

STUDYING WRITERS' REVISING PATTERNS WITH S-NOTATION ANALYSIS

PY KOLLBERG & KERSTIN SEVERINSON EKLUNDH

Royal Institute of Technology (Sweden)

Abstract. This chapter describes a computer-based technique for analysing revision and text production strategies, called S-notation. On the basis of a keystroke record, a representation is created of all revisions made to a text during a writing session, including their order and internal structure. The resulting representation is embedded in an interactive program, which enables both qualitative and quantitative analyses of revisions made during a writing session. The tools described also support automatic identification of connected episodes of revision, which are relevant for the study of individual and task-specific writing strategies. We present the methodological and theoretical rationale behind the development of the method, and the basic principles for the creation of the S-notation. Subsequently, we describe how the tools can be used to answer particular research questions, including a discussion of the limitations of a purely formal treatment of revision. A multi-level framework is outlined for how to gain knowledge of revision processes using different kinds of data. Finally, we summarize studies made by ourselves as well as other researchers using S-notation analysis.

Keywords: revision, writing, revision trace, revision episode, logging, keystroke methods.

1. INTRODUCTION

The use of computers for writing has profoundly changed the study of revision processes. At present, virtually all writers use a computer, which has opened up new methods of observing and analysing the writer's actions during composing. This is also an important step in order to understand underlying composing processes.

Within the cognitive tradition of writing research, revision is seen as the result of a complex reviewing process, involving the detection and possible diagnosis of a problem in the text (Flower, Hayes, Carey, Schriver, & Stratman, 1986; Flower & Hayes, 1981). By reading and evaluating what has been written, the writer forms a representation of the text, and compares it with his or her intentions. In this process, a problem may be discovered (a *detection*). If the writer can recognize the problem (forming a *diagnosis*), this may lead to finding a strategy for changing the text (i.e., making a *revision*), and thus eliminating the problem. However, this process may be difficult for many reasons, and many writers fail to reach the level of a successful revision. For

example, writers may not have an adequate representation of their texts, or of their own intentions. Moreover, even if they do detect a problem, it is possible that they do not succeed in diagnosing it. Therefore, the strategy decided upon may be to *rewrite* the whole or part of the text rather than to *revise* it. Flower *et al.* (1986) presented this model of revision as a basis for an understanding of writers' cognitive difficulties in revision, and discussed some of its pedagogical implications.

This revision process often starts with the writer's reading the text in order to evaluate it, and it may or may not eventually lead to an actual change in the text. In principle, revision may occur at any time in the writing process, reflecting the recursive nature of writing and the gradual development of the writer's plans. Revisions may affect the text at higher or lower levels, or they may only change the writer's mental representation of the task.

Methodologically, the study of revision involves many challenges. An obvious problem in studying revision is that there is a gap between the cognitive sides of it and the actual changes that the writer makes in the text. In general, it is not possible to relate a change in the text to a certain goal on the part of the writer, without having additional information about the writer's thinking processes. However, the need for a clear distinction between what the writer does and his/her underlying intentions is often ignored, and researchers tend to classify revisions in terms of their assumed purpose rather than just their form.

In this chapter we present a general computer-based method, S-notation¹, for representing and analysing writers' revisions to a text. The method simplifies the study of revision as an online process, by making it possible to follow in detail the continuous shaping and re-shaping of the written text. In contrast to other methods, it gives a complete history of the changes that occur in a text during a writing session, including their order and structure. This representation can be used in combination with other data to provide an understanding of revision at both high and low levels. As we will show, such a computer-based analysis emphasizes the necessity of separating the physical and the cognitive levels of revision. We will outline a multi-level framework, illustrating how different data can be used to reach a successively more advanced understanding of revision processes.

2. THEORETICAL AND METHODOLOGICAL RATIONALES FOR THE S-NOTATION

The work with developing the S-notation grew out of research needs that arose when using keystroke analysis to study revision (Severinson Eklundh, 1990; Severinson Eklundh & Kollberg, 1996b). Keystroke logging is a straightforward way of tracking writers' actions, but when trying to make sense of logging data, the researcher often confronts a number of problems. The records usually represent a writing session as a sequence of keystroke events, including typed characters, pauses, and commands to

¹ For convenience, we use the word 'S-notation' both for the notation (representation) of the revisions made to a text, and for the method for generating this representation from a keystroke record. The name 'S-notation' was originally chosen because of the first letter S of the Swedish word for 'write' (*skriva*). An alternative and simpler term might be just 'revision trace'.

the computer. Because of the variation in the functions and interface of word processors, the language of the keystroke records varies considerably from system to system. And since the actions are represented in a chronological order, it is often quite difficult to identify where the changes made by the writer are located, and how they affect the text. As a result, the researcher must often create meta-level protocols to account for what is really happening in the keystroke session.

The S-notation is an automatic method to generate such a meta-level revision protocol from keystroke data. In contrast to ordinary keystroke logfiles, this notation represents revisions at their place in the text, so that the researcher can directly see the surrounding context including text and other revisions. It also encompasses information about the range, order and structure of revisions, allowing for analyses of more complex patterns of revising activity. The method is intended to be used as an instrument in the study of revision, being complemented by contextual data about the writing session to inform a particular research question.

The design of the S-notation was inspired by a manual notation for revision invented by Matsuhashi (1987). Her concern was the analysis of writing as an ongoing cognitive process, governed at the point of inscription by a set of constraints imposed by characteristics of the writing task. The notation used was applied to handwritten data, and was only partly formalised. However, the application of this framework to a qualitative analysis of a teenage writer's composing sessions embodied some fundamental conceptual distinctions that we found to be useful in the treatment of keystroke data. Matsuhashi's notation transcribed the successive changes made by the writer, the place in the text where a revision was made, and their order. Importantly, this allowed for representation not only of when and how a revision is made, but also its relationship to preceding and following actions.

We developed the S-notation as a formally specified, computer-based representation of revisions, based on some of the principles from Matsuhashi. The following goals were formulated, and eventually realized in the implementation of the method:

- 1) The S-notation should represent a writing session as a sequence of revisions made to the text – rather than as operations performed in the word processor as in keystroke logfiles – making it possible to investigate revisions in detail.
- 2) It should be possible for a computer program to generate the S-notation from a keystroke record.
- 3) The S-notation should be independent of the word processor used during the writing session. Consequently, it should not contain commands made to a computer, but only their effects on the text.
- 4) It should be possible for a computer to replay a writing session revision by revision, forward or backwards, using only the S-notation as an input.
- 5) The S-notation should be readable for the user.

Below we summarize the most important principles of the S-notation. For a more detailed account, including the formal rules for the notation and their application, see Kollberg (1998). Aspects of the use of S-notation to analyse online revision processes, and especially the problems of interpreting writers' actions at the computer have been more thoroughly discussed in Severinson Eklundh and Kollberg (1996a).

3. BASIC PRINCIPLES OF THE S-NOTATION

3.1. Definitions and transcription conventions

In S-notation, a revision is either an *insertion* or a *deletion*. Each revision is associated with an *interruption* (or a break in text production) at the position of the last action preceding the revision. The interruptions are numbered according to the sequence of their occurrence in the writing session.

The S-notation uses the following symbols²:

I	The interruption with sequential number # i
{inserted text} ⁱ	An insertion following interruption # i
[deleted text] ⁱ	A deletion following interruption # i

The following is an example of a short writing session transcribed in S-notation.

Now I am writing a {short }¹text.|₁ I will[probably]²|₃ change it[somewhat]³
later.|₂ Now I am finished.

The text produced in this example reads as follows:

Now I am writing a short text. I will change it later. Now I am finished.

The above transcription corresponds to the following sequence of actions by the writer:

- 1) Types *Now I am writing a text*.
- 2) Inserts *short* between *a* and *text*.
- 3) Resumes writing at the end of the text: *I will probably change it somewhat later*.
- 4) Deletes *probably*.
- 5) Deletes *somewhat*.
- 6) Resumes writing at the end of the text: *Now I am finished*.

An important requirement on a formal revision notation is that it should be neutral with respect to the writers' intentions, and thus not make any assumptions as to the purpose of a certain change in the text. This was actually a problem with Matsuhashi's notation that has also been present in many previous accounts of revision. For example, if a writer deletes a word, and subsequently inserts another word at the same position in the text, one cannot deduce that the writer intended the second word to replace the first (even if this is often the case). The idea of a computer-generated record requires a strictly formal analysis, only taking account of the overt actions made. As a consequence, S-notation only represents what we call *elementary revisions* (deletions and insertions) made in the word processor, yielding a strictly syntactic repre-

² When needed for the sake of clarity, the index is shown both at the beginning and at the end of a revision.

sensation of the changes made to the text. The deeper, semantic analysis of revisions is left to an interpretation stage, being performed on top of, and with the aid of the S-notation. Depending on the research question, this process may also require access to other data than keystroke logfiles, such as verbal reports. In Severinson Eklundh & Kollberg (1996a), a range of methodological problems in using the S-notation to study revision is discussed.

3.2. Procedure for creating a revision record

The S-notation is generated from a keystroke logfile, by first transforming the logfile to an intermediate format (MID-format). The MID-file consists of a list of elementary operations (moves, insertions and deletions) in their order of occurrence during the writing session. This format is independent of the word processor used, and could in principle be generated by any word processor. Thus, it would be possible to generate a MID file directly rather than using it as an intermediate format.

The following example, which corresponds to the short writing session above, demonstrates the simple syntax of the MID format:

```
1.1 I 23 4.8 Now I am writing a text
8.4 I 1 0.0.
3.8 M -5
1.1 I 6 1.2 short
3.1 M 5
1.7 I 41 8.9 I will probably change it somewhat later
11.0 I 1 0.0.
4.2 M -35
3.4 D 9
1.8 M 10
1.3 D 9
1.1 M 7
0.9 I 19 4.3 Now I am finished.
```

Each line in the MID file contains one elementary operation, preceded by the time elapsed since the previous operation, and followed by the number of characters affected by the operation. For movements, the direction of the movement is indicated in the positive or negative sign before the number of characters. For insertions, the text of the insertion also follows. A more detailed explanation can be found in Kollberg (1998).

The build-up of the MID representation was designed for future inclusion of pauses into the S-notation. Therefore, time information is included, but is presently not used in the interactive program (Trace-it; see below) used for S-notation analysis. This means that in order to study how pauses and other temporal aspects of a writing session relate to revising, one must have access to the original keystroke logfile. This is also necessary to get a more complete account of the writer's activity, including problems in the handling of the word processor.

Contemporary Tools and Techniques for Studying
Writing

Olive, T.; Levy, C.M. (Eds.)

2002, IX, 163 p., Hardcover

ISBN: 978-1-4020-0035-5