

Translog-II: A Program for Recording User Activity Data for Empirical Translation Process Research

MICHAEL CARL

Copenhagen Business School, Denmark

ABSTRACT

This paper presents a novel implementation of Translog-II. *Translog-II is a Windows-oriented program to record and study reading and writing processes on a computer. In our research, it is an instrument to acquire objective, digital data of human translation processes. As their predecessors, Translog 2000, Translog 2006, also Translog-II consists of two main components: Translog-II Supervisor and Translog-II User. The two components are interdependent: Translog-II User serves to record experimental translation sessions, while in Translog-II Supervisor project files can be generated, log files replayed and analysed. This paper gives an overview of the tool and its data visualization options.*

KEYWORDS: *Data acquisition software, Multi-modal data acquisition, Human translation process research, User-activity data.*

1 INTRODUCTION

Human translation process research analyses the translation behaviour of translators such as properties of reading and text production rhythms, mental memory and search strategies, types of textual units that translators focus on, etc. It investigates the temporal and contextual

structure of those activities and describes inter- and intra personal variation in terms of translation competence and translation performance.

In order to acquire objective data about human translation processes, the program Translog has been designed. Translog can be used to study translation processes, hence the name Translog, but it can be equally used for other kinds of computer-based reading or writing. Since its first conception in 1995, Translog has gone through several re-implementations. Right from its beginnings, Translog had two main components, originally called *Writelog* and *Translog*, (Schou *et al.* 2010) the former component was designed for recording writing

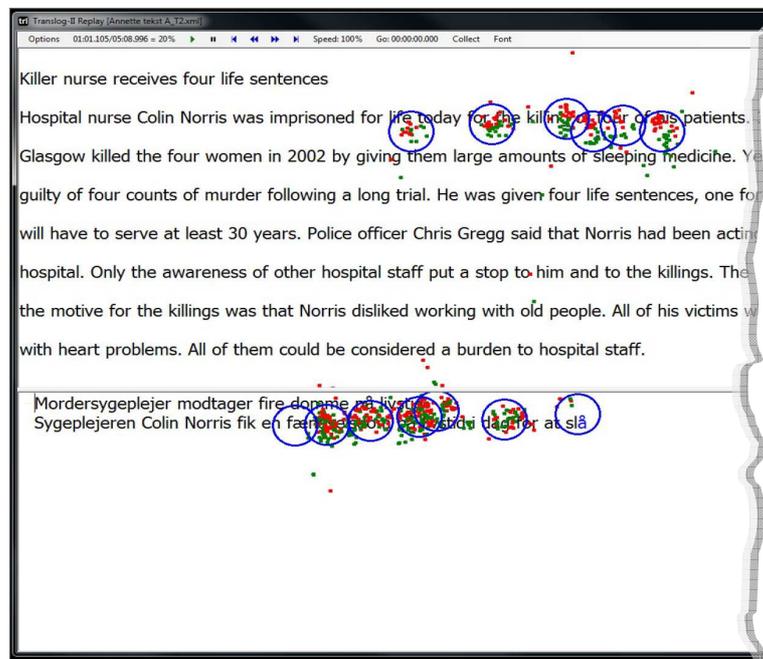


Fig. 1. The screenshot shows the source text (top) and the beginning of a translation (bottom) in the target text window. It plots the accumulated gaze movement during the translation of the 8 words: “was imprisoned for life today for the killings” into Danish, which lasted approximately 10 seconds. Red and green dots are gaze-sample points (sampling rate 60Hz.) for the right and left eye respectively, and the blue circles represent fixations. Much more gaze activity is measured on the target window during translation.

processes in time, while the latter component served for playback. These components are now referred to as the *Translog-User* and the *Translog-Supervisor* which are two interdependent programs. A major extension was introduced in the context of the EU project Eye-to-IT in 2006 when a new version *Translog-2006* could connect to eye-tracker through the GWM module (Sparkov, 2008) so as to record both, keyboard and gaze behaviour in time. *Translog-2006* was a complete re-implementation in C#, supporting Unicode and XML. However, the communication with the eye tracker through GWM¹ turned out to be too inflexible and so a further development of *Translog-II* now communicates directly with the eyetracker. This paper describes the purpose and usage of the *Translog-II* software.

Similar programs such as *ScriptLog* (www.scriptlog.net/demo.asp), and *InpuLog* (www.inputlog.net/download.html) are mainly intended for logging and analyzing writing processes, while *Translog* is specially designed for the acquisition of data for translation process research, and is widely used in the translation process research community. Schou et al (2009) count more than 80 publications making use of *Translog*, for translation process research of linguistic phenomena, (e.g. the translation of metaphors, cognates, idioms, etc.) as well as translator behaviour and cognitive processes (e.g. translator's awareness, memory constraints, (self)revision etc.), translation expertise, translation under time pressure, and machine translation post-editing. *Translog* is also used for translator training, teaching and learning purposes.

Translog-II records user activity data (UAD) all the keystrokes and gaze movements (if an eye-tracker is connected). It classifies the keystroke data as 1) insertion, 2) deletion (delete and backspace), 3) navigation (cursor movements), 4) copy/cut-and-paste, 5) return key or 6) mouse operations. Since the keylogger runs in the background, the recording does not interfere with the writing or translation process. *Translog-II* logs the exact time at which each keystroke operation is made. If connected to an eye-tracker,² *Translog-II* also records 7) gaze-sample points, 8) computes fixations (i.e. clusters of gaze-samples) and 9) mappings of fixations to the closest character on the screen. This latter operation performs a mapping from the spatial location of the gaze on the screen to a character offset in the text. That is, an X/Y

¹ These were Borland C++ implementations communicating with *Translog* through COM interface.

² Currently connection to Tobii eye tracker is supported, but other interfaces are planned.

coordinate of a fixation center is mapped onto a character position of the text that is being looked at. Since there is some noise in the recordings of gaze-sample points, the representation in the log file is such that fixations and to a certain extent also mappings can be re-computed offline. The gaze and the keystroke information can then be correlated, as they both refer to textual positions. The information is stored in an XML format and can be replayed or analyzed with Translog-II or analyzed in external tool. Carl and Müller (2011) and Carl and Jakobsen (2009) give more information on the XML representation. Here we describe the data acquisition software *Translog-II*.

2 FUNCTIONS OF TRANSLOG-II

Translog-II has three main functions:

1. Create a project file:
 - Determine the size and orientation of a source and a target window on the screen for reading and writing permission respectively.
 - Produce texts for the source and/or the target window, their layout, text font, size, color, line spacing etc.
 - Determine which data are to be logged, keyboard and eye-tracking,
2. Run and record a Translog-II session:
 - Load a project file
 - Calibrate eye-tracker (if connected)
 - Record and log UAD
3. Replay and analyze a recorded log file:
 - Statistics: figures about text production/elimination/navigation events
 - User view: replays the translation session in time (Figure 1)
 - linear view: plots a textual representation of the UAD (Figure 2)
 - Pause plot: shows a 2D representation how the text emerge in time (Figure 3)



Fig. 2. Two linear view screen shots of the same text with different temporal resolution. Top: each dot represents 1 second pause. Bottom: a dot represents 0.1 seconds between successive keyboard activities.

The Translog-II Supervisor program implements the functions 1 (create a project file) and 3 (replay a log file), Translog-II User is only used to record a Translog session and to store the UAD in a log file. A Translog-II project file can be configured for a reading experiment, where only the “source window” which contains the text to

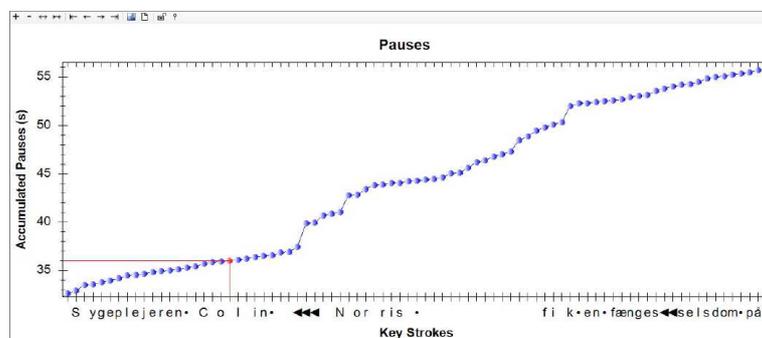


Fig. 3 Screen shot of the pause plot: Blue dots indicate the accumulation of pauses during a translation session (in seconds).

be read will be visible in the recording session, it can be configured for a writing experiment, where only the “target window” is visible, in which a text can be typed, or for a translation experiment, in which both windows are visible. In fact Translog-II also allows for post-editing texts, if a pre-defined text is entered in the target window. Translog-II allows the source and the target windows to be horizontally or vertically oriented and the source or target windows to be left or right, or bottom or top.

As in previous Translog versions, texts can be displayed in smaller portions, e.g. one sentence at a time. Each portion can be displayed for a certain number of pre-defined seconds, or the writer may decide to go on to the next portion of source text when ready to do so.

2.1 *Translog User*

The Translog-II User program is an interface for displaying and typing text and for logging UAD. To start a translation session, a project file must be loaded. According to the settings in a project file, the eye-tracker needs be calibrated, then Translog-II User opens a source and/or a target text window, plots the pre-defined texts in the source window, and waits for the translator to type a translation into the target window. As the size, orientation and rendering of the windows and the font is defined in the project file, it is not possible to re-size the windows in Translog-II User, to change the font. It is possible to use Translog-II User as a post-editing, by providing (machine) translation in the target window and to record text modifications during post-editing.

2.2 *Translog-II replay mode*

The most interesting feature in Translog is the replay mode. Translog-II Supervisor computes some statistical figures on the number of keystrokes, but more interesting are certainly the possibilities to replay the log file. As mentioned above, there are three different ways to visualize the UAD, the user view, the linear view, and the pause plot which are respectively presented in the figures 1, 2 and 3.

The user view (a screen shot is shown in Figure 1) replays the typing process in real-time, and radio buttons can be used to accelerated or decelerated, to pause the replay, rewind or forward it etc. In addition to the keystrokes, Translog-II also plots the gaze-sample points, fixations, and fixated words. In Figure 1, gaze sample points and fixations were collected over a period of approx. 30 seconds illustrating the gaze path and the coordination of reading and writing activities. It is possible to select whether gaze and fixation information should be plotted.

The linear view represents the UAD in a textual (linear) manner. Each key and mouse activity³ has a representation in the linear view, and pauses are either indicated as asterisks, and/or numeric value indicating the duration between successive activities. The granularity of the pause display can be selected starting from 1ms up to any amount of time. This gives the possibility to get an overview over the coarse temporal structure of a translation session, reducing the temporal information to a minimum (Figure 2, top), or to zoom into a sequence to study pausing behaviour as small as a few hundreds of seconds (Figure 2, bottom).

The third Translog-II replay mode is the pause plot. A pause plot represents essentially the same information as the linear view does, this time in as 2D graph. Keyboard activities are indicated on the horizontal X-axis, while the vertical Y-axis shows the accumulation of time (pauses). Figure 3 shows a segment of a translation session.

It is possible to scroll through the pause plot, to zoom in or out. Translog-II also allows to synchronize all three visualization methods. That is, all three windows (user and linear view as well as pause plot) can be opened at the same time, and by clicking the synchronization item the cursor in all three windows will be positioned at the same time. The option in the user view would then trigger a synchronous replay in the three windows.

³ It is also possible to visualize gaze-samples and fixations in the linear view, which is omitted here.

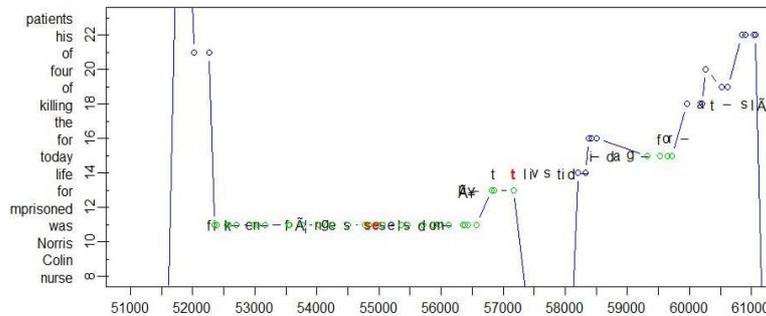


Fig. 4. The graph visualizes the translation progression of the 8 English words: “was imprisoned for life today for the killings” into Danish: “*fik en fængselsdom på livstid i dag for at slå*”. This translation segment corresponds to the accumulated gaze movements in Figure 1, which lasted approximately 10 seconds. Blue dots represent fixations on the source text, green dots fixations on the target text, the characters are insertions and the red characters are deletions.

3 TRANSLATION PROGRESSION GRAPHS AND PRODUCT DATA ALIGNMENT

While the visualization options in Translog-II (Figures 1–3) trace how the *target text* emerges in time, we have also developed more powerful visualization possibilities that show how the *translation* evolves. Figure 4 plots the relation between the word positions in the source text (vertical axis) and the translation activity on the horizontal axis in time. The figure presents a time segment of ca. 10 seconds (from secs. 51 to 61) in which the Danish words: “*fik en fængselsdom på livstid i dag for at slå*” are produced. Each keystroke is mapped onto the source text segment to the translation of which it contributes. Thus, line 11 shows all activities that relate to the production of the translation for English “was imprisoned”, line 13 for that of “for”, line 14 of “life” etc. The graph also shows gaze activities in relation to source segments.

Translation progression graphs require additional alignment knowledge of the source and the target texts, and are therefore not supported inside Translog-II. A set of additional tools are used 1. to align the translation, 2. to compute the keystroke-to-source text mapping and 3. to visualize the graph. Carl and Jakobsen (2009) describe a rule formalism and a general method to map keystrokes on

Product Data				
Source sentence word number	1	2	3	4
Source sentence	Das	ist	ein	Testsatz
Alignment			/	\
Translation	This	is	a	test sentence
Target text cursor position	1-4	6-7	9	11-23

Process Data					
time	operation	char	pos	len	Text at time T_i
T_{f-0}	translation			23	This is a test sentence
T_{f-1}	insert	t	14	22	This is a tes sentence
T_{f-2}	insert	s	13	21	This is a te sentence
T_{f-3}	insert	e	12	20	This is a t sentence
T_{f-4}	delete	r	12	21	This is a tr sentence
T_{f-5}	delete	i	13	22	This is a tri sentence
T_{f-6}	delete	a	14	23	This is a tria sentence
T_{f-7}	delete	l	15	24	This is a trial sentence

Fig. 5. The mapping algorithm traverses the process data against the time line, from the final translation product towards the start of the translation session. The arrow on the left indicates the time flow in the process data. While traversing the process data, all operations are associated with a target text position, and hence a source sentence number.

source text words. They iteratively retrieve all keystrokes which contribute to the creation of a target text segment T_i . Given alignment information for the target text segment T_i to a source segment S_j , the retrieved keystrokes can be mapped via T_i to source segment S_j . With an exhaustive fragmentation of the target text into n non-overlapping text segments $T_1 \dots T_n$, and a complete alignment of the source and the target texts, every keystroke can be associated with a source segment S_j . As illustrated in Figure 5, we have re-implemented this algorithm in a more efficient way: From the alignment of the translation product we know which source words are linked to which cursor positions in the target text. The text processing operations can then be mapped on the target text positions and further on source words. Figure 5 illustrates the correction of “trial sentence” into “test sentence” in the target text. These operations are mapped on source word 5 “Testsatz”.

Figure 5 shows that the four last letters “rial” of the word “trial” were deleted and then substituted by “est”. As all deletion and insertion activities take place between cursor positions 11 and 23, these keystrokes are part of the operations that lead to the translation of

“Testsatz”. To compute these mappings, the algorithm looks backwards into the activity data. The last operation in Figure 5 was the insertion of the letter “t” at time T_{f-1} , which took place at cursor position 14. Accordingly, the text length was accordingly one character shorter and the translation of “Testsatz” consisted only of the characters 11 to 22. Starting at T_f , the time of the final translation, the loop ends with the first operation, where all keystrokes are mapped on a source text word.

A free version of Translog-II can be downloaded for academic use upon request to the author of this paper.

ACKNOWLEDGEMENT. The implementation of Translog-II would not have been possible without previous work on the various Translog versions in the development of which were involved Lasse Schou, Arnt Lykke Jakobsen, Morten Lemvig and Jakob Elming.

References

1. Carl, M., Jakobsen, A.L.: Towards Statistical Modeling of Translators' Activity Data. In *International Journal of Speech Technology*, Volume 12, Number 4, (2009) 125–138
2. Michael Carl and Henrik Høeg Müller: CRITT NLP Resources for Translation Representation of User Activity Data in Translog-II. In *Proceedings of LTC, Poznan* (2011)
3. Lykke Jakobsen, A.: Logging target text production with Translog. In Hansen, G. (ed.), *Probing the process in translation: methods and results*, Copenhagen Studies in Language. Volume 24. Copenhagen: Samfundslitteratur. (1999) 9–20
4. Lasse Schou, Barbara Dragsted & Michael Carl: Ten years of Translog. *Copenhagen Studies in Language* (37), (2009) 37–51
5. Špakov, O.: GWM – the Gaze-to-Word Mapping Tool. (2007). Available online at: <http://www.cs.uta.fi/~oleg/gwm.html> (visited on June 21, 2011).

MICHAEL CARL
COPENHAGEN BUSINESS SCHOOL,
DENMARK
E-MAIL: <MC.ISV@CBS.DK>