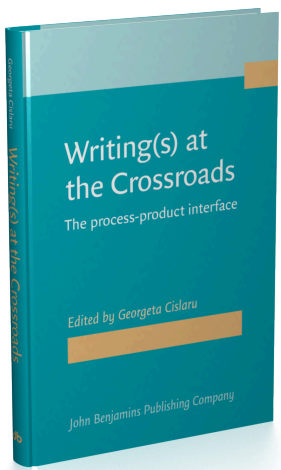
A graph theory approach to online writing data visualization

Christophe Leblay | ITEM-TERs and University of Turku, Finland
Gilles Caporossi | ITEM-TERs and HEC Montreal, Canada

https://doi.org/10.1075/z.194.09leb



Pages 171–182 of
Writing(s) at the Crossroads: The process–product interface
Edited by Georgeta Cislaru
2015. vi, 304 pp.

A graph theory approach to online writing data visualization

Christophe Leblay & Gilles Caporossi
ITEM-TERs and University of Turku, Finland / ITEM-TERs and HEC Montreal, Canada

There are currently several systems for collecting online writing data using keystroke logging. Each of these systems provides reliable and very precise data. Unfortunately, with the exception of very brief recordings, such huge amounts of data are generated that it is virtually impossible to analyze them. In this chapter, we describe a representation technique based on graph theory that allows the writing process to be understood from a fresh viewpoint. This application was originally intended to represent the data provided by ScriptLog, but the concepts can be applied in other contexts, too.

Keywords: keystroke logging; writing process; visualization; graph theory

1. Introduction

The process of writing (activity), unlike its end product (text), has two dimensions: space and time. Although there is obviously a relationship between process and product, it is not possible to reconstruct the former by analyzing the latter. It is therefore interesting and important to find ways of studying the writing activity per se, bearing in mind that word processing differs from handwriting.

Recent approaches to the study of writing based on online recordings contrast with those based on the analysis of paper versions, in that the latter focus on the page space and the former on the temporal dimension.

Writing models based on online recordings were first developed in the 1980s, in the wake of Matsuhashi (1987)'s pioneering work. Adopting a bipolar division, Matsuhashi suggested distinguishing between the conceptual level (semantics, grammar and spelling) and the sequential plan (planning and phrasing).

Virtually all the research that followed concerned the software used for recording, with Ahlsén and Strömqvist (1999), Wengelin (2006), and Doquet and Leblay (2014) focusing on ScriptLog software applications, Sullivan and Lindgren (2006) on JEdit applications, Van Waes and Schellens (2003) and Van Waes and

DOI 10.1075/z.194.09leb

Leijten (2006) on Inputlog software, Jakobsen (1999, 2006) on TransLog software, and Chesnet and Alamargot (2005) on Eye and Pen software.

While not denying all the work done on text revisions treated as a product (final text), these recent approaches all indicate that writing is primarily a temporal activity. The multitude of software approaches developed for the online recording of the writing activity reflects a clear interest in the study of the writing process. Be it from a cognitive psychology or a didactic point of view, analyzing the writing activity as a process is very important for researchers. However, although the nature of the recording depends on the type of software used, the resulting *logfiles* all share similar characteristics. In particular, they are all exhaustive, and the large amounts of data they contain are difficult for human researchers to analyze. Depending on the point of view and the topic of the research, the data therefore have to be processed and filtered. They need to be converted into a more legible format for the researcher. They may become less exhaustive in the process, but we are left with the information that is most relevant to the analysis.

An important topic in raw data transformation is visualization. Visualization allows researchers to form intuitions and develop a clearer understanding of the underlying process of creation. As the goal of the representation changes with the research topic, we cannot expect a single representation to be adapted to every possible study.

Although the technical format of the data depends on which software is used, all systems (except in the case of handwriting) record the same elementary events (i.e. keyboard keystrokes or mouse clicks). An example from ScriptLog is shown in Figure 1.

time	type	from	to	key
0.00	10	1	0	<START>
4.21	7	0	0	L
4.46	7	1	1	I
4.75	7	2	2	E
5.05	7	3	3	U
5.26	7	4	4	
5.70	7	5	5	I
5.86	7	6	6	D
8.08	7	7	7	…
8.36	7	8	8	A
8.53	7	9	9	L
8.81	7	10	10	
12.28	5	11	11	<DELETE>
12.45	5	10	10	<DELETE>
12.61	5	9	9	<DELETE>
12.78	5	8	8	<DELETE>

Figure 1. Excerpt from a logfile produced by ScriptLog

Here, each line is associated with an elementary event. Technically speaking, this record represents the whole of the writing process, but without any prepro-

cessing it is extremely unwieldy, and this preprocessing will differ according to the task being carried out by the researcher.

For instance, studying the length and location of pauses does not require the same level of information as studying the text revision process. In both cases, the same basic information is used, but the researchers have to apply the aggregation to a different level, according to their needs. Data preprocessing is sometimes unavoidable, and given the large amount of data produced by the system (a logfile corresponding to a 15-minute recording may contain up to 2,000 lines), this preprocessing should ideally be automated, in order to avoid errors.

2. Visualization techniques

Although visualization is an important part of writing process analysis, only a few visualization techniques currently exist.

One of these techniques is so-called *linear representation*, in which every character that is written is displayed. If a portion is deleted, for instance, it is crossed out instead of being removed, in order to show the process of text production and not just the final product. Cursor movements using arrows or the mouse are also identified, so that it is possible to track the text construction process. This kind of representation emphasizes the spatial dimension, and the temporal dimension is difficult to follow (e.g. the user has to interpret numbers indicating cursor movements, which is not very convenient). An example of linear representation is given in Figure 2. This type of representation has the advantage of displaying the text, but it can be difficult to decipher when the process of creation is particularly complex.

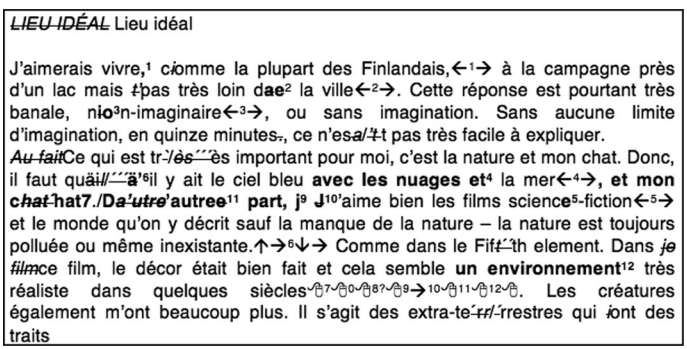


Figure 2. Linear representation of a short, 15-minute text (Leblay 2009)

Another way of visualizing the creative process is to focus on just a few values representing the text produced so far. The *Fil de la plume* graph (Chenouf et al. 1996) displays the position of the cursor, as well as the total length of the text, as

a function of time, and indicates zones where text that has already been written is modified by the writer, as we can see in Figure 3. This type of representation, which relies on a geographic information system, is known as *GIS representation*, and is used in various software, such as Inputlog (Van Waes & Leijten 2006).

Although this approach represents both spatial and temporal dimensions, a major weakness of GIS representation is that the position of visible text may cease to be correct when insertions or deletions occur upstream. If the position of a character is altered, it becomes difficult to figure out which parts of the text are involved in any subsequent modifications. Another drawback is that users cannot tell which points in the graph correspond to which places in the text.

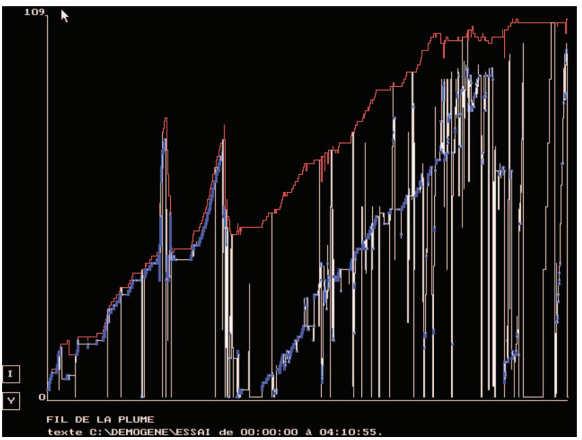


Figure 3. *Fil de la plume* GIS representation (Chenouf 1996)

3. An alternative: The graph representation

In this section, we describe a new representation technique (Leblay & Caporossi 2014) that allows for the visual identification of basic operations such as insertion and deletion, but also makes it possible to isolate portions of the document according to the processing activity performed by the writer. Each time this new representation technique has been presented to psychologists or linguists, it has been given a very positive reception.

To sidestep the problem of the written text changing position as a result of revisions, we have come up with a slightly different approach whereby the characters are given a relative, rather than an absolute position, when they are first written. This solution has proved to be more suitable for representing the dynamic aspect of the writing activity. Keystroke sequences are merged to form a textual entity that is represented by a node in the graph. If two nodes interact, either chronologically or spatially, they are connected by an edge (or link) showing this relation.

Graphs are mathematical tools based on nodes or vertices that can be connected by links or edges. Application fields may be more or less concerned by graph theory. For instance, some graph-theoretical results directly apply to chemistry (Caporossi et al. 1999a and 1999b). For other fields, such as transportation, scheduling and communication, the algorithms underlying graph theory and networks may be more relevant. Since the 1990s, graphs in the shape of concepts maps have also been used for representational purposes in the human sciences (Novak 1990). Here, we describe how graph representation can be used to visualize data pertaining to the writing process. Examples of graph representations of the writing process are provided in Figures 4 (novice) and 5 (expert).

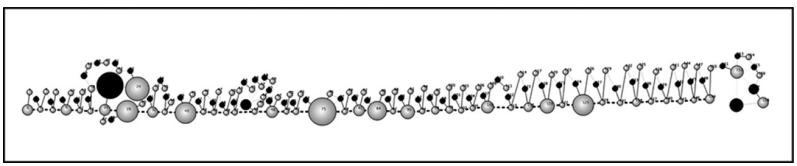


Figure 4. Graph visualization: an example of a novice writer (global view)

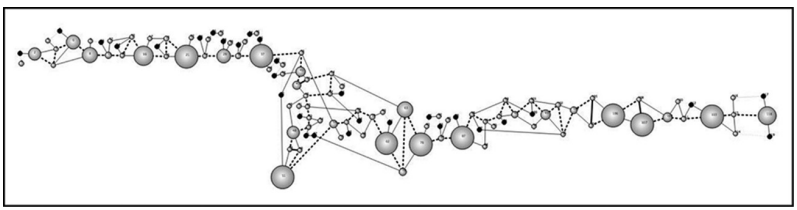


Figure 5. Graph visualization: an example of an expert writer (global view)

The size of a vertex depends on the number of elementary operations it represents. In the case of the novice writer, we can see that there are only a few large nodes, indicating a higher frequency of errors or typos. We could have displayed the section of text corresponding to each node, thus providing a virtually linear representation, but decided not to here, in order to keep the graphs as simple as possible.

The structure of the graph is also very informative: it is almost linear for the novice, whereas for the expert, the central portion is much more complicated. This complex portion, in the middle of the graph Figure 5, represents a section of the text that was rewritten and changed at a higher level and clearly not from just the lexical point of view. Analysis of that portion of the graph reveals that the author modified the text in four successive passes.

There are three different types of key or mouse events: (i) additions or insertions of characters or spaces; (ii) deletions of characters or spaces; and (iii) cursor moves using arrows or the mouse. Spatially and temporally contiguous sequences are merged and represented by the nodes.

3.1 Nodes

The size of a node reflects the number of elementary events it represents, and its color the nature of these events. A light color represents an addition, and black a deletion. The nodes are numbered according to their order of creation.

3.2 Links

The nodes are connected by links (or edges) representing spatial or temporal relations. The nature and width of these edges indicate the type of relation. A solid line represents a chronological link (solid lines link Node 0 to the last node, running through all the nodes in chronological order). All the other links between the nodes correspond to spatial relations. The link between an addition node and its deletion counterpart is narrow, and the spatial link between nodes that form part of the final text is broad. The content of the nodes with broad links therefore corresponds to the final version of the whole text (these broad links form a path representing the final text; the spatial dimension).

3.3 Analysis of graphic patterns

These graphs can be analyzed in a number of different ways, and here we concentrate on the most useful ones. We begin by identifying patterns that correspond to some of the classic operations in the writing process. From a technical point of view, these operations correspond to easily recognizable subgraphs. It is useful to be able to identify these subgraphs, in order to analyze the graph as a representation of the writing process.

3.3.1 *Additions and insertions*

Text can be added in three ways, two of which are very similar:

i. Adding text to the end of the node that is currently being written is not represented by any particular pattern, but the node increases in size;
ii. Inserting text in the node that is currently being written (i.e. not at the end) causes the node to split, as illustrated in Figure 6;

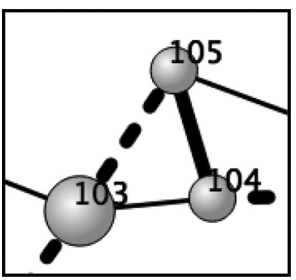


Figure 6. Insertion in the current node: Nodes 103 and 104 were merged until Node 105 was inserted

iii. Inserting text in a node that has already been written causes the node to split, and the resulting configuration is shown in Figure 7.

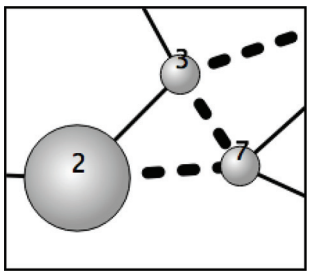


Figure 7. Insertion: Nodes 2 and 3 were merged until Node 7 was inserted

From the graphic and linguistic standpoints, *insertions* correspond to internal additions (ii and iii) while *addition* corresponds to an extension of the text (i).

3.3.2 *Deletions*

Like additions, deletions result in different subgraphs, depending on whether they erase the end of the most recent node, an internal part of the most recent node or an internal part of a node that has already been written:

i. The case of an immediate deletion (e.g. after a typing error) is shown in Figure 8. We can see that this was the most frequent operation for the *novice* (Figure 4), meaning that the text was scarcely modified once it had been written;

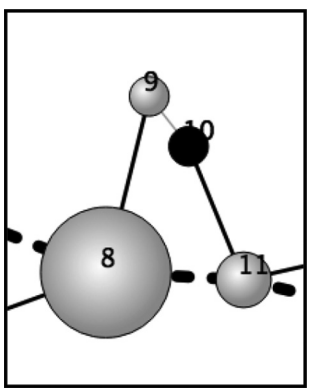


Figure 8. Immediate deletion: Nodes 8 and 9 were merged until the portion of text corresponding to Node 9 was deleted (Node 10)

ii. A deletion in the most recent node (not at the end) is shown in Figure 9;

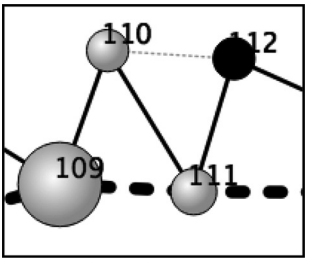


Figure 9. Deletion in the most recent node (but not at the end): Nodes 109, 110 and 111 were merged until the portion of text represented by Node 110 was deleted (Node 112)

iii. Delayed deletion results in the subgraph shown in Figure 10.

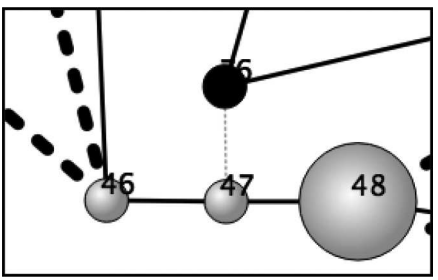


Figure 10. Deletion: Nodes 46, 47 and 48 were merged until the portion of text represented by Node 47 was deleted (Node 76)

3.3.3 *Substitutions*

Although more complex operations can be regarded as sequences of the simple operations described above, they nevertheless give rise to particular subgraphs that are easily recognized. For instance, replacement can be viewed as deletion immediately followed by insertion at the same place. Figure 11 represents the subgraph corresponding to a replacement in a node that has already been written. We can see that substitutions usually occur when writers are correcting their text, this activity usually being associated with *expert* writers rather than *novices*.

4. Summary and future research directions

In this chapter, we described a new technique for representing written language production that offers a solution to the problem of shifting text positions. This technique allows the researcher to easily identify each portion of the document that is modified by the writer. According to linguistics researchers working on the writing process, this type of representation is easier to understand than those that were

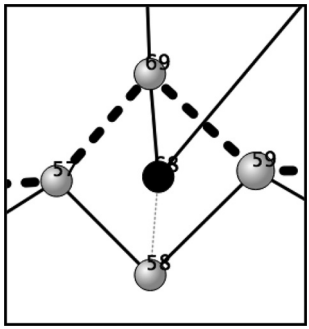


Figure 11. Substitution: Nodes 57, 58 and 59 were merged until the portion of text corresponding to Node 58 was deleted (Node 68) and replaced by an insertion (Node 69)

previously available. One major advantage is the ability to visualize modification patterns from both a spatial and a temporal point of view in the same representation. Intuition also seems to be stimulated more by a graph representation than it is by linear or GIS representations.

Several important aspects of graph representation in writing require further investigation, especially the temporal dimension. For instance, nodes corresponding to long pauses (the definition of the minimum duration of a pause being defined by the user) could be inserted, or the time and duration corresponding to each node could be indicated.

It is also important to distinguish between the various levels of text improvement as defined by Faigley & Witte (1981), by distinguishing surface modifications (correction of typos, orthographic adjustments, etc.) from text-based modifications (reformulation, syntactic modifications, etc.). A first step in this direction would be to highlight nodes containing more than one word, possibly with a space before and after visible characters. A second step would be to better define the nature of the transformation represented by a given node. This would require computational linguistics tools.

The graph is currently drawn by hand. We therefore need to devise an algorithm that can automatically place vertices in such a way that (i) patterns are easy to recognize and (ii) the spatial aspect is preserved as much as possible, so that the writing process remains easy to follow.

References

Ahlsén, Elisabeth, and Sven Strömqvist. 1999. "ScriptLog: A Tool for Logging the Writing Process and its Possible Diagnostic Use." In *Augmentative and Alternative Communication: New Directions in Research and Practice*, ed. by Filip Loncke, John Clibbens, Helen Arvidson, and Lyle Lloyd, 144–149. London: Whurr Publishers.

Caporossi, Gilles, Dragos Cvetkovic, Ivan Gutman and Pierre Hansen. 1999a. "Variable Neighborhood Search for Extremal Graphs 2. Finding Graphs with Extremal Energy." *Journal of Chemical Information and Computer Sciences* 39: 984–996. DOI: 10.1021/ci9801419

Caporossi, Gilles, Ivan Gutman, and Pierre Hansen. 1999b. "Variable Neighborhood Search for Extremal Graphs 4. Chemical Trees with Extremal Connectivity Index." *Computers and Chemistry* 23(5): 469–477. DOI: 10.1016/s0097-8485(99)00031–5

Chenouf, Yvanne, Jean Foucambert, and Michel Violet. 1996. *Genèse du texte*, vol. 2 [INRP research report].

Chesnet, David, and Denis Alamargot. 2005. "Analyse en temps réel des activités oculaires et grapho-motrices du scripteur: intérêts du dispositif Eye and Pen." *L'année psychologique* 105(3): 477–520. DOI: 10.3406/psy.2005.29706

Doquet, Claire, and Christophe Leblay. 2014. "Temporalité de l'écriture et génétique textuelle: vers un nouveau langage?" In *Actes du Congrès mondial de linguistique française vol. 8*, ed. by Franck Neveu, Peter Blumenthal, Linda Hriba, Annette Gerstenberg, Judith Meinschaefer, and Sophie Prévost, 2767–2781. Berlin. ⟨http://www.shs-conferences.org/articles/shsconf/abs/2014/05/shsconf_cmlf14_01204/shsconf_cmlf14_01204.html⟩ DOI: 10.1051/shsconf/20140801204

Faigley, Lester, and Stephen Witte. 1981. "Analysing Revision." *College Composition and Communication* 32: 400–414. DOI: 10.2307/356602

Jakobsen, Amt Lykke. 1999. "Logging Target Text Production with Translog." In *Probing the Process in Translation. Methods and Results*, ed. by Gyde Hansen, 9–20. Copenhagen: Samfundslitteratur.

Jakobsen, Amt Lykke. 2006. "Research Methods in Translation: Translog." In *Computer Key-Stroke Logging and Writing: Methods and Applications*, ed. by Kirk Sullivan and Eva Lindgren, 95–105. Amsterdam: Elsevier.

Leblay, Christophe, and Gilles Caporossi. 2014. *Temps de l'écriture. Enregistrements et représentations*. Louvain-La-Neuve: Academia-L'Harmattan (coll. Sciences du langage: Carrefours et points de vue). DOI: 10.4000/questionsdecommunication.6988

Leblay, Christophe. 2009. "Les invariants processuels. En deça du bien et du mal écrire." *Pratiques* 143/144: 153–167. DOI: 10.4000/pratiques.1430

Leijten, Mariëlle, and Luuk Van Waes. 2005. "Writing with Speech Recognition: The Adaptation Process of Professional Writers." *Interacting with Computers* 17(6): 736–772. DOI: 10.1016/j.intcom.2005.01.005

Lindgren, Eva, Kirk Sullivan, Urban Lindgren, and Kristyan Spelman Miller. 2007. "GIS for Writing: Applying Geographic Information System Techniques to Data-Mine Writing's Cognitive Processes." In *Writing and Cognition: Research and Applications*, ed. by Gert Rijlaarsdam (series ed.) and Mark Torrance, Luuk Van Waes and David Galbraith (vol. eds.), 83–96. Amsterdam: Elsevier.

Matsuhashi, Ann. 1987. "Revising the Plan and Altering the Text." In *Writing in Real Time*, ed. by Ann Matsuhashi, 197–223. Norwood, NJ: Ablex Publishing Corporation.

Novak, Joseph D. 1990. "Concept Maps and Vee Diagrams: Two Metacognitive Tools for Science and Mathematics Education." *Instructional Science* 19: 29–52. DOI: 10.1007/bf00377984

Strömqvist, Sven, and Henrik Karlsson. 2002. *ScriptLog for Windows – User's manual*. Technical Report. University of Lund: Department of Linguistics and University College of Stavanger, Centre for Reading Research.

Sullivan, Kirk, and Eva Lindgren (Eds.). 2006. *Computer Keystroke Logging and Writing. Methods and Applications*. Amsterdam: Elsevier.

Van Waes, Luuk, and Peter Jan Schellens. 2003. "Writing profiles: The effect of the writing mode on pausing and revision patterns of experienced writers." *Journal of Pragmatics* 35 (6): 829–853. DOI: 10.1016/s0378-2166(02)00121–2

Van Waes, Luuk, and Mariëlle Leijten. 2006. "Inputlog: New Perspectives on the Logging of On-Line Writing Processes in a Windows Environment." In *Computer Key-Stroke Logging and Writing: Methods and Applications*, ed. by Kirk Sullivan, and Eva Lindgren, 73–93. Amsterdam: Elsevier.

Wengelin, Åsa. 2006. "Examining Pauses in Writing: Theories, Methods and Empirical Data." In *Computer Key-Stroke Logging and Writing: Methods and Applications*, ed. by Kirk Sullivan, and Eva Lindgren, 107–130. Amsterdam: Elsevier.