vaerenbergh, L., & Van Waes, L. (2011). Transediting in students' translation processes. In K. Peeters (Ed.), *Artesis VT working papers in translation studies* (pp. 31). Antwerp: Artesis University College.
- Severinson Eklundh, K. S. (1994). Linear and Non-linear strategies in computer-based writing. *Computers and Composition*, 11, 203-216.
- Severinson Eklundh, K. S., & Kollberg, P. (1992). Translating keystroke records into a general notation for the writing process Stockholm: Department of Numerical Analysis and Computing Science, Royal Institute of Technology.
- Severinson Eklundh, K. S., & Kollberg, P. (1996a). Computer tools for tracing the writing process: From keystroke records to S-notation. In G. Rijlaarsdam, H. Van den Bergh & M. Couzijn (Eds.), *Theories, Models and Methodology in writing research* (pp. 526-541). Amsterdam: University Press.
- Severinson Eklundh, K. S., & Kollberg, P. (1996b). Computer tools for tracing the writing process: From keystroke records to S-notation. In G. Rijlaarsdam, H. Van den Bergh & M. Couzijn (Eds.), *Models and Methodology in writing research* (pp. 526-541). Amsterdam: Amsterdam University Press, .
- Severinson Eklundh, K. S., & Kollberg, P. (2003). Emerging discourse structure: Computer-assisted episode analysis as a window to global revision in university students' writing. *Journal of Pragmatics*, 35, 869-891.
- Simpson, S., & Torrance, M. (Producer). (2007). EyeWrite (Version 5.1).
- Smagorinsky, P. (1989). The reliability and validity of protocol analysis. *Written communication*, 6, 463-479.
- Spelman Miller, K. (2000). Academic writers on-line: Investigating pausing in the production of text. *Language Teaching Research*, 4, 123-148.
- Spelman Miller, K., & Sullivan, K. P. H. (2006). Keystroke logging – an introduction. In K. P. H. Sullivan & E. Lindgren (Eds.), *Computer Keystroke Logging and Writing: Methods and Applications*. Amsterdam: Elsevier.
- Strömqvist, S., Holmqvist, K., Johansson, V., Karlsson, H., & Wengelin, Å. (2006). What key-logging can reveal about writing. In K. P. H. Sullivan & E. Lindgren (Eds.), *Computer Key-stroke Logging and Writing: Methods and Applications*. Amsterdam: Elsevier.
- Strömqvist, S., & Karlsson, H. (2001). *ScriptLog for Windows: Users manual*. Lund: University of Lund.
- Sullivan, K. P. H., & Lindgren, E. (2006). *Computer Key-Stroke Logging and Writing*. Oxford: Elsevier Science.
- Van den Bosch, A., Busser, G. J., Daelemans, W., & Canisius, S. (2007). An efficient memory-based morphosyntactic tagger and parser for Dutch. In P. Van Eynde, P. Dirix, I. Schuurman & V. Vandeghinste (Eds.), *Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting* (pp. 99-114). Leuven.
- Van den Haak, M. J., De Jong, M. D. T., & Schellens, P. J. (2003). Retrospective versus concurrent think-aloud protocols: Testing the usability of an online library catalogue. *Behaviour and Information Technology*, 22(5), 339-351.
- Van Hout, T. (2010). *Writing form sources: ethnographic insights into business news production*. Ghent Ghent University.

- Van Hout, T., & Macgilchrist, F. (2010). Framing the news: an ethnographic view of financial newswriting. *Text & Talk*, 30(2), 147-169.
- Van Waes, L., & Leijten, M. (2006). Logging writing processes with Inputlog. In L. Van Waes, M. Leijten & C. Neuwirth (Eds.), *Writing and Digital Media* (Vol. 17, pp. 158-166). Oxford: Elsevier.
- Van Waes, L., Leijten, M., & Neuwirth, C. (Eds.). (2006). *Writing and Digital Media* (Vol. 17). Oxford: Elsevier.
- Van Waes, L., Leijten, M., & Quinlan, T. (2010). Reading during sentence composing and error correction: A multilevel analysis of the influences of task complexity. *Reading and Writing*, 23(7), 803-834.
- Van Waes, L., Leijten, M., & Quinlan, T. (in press). Reading during sentence composing and error correction: A multilevel analysis of the influences of task complexity. *Reading and Writing: An Interdisciplinary Journal*.
- Van Waes, L., Leijten, M., & Van Weijen, D. (2009). Keystroke logging in writing research: Observing writing processes with Inputlog. *GFL-German as a foreign language*, 2(3), 41-64.
- Van Waes, L., Leijten, M., Wengelin, A., & Lindgren, E. (2011). Logging Tools to Study Digital Writing Processes. In V. Berninger (Ed.), *Past, Present, and Future Contributions of Cognitive Writing Research to Cognitive Psychology*. London: Psychology Press, Taylor & Francis Group.
- Van Waes, L., & Schellens, P. J. (2003). Writing profiles: The effect of the writing mode on pausing and revision patterns of experienced writers. *Journal of Pragmatics*, 35(6), 829-853.
- Van Weijen, D. (2008). Writing processes, text quality, and task effects. *Utrecht, the Netherlands: LOT*.
- Van Weijen, D. (2009). Writing profiles, learning styles and bad news letters *Quadem reports* (pp. 20). Antwerp: University of Antwerp.
- Vandeghinste, V. (2008). *A Hybrid Modular Machine Translation System. LoRe-MT: Low Resources Machine Translation*. PhD thesis, KULeuven, Louvain.
- Weingarten, R., Nottbusch, G., & Will, U. (2004). Morphemes, syllables and graphemes in written word production. In T. Pechmann & C. Habel (Eds.), *Multidisciplinary approaches to speech production* (pp. 529-572). Berlin: Mouton de Gruyter.
- Wengelin, A. (2006). Examining pauses in writing: Theories, methods and empirical data. In K. P. H. Sullivan & E. Lindgren (Eds.), *Computer Key-Stroke logging and Writing: Methods and Applications* (Vol. 18, pp. 107-130). Oxford: Elsevier.
- Wengelin, A., Leijten, M., & Van Waes, L. Studying reading during writing: new perspectives in research. *Reading and Writing*.
- Wengelin, A., Torrance, M., Holmqvist, K., Simpson, S., Galbraith, D., Johansson, V., et al. (2009a). Combined eye-tracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior Research Methods*, 41, 337-351.
- Wengelin, A., Torrance, M., Holmqvist, K., Simpson, S., Galbraith, D., Johansson, V., et al. (2009b). Combined eye-tracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior Research Methods*, 41(2), 337-351.

## Figures

```

<Event>
  <WritingMode>2</WritingMode>
  <Output>LeftButton</Output>
  <StartClock>0:00:16</StartClock>
  <StartTime>16672</StartTime>
  <EndClock>0:00:16</EndClock>
  <EndTime>16797</EndTime>
  <ActionTime>125</ActionTime>
  <PauseTime>16672</PauseTime>
  <X>252</X>
  <Y>55</Y>
</Event>

```

Figure 1. Example of XML notation used in Inputlog 4.1.

inputlog										
General Logging File										
WritingMode	Output	StartClock	StartTime	EndClock	EndTime	ActionTime	PauseTime	X	Y	
2	Movement	0:00:00	0	0:00:16	16532	16532	0	252	55	
2	Left Button	0:00:16	16672	0:00:16	16797	125	16672	252	55	
2	Movement	0:00:16	16891	0:00:17	17891	1000	219	219	260	
2	Left Button	0:00:17	17907	0:00:17	17969	62	1016	219	260	
3 - dict	context in	0:04:39	279051	0:04:40	280577	1526	1673			
1	ENTER	0:04:42	282250	0:04:42	282328	78	3199			
1	ENTER	0:04:42	282578	0:04:42	282641	63	328			
3 - dict	de algemene verplichtingen van de werkgever met betrekking tot de bescherming van gezondheid en veiligheid van werknemers op de werkplaats	0:04:43	283502	0:04:51	291922	8420	924			
3 - dict	.punt	0:04:55	295051	0:04:55	295928	877	3129			
1	BS	0:05:01	301594	0:05:01	301688	94	5666			
1	SPACE	0:05:02	302297	0:05:02	302375	78	703			
3 - dict	kaderen in de ontwikkeling ervan een preventiebeleid .punt	0:05:02	302687	0:05:06	306248	3561	390			

Figure 2. General logging file with data of Dragon Naturally Speaking.

interval	Output
00:00:00	{2688}[LeftButton]{2859}The·quick·brown·{7172}fo
00:00:18	x·jumps·over·the·lazy·{5438}dog.[Movement][RightButton][Movement][LeftButton]

Figure 3. Example of linear file



Figure 4. Example of replay full writing process and revisions.

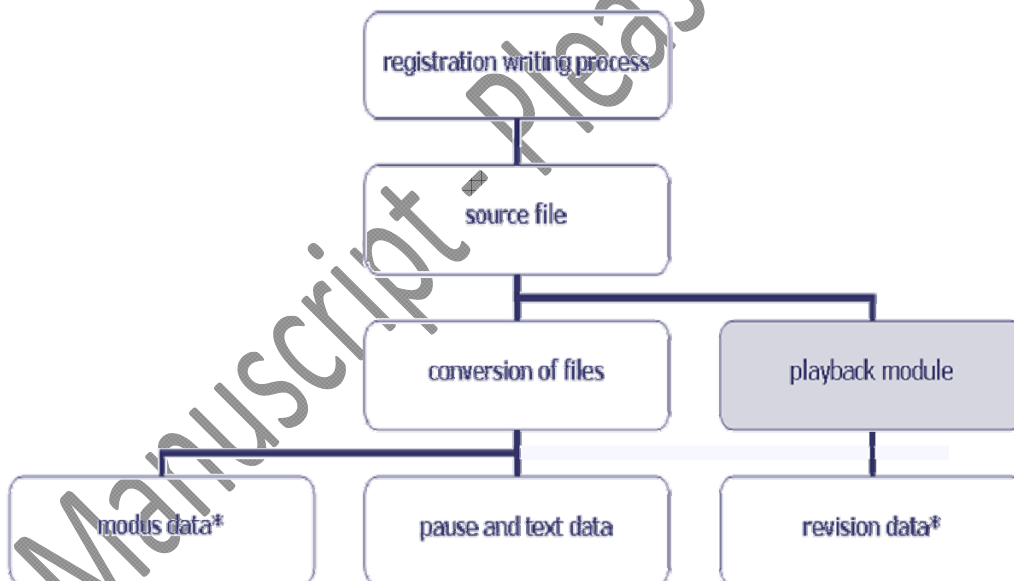


Figure 5. Flow of Inputlog.

Process time	Linear logging file	Product data
4:54-7:13 (argumentation)	<p>Bij•nazicht•van•uw{1248}[LeftButton]•aanvraagformulier{1639}[Movement][LeftButton][Movement]{1077}[Movement][LeftButton][Movement]{2995}[<i>technical flaw</i>]{1747}•aanvra{1685}ag{1935}formulier•[Movement]{11856}hebben•wij•vastgesteld•dat•de•inkomsten•van{3807}•uw•inwonende•zoon•door•u•niet•werden•vermeld{1279}•{10765}[ENTER]{1294}</p> <p>1 revision: addition of name son: Jan (15:35) Fluency:16,2 words per minute</p>	<p>[Bij nazicht van uw aanvraagformulier hebben wij vastgesteld dat de inkomsten van uw inwonende <i>zoon Jan</i> door u niet werden vermeld.]</p> <p>When checking your application form we noticed that you failed to declare the income of your living-in <i>son Jan</i>.</p>
11:58-12:56 (FTA)	<p>Tot•onze•spijt•moeten•wij•u•dan•ook•medelen•dat•u•niet{6661}[BS4]•momenteel•niet{6615}•aan•de{2169}•{2933}vastgestelde•in{1077}kome nsvoorwa{1170}arden•voldoet{8658}•{3994}</p> <p>1 revision: [niet &gt; momenteel niet]; not &gt; at present not Fluency:18,6 words per minute</p>	<p>[Tot onze spijt moeten wij u dan ook mededelen dat u momenteel niet aan de vastgestelde inkomensvoorwaarden voldoet.]</p> <p>We regret to inform you that at present you do not meet the determined income conditions.</p>

Figure 6. Production of Face Threatening Act by expert writer.

Output	Position	Doclength	StartTime	StartClock	EndTime	EndClock	ActionTime	PauseTime
Calliope - Slechtnieuwsbrieven - Structuur – Microsoft IE			<a href="#">2157735</a>	00:35:57	2157735	00:35:57	0	0
Movement			<a href="#">2157735</a>	00:35:57	2167750	00:36:07	10015	0
Movement			<a href="#">2169281</a>	00:36:09	2175406	00:36:15	6125	1531
Movement			<a href="#">2180094</a>	00:36:20	2182719	00:36:22	2625	4688
Left Button			<a href="#">2182797</a>	00:36:22	2182922	00:36:22	125	78
Calliope - Slechtnieuwsbrieven - Onderdelen – Microsoft IE			<a href="#">2182797</a>	00:36:22	2182797	00:36:22	0	0
Movement			<a href="#">2182922</a>	00:36:22	2184250	00:36:24	1328	0
Movement			<a href="#">2187813</a>	00:36:27	2188125	00:36:28	312	3563
Movement			<a href="#">2200672</a>	00:36:40	2201922	00:36:41	1250	12547
Left Button			<a href="#">2201985</a>	00:36:41	2202094	00:36:42	109	63
document1.doc – Microsoft Word			<a href="#">2201985</a>	00:36:41	2201985	00:36:41	0	0
Movement			<a href="#">2202110</a>	00:36:42	2202531	00:36:42	421	16
B	2	2	<a href="#">2203500</a>	00:36:43	2203781	00:36:43	281	969
t	3	3	<a href="#">2203797</a>	00:36:43	2203797	00:36:43	0	297
e	4	4	<a href="#">2203797</a>	00:36:43	2203891	00:36:43	94	0
r	5	5	<a href="#">2203875</a>	00:36:43	2203969	00:36:43	94	78
f	6	6	<a href="#">2204031</a>	00:36:44	2204125	00:36:44	94	156
e	7	7	<a href="#">2204094</a>	00:36:44	2204188	00:36:44	94	63
t	8	8	<a href="#">2204219</a>	00:36:44	2204297	00:36:44	78	125
BS	7	7	<a href="#">2204516</a>	00:36:44	2204594	00:36:44	78	297
BS	6	6	<a href="#">2204672</a>	00:36:44	2204750	00:36:44	78	156
BS	5	5	<a href="#">2204813</a>	00:36:44	2204875	00:36:44	62	141
BS	4	4	<a href="#">2204938</a>	00:36:44	2205016	00:36:45	78	125
BS	3	3	<a href="#">2205094</a>	00:36:45	2205188	00:36:45	94	156
BS	2	2	<a href="#">2205297</a>	00:36:45	2205344	00:36:45	47	203
e	3	3	<a href="#">2205781</a>	00:36:45	2205860	00:36:45	79	484
t	4	4	<a href="#">2205844</a>	00:36:45	2205922	00:36:45	78	63
r	5	5	<a href="#">2206000</a>	00:36:46	2206110	00:36:46	110	156
e	6	6	<a href="#">2206078</a>	00:36:46	2206172	00:36:46	94	78
f	7	7	<a href="#">2206156</a>	00:36:46	2206266	00:36:46	110	78
t	8	8	<a href="#">2206328</a>	00:36:46	2206422	00:36:46	94	172

Figure 7. Fragment from the Inputlog general file from the Calliope study.

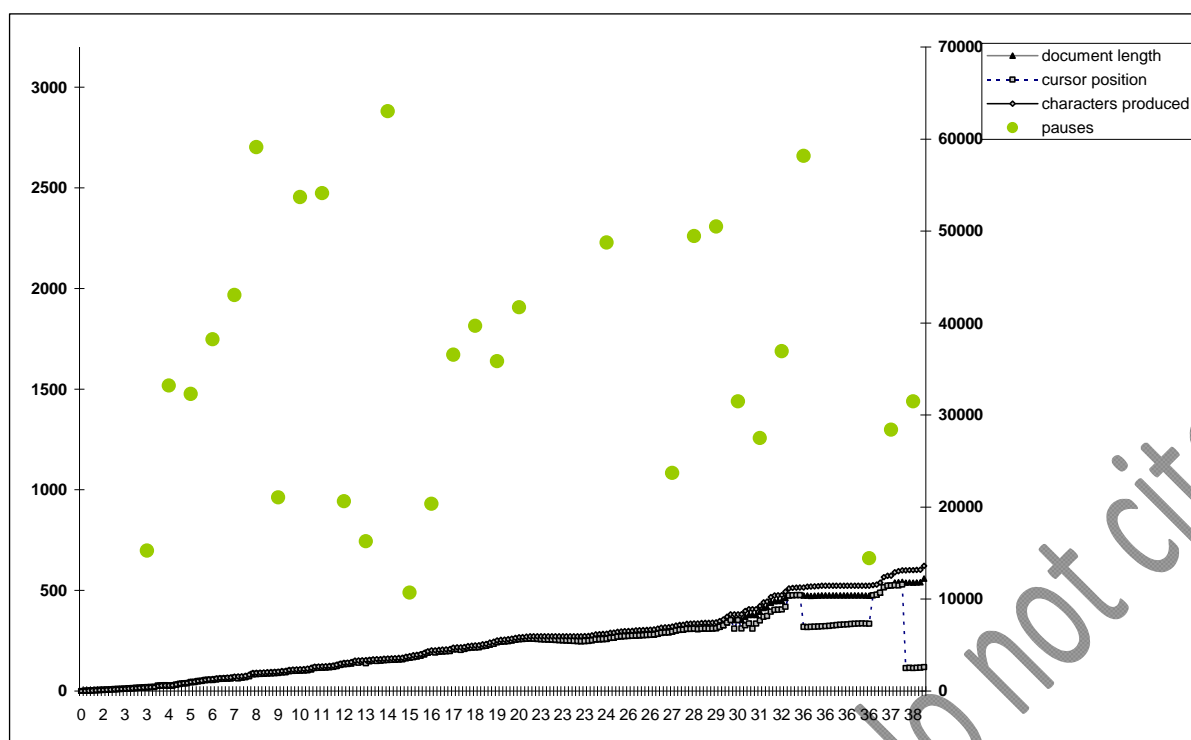


Figure 8. Graphical representation of Judith's writing process (instructional text).

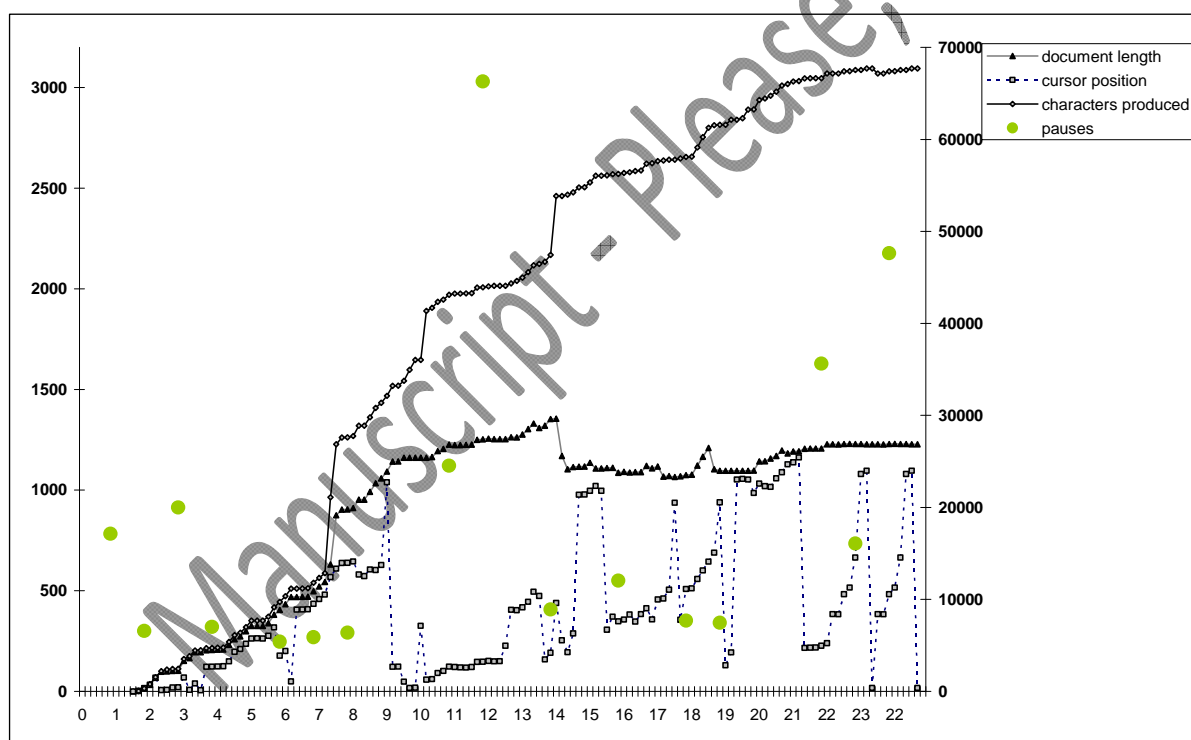


Figure 9. Graphical representation of Dieter's writing process (instructional text).

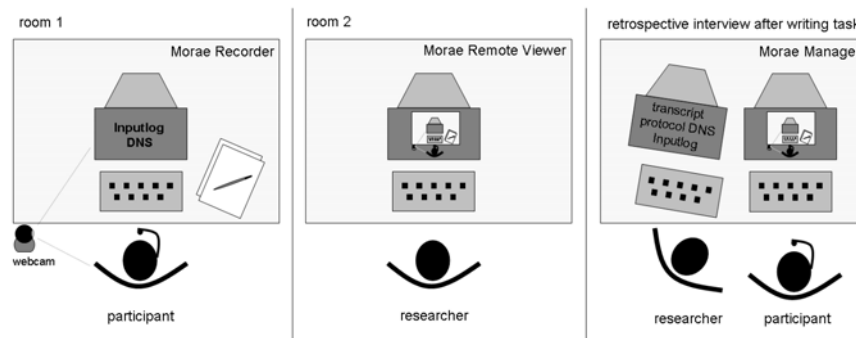


Figure 10. Observation setting of experiment.

Morae	Excel & SPSS	Morae	Morae	Inputlog	Inputlog & Dragon Naturally Speaking	Inputlog	Inputlog
ClockTime	Mergetime	Marker	Code	WritingMode	Output	StartClock	PauseTime
	214000			3 - dict	context in	0:04:39	1673
	218000			1	ENTER	0:04:42	3199
	218000			1	ENTER	0:04:42	328
	219000			3 - dict	de algemene verplichtingen van de werkgever met betrekking tot de bescherming van gezondheid en veiligheid van werknemers op de werkplaats	0:04:43	924
	230000			3 - dict	.punt	0:04:55	3129
	237000			1	BS	0:05:01	5666
	238000			1	SPACE	0:05:02	703
	238000			3 - dict	kaderen in de ontwikkeling ervan een preventiebeleid .punt	0:05:02	390
0,05,42.77	244000	Marker	O				.

Figure 11. Example of merged logging data of Inputlog and Dragon Naturally Speaking.

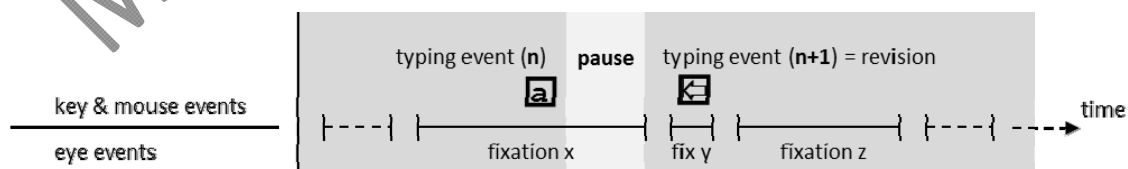


Figure 12. A revision as a critical discourse event during writing

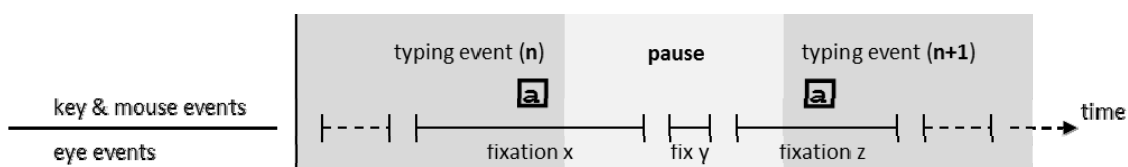


Figure 13. A pause as a critical discourse event during writing.

Manuscript - Please, do not cite

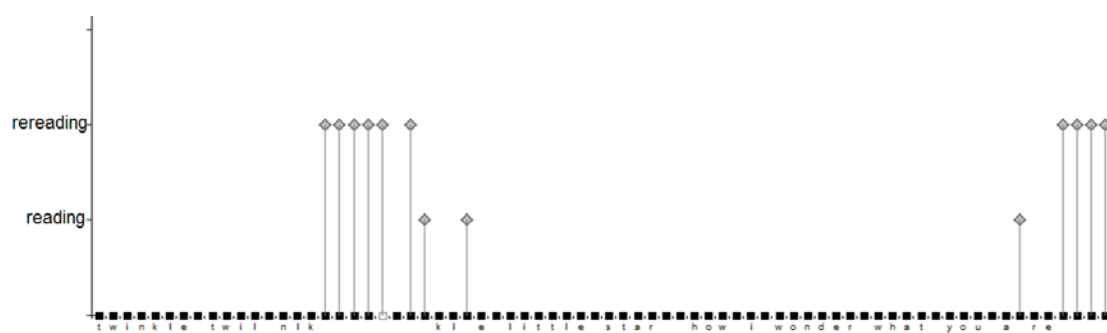


Figure 14. Reading during writing behavior of hunt-and-peck typist.

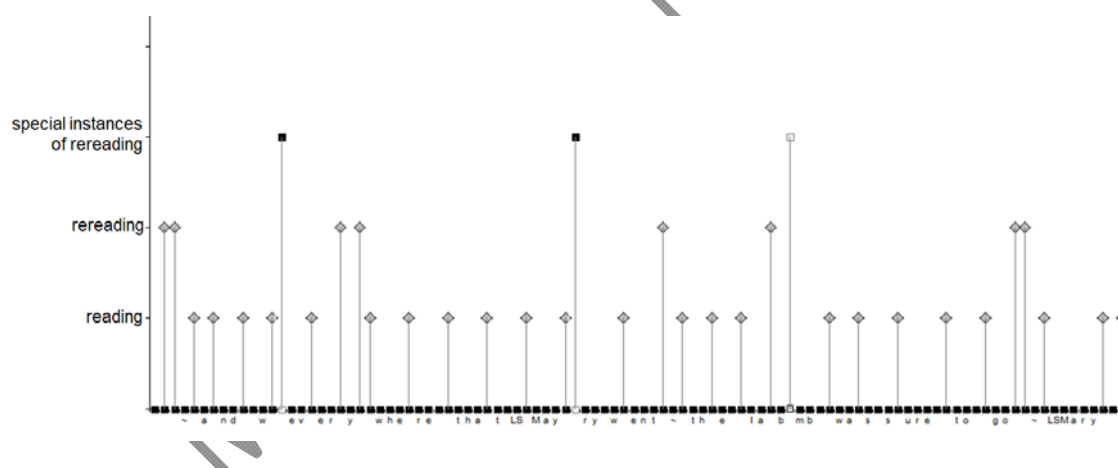


Figure 15. Reading during writing behavior of monitorgazer.

## Tables

Table 2

Classification of writing observation methods (based on Janssen, Van Waes & Van den Bergh, 1996)

	direct research methods	indirect research methods
synchronous	concurrent think aloud protocols prompted pauses	keystroke logging video observation double task method eye-tracking EVP or fMRI
asynchronous	retrospective protocols	text analysis versioning

Table 2. Example of a combined TAP and a linear KSL representation

Thinking aloud protocol			Inputlog output	
Start time	Transcript protocol	Classification (Klings' model)	Start time	Linear representation
00:51:43	het is aangewezen geen alcoholische dranken te gebruiken (-) te consumeren (----) te gebruiken  <i>[It is appropriate not to use (-) to consume (---) to use alcoholic beverages]</i>	TARGET/PROD/ CONCRETE/VARIANT	00:51:43	Hezt-[BS4]t-is-aangewezen-geen -alcoholische-dranken-te-
			00:51:53	{3031}
			00:51:55	gebruiken [use]

Table 3. Example of a combined TAP and a linear keystroke logging representation

Thinking aloud protocol			Inputlog output	
Start time	Transcript protocol	Classification (Krings' model)	Start time	Linear representation
00:17:35	(7.0)		00:17:37	\$(BS)■- wanneer- [when]
00:17:42	"wanneer" / een opsomming in puntjes omdat dat goed is voor de leesbaarheid hm <i>["when" / a bulleted list because that is good for the readability]</i>	TARGET/RED/EMEA/ TECH/OPMAAK/MOT	00:17:41	{56906}
00:17:50	(2.0)			
00:17:52	en dus die "wanneer" moet dan altijd herhaald worden (----) hm <i>[and so the "when" should always be repeated]</i>	TARGET/RED/EMEA/ INHOUD/CONSIST		
00:17:57	(19.0)			
00:18:16	dus het rubriekske "who should not take Geodon" hoort hier eigenlijk bij <i>[thus the section "who should not take Geodon" belongs here]</i>	TARGET/RED/EMEA/ INHOUD/RUBR/PPI		
00:18:20	(3.0)			
00:18:23	Hm			
00:18:24	(2.0)			
00:18:26	dus als u hm <i>[so when you]</i>	TARGET/PROD/ CONCRETE		
00:18:28	(10.0)			
00:18:38	dus wanneer u lijdt aan een (----) hm (----) psychose (----) gerelateerd nee (----) hm <i>[so when you suffer from a psychosis related to / no]</i>	TARGET/PROD/ ABSTRACT/PPI	00:18:38	u-lijdt-aan-u[BS]leen- [you suffer from a]
			00:18:43	{2922}
			00:18:45	psychose-geraltee [psychosis related]

Table 4. Example of a merged general log file (keyboard, mouse, speech)

Medium	Output	Starting ms	Time h:mm:ss
speech	hij·behoorde·bij·de·groepen·die·zich·niet·wou ·aansluiten·bij·het·vee·en·vee [he belonged to groups who did not want to join the <i>cattle and cattle</i> ]	3347755	0:55:47
mouse	Movement	3350986	0:55:50
keyboard	UP	3353793	0:55:53
keyboard	BS	3354454	0:55:54
keyboard	BS	3355565	0:55:55
keyboard	BS	3355726	0:55:55
keyboard	V	3356306	0:55:56
keyboard	N	3356527	0:55:56
keyboard	V	3356727	0:55:56
keyboard	SHIFT + CTRL + LEFT	3361053	0:56:01
keyboard	SHIFT + CTRL + LEFT	3361193	0:56:01
keyboard	BS	3361874	0:56:01
speech	tijdens·de·oorlog. [during the war]	3362976	0:56:02
keyboard	PD	3363096	0:56:03
swift	Hij·behoorde·bij·de·groepen·die·zich·niet·wou ·aansluiten·bij·het·VNV tijdens de oorlog. [He belonged to groups who did not want to join the VNV during the war.]	3363296	0:56:03
speech	met·de·top·van·de·arbeidersbeweging [with the top of the labor movement]	3364690	0:56:04