

Writing and Speech Recognition

Observing Error Correction Strategies
of Professional Writers

Published by
LOT
Janskerkhof 13
3512 BL Utrecht
The Netherlands

phone: + 31 30 253 60 06
fax: + 31 30 253 60 00
e-mail: lot@let.uu.nl
<http://www.lotschool.nl>

Illustrations: Mariëlle Leijten & Yvonne Wanders

ISBN 978-90-78328-31-5
NUR 616

Copyright © 2007: Mariëlle Leijten. All rights reserved.

Writing and Speech Recognition

Observing Error Correction Strategies of Professional Writers

Schrijven en spraaktechnologie
correctiestrategieën van professionele schrijvers

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van
rector magnificus prof. dr. W.H. Gispen, ingevolge het besluit van het college voor
promoties in het openbaar te verdediging op vrijdag 22 juni 2007
des middags te 4.15 uur

door

Maria Adriana Johanna Catharina Leijten

geboren op 24 mei 1975 te Raamsdonk

Promotor: Prof. dr. mr. P.J. van den Hoven

Co-promotoren: Dr. D.M.L. Janssen
Prof. dr. L.L.M. Van Waes

Voorwoord

Als het enkel aan mij gelegen zou hebben, zou dit voorwoord er nu niet zijn. De grote ambitie van een proefschrift heb ik nooit gehad. Onderzoek doen, dat was wat ik wilde. Enige voorzichtige dwang van mijn begeleiders om het toch tot een proefschrift te laten komen, heeft er waarschijnlijk voor gezorgd dat ik nu deze woorden toch op papier zet. Het volledige proefschrift is in de we-vorm geschreven. Onderzoek is namelijk iets dat je niet alleen doet. In mijn ogen is dit proefschrift – minstens – het werk van een vrouw en drie mannen. In de eerste plaats wil ik graag mijn drie mannen bedanken: Luuk Van Waes (dagelijks begeleider en co-promotor), Daniel Janssen (co-promotor) en Paul van den Hoven (promotor).

Luuk dank ik in de eerste plaats omdat hij me heeft meegenomen naar Antwerpen, zodat ik kon ontdekken wat ik graag doe. Zonder hem voelde ik me nu waarschijnlijk noch half Vlaming, noch onderzoeker. Onze gezamenlijke liefde voor onderzoek heeft een bijzondere band geschept. Luuk is de meest kritische man die ik ken. Zelfs als hij zegt dat het goed is, kun je er zeker van zijn dat er nog heel wat werk te verzetten is. Maar wie zelf kritisch is, mag een kritische houding verwachten. Dit leidde vaak tot allerlei uiteenlopende discussies, maar ook tot – in onze ogen – verbeteringen van wat er al lag. Hoezeer ik zijn kritische houding en onaflatende steun waardeer, hoop ik vooral in de toekomst te kunnen laten blijken. [Thank you for being a special friend and co-research-addict.]

Daniël dank ik voor de prettige samenwerking. Daniël nam me altijd in bescherming om zo goed mogelijk aan mijn onderzoek te kunnen werken. Hij was kritisch op andere terreinen dan Luuk en daardoor ook prettig complementair. Daniël ben ik in de loop van de periode steeds meer gaan waarderen als collega en als mens. Van ‘de schrijver van Zakelijke Communicatie’ is hij veranderd in ‘de Hasselblad-specialist’. Voor beiden heb ik veel bewondering. [Thank you for taking good care of me].

Paul is een promotor geweest die met een scherpe helicopterblik de voortgang in het oog heeft gehouden. Waar ik te scherp wilde afwijken van wat verstandig zou zijn, liet hij me zelf ontdekken dat ik beter een andere pad kon kiezen. Op die manier had ik het gevoel dat ik heerlijk mijn zin mocht doen. Bedankt daarvoor. [Thank you for the guidance and freedom].

Zonder proefpersonen was er geen proefschrift. Vandaar dat mijn dank uitgaat naar de tien advocaten en de tien academici die voor het eerste onderzoek hebben leren werken met spraakherkenning. Ook de vele studenten hebben zeer correct meegewerkt aan het tweede experiment. Ten slotte hebben Bernard Dewulf en Erik Desnerck ervoor gezorgd dat de medewerkers van de afdeling Ondertitelen van de VRT hun medewerking hebben verleend aan het laatste onderzoek. Ook zelf hebben ze zich niet onbetuigd gelaten. Hartelijk dank.

Een woord van dank gaat ook naar de verschillende collega's. Met de huidige collega's van onze sectie – en vaste waarden – Liesbeth Opdenacker en Suzy Stals, hebben we eens bekeken hoe verschillend ‘verschillend’ is. Alle typen persoonlijkheden blijken vertegenwoordigd in onze

groep, wat vaak zorgt voor een goede aanvulling. Voorbeelden hiervan zijn de samenwerking bij de organisatie van het Viot congres in 2002 en later het Sig Writing-congres in 2006. Voor mij hoogtepunten in mijn prille academische loopbaan. Liesbeth, mijn Vlaamse paranimf, bedank ik voor haar aanhoudende extraverte vrolijkheid en zeer diplomatische eigenschappen; Suzy voor haar streven naar positieve feedback en gevoel voor 'zen'. Op deze plaats dank ik ook graag Luuk voor de vele kopjes thee met bijbehorende babbel, waarbij alle grote en kleine ongemakken van het leven opgelost werden. (En als dat niet onmiddellijk kon, dan toch wel binnen afzienbare tijd.)

Ik heb veel plezier beleefd met ook weer zeer verschillende collega's uit het binnen- en buitenland. Binnenland is voor mij ondertussen België en Nederland geworden. Met naam wil ik noemen Geert Jacobs en Tom Van Hout. Thank you very much for proofreading this dissertation. (However, all remaining mistakes are mine.) Grote dank ook aan 'de jongens' van Inputlog (namens velen, vermoed ik). Sven De Mayer, bedank ik voor de prettige, geduldige en begrijpelijke introductie in de wonderde wereld van multilevel analyse. Ten slotte dank aan mijn Nederlandse paranimf Daphne Van Weijen, een onderzoekster die ook echt de lol van onderzoek inziet.

I really enjoyed cooperating with various colleagues from abroad: varying from one-day discussions to more than half a year of cooperation. Thank you to Sarah Ransdell, Ronald Kellogg and Kris Spellman for our fruitful discussions. Thomas Quinlan, who was a research fellow at the University of Antwerp, provided me with a subtle different insight in the psychology of writing, together with Maaïke Loncke. A special word of appreciation goes to the sig writing community of EARLI and in particular to the keystroke logging group. Finally, I would like to say that I am very happy that Eva Lindgren is collaborating on the development of a revision module in Inputlog. All these forms of cooperation have one thing in common: the ambition to improve.

Zeker in de afrondingsfase van mijn proefschrift heb ik niet veel andere mensen gezien dan bovengenoemde collega's. Gelukkig dat ik hen niet alleen als collega's zie, maar ondertussen ook als vrienden. Eigenlijk ben ik de mensen uit de persoonlijke kring vooral dankbaar voor de manier waarop ze mij 'mijn ding' hebben laten doen. De zeer welgemeende aanmoedigingen zijn een gewaardeerde vorm van steun geweest. Mijn ouders hebben niet rechtstreeks meegeholpen aan dit proefschrift, hoewel ze dat wellicht wel hadden gewild, maar onrechtstreeks hebben ze een grote invloed gehad. Zeer concreet omdat ze zeer tegen hun aard in tijdelijk gestopt zijn met het verbouwen van ons huis. Zij zagen ook wel in dat een proefschrift opbouwen en een huis verbouwen zeer lastig samen gaan. Op een abstracter niveau hebben ze me de waarden gedrevenheid en discipline meegegeven. Eigenschappen die nu zeer van pas kwamen.

Ten slotte bedank ik vanuit de grond van mijn hart Yvonne. Zij heeft me meegenomen naar Antwerpen, zodat ik beter kon doen wat ik graag doe. Yvonne heeft de eigenschap om overal een prettige sfeer te creëren. Ze heeft ervoor gezorgd dat de laatste loodjes ook nog prettig waren. Naast haar onvoorwaardelijke steun, hebben we vooral veel – en vaak – gezellig samengewerkt. Met alle geduld van de wereld heeft ze ervoor gezorgd dat ik toch nog dat ene getalletje kon toevoegen, heeft ze verscheidene tekstversies zeer zorgvuldig gecontroleerd ... Met wat ik ook schrijf, doe ik haar tekort. Bedankt voor alle grote en kleine dingen.

Antwerpen, april 2007
Mariëlle Leijten

Table of contents

1	Introduction	
	1 Writing and speech technology as a research object	3
	2 Converging research methods	6
	3 Practical organization of this thesis	9
	Section I	
	Adaptation and writing processes of novice speech recognition users in their professional working environments	
2	How do writers adapt to speech recognition software?	
	The influence of learning styles on writing processes in speech technology environments	
	1 Introduction	14
	2 Description of the study	17
	3 Results	18
	3.1 Material	19
	3.2 Writing modes	20
	3.3 Repairs	21
	4 Conclusion	23
	5 Discussion and further research	25
	Acknowledgements	26
3	Writing with speech recognition	
	The adaptation process of professional writers with and without dictating experience	
	1 Introduction	28
	2 Related research: writing processes	29
	2.1 Speech recognition and writing	32
	2.2 Research questions	34
	3 Description of the research project	34
	3.1 Participants	35
	3.2 Design and procedure	35
	3.3 Materials	37
	3.4 Selection of data	37
	4 Analysis	38
	4.1 Categorization model	38
	4.2 Transcription model	45

5	Quantitative study: writing modes	48
6	Case study	52
6.1	Writing modes	54
6.2	Repairs	57
6.3	Revisions	58
6.4	Pauses	60
6.5	Transcriptions	61
7	Conclusions	65
8	Discussion and further research	66
	Notes	67
	Acknowledgements	67
	Appendix	69
4	Repair strategies in writing with speech recognition	
	The effect of experience with classical dictating	
1	Introduction	72
1.1	Speech recognition and writing	72
1.2	The text produced so far and writing media	73
2	Research questions	76
3	Description of the case study	76
3.1	Participants and writing tasks	76
3.2	Design and procedure	77
3.3	Categorization model: repairs	77
4	Case study	78
4.1	General	78
4.2	Distribution of the repairs over the writing process	79
4.3	Effect of repairs on modus monitoring	81
4.4	Immediate or delayed repair of errors	82
5	Conclusions and discussion	84
	Notes	85
	Acknowledgements	85
	Appendix	86



Section II Error correction strategies in isolated contexts

5	The effect of errors in the text produced so far	
	Strategy decisions based on error span, input mode, and lexicality	
1	Introduction	90
1.1	Successful error detection	92
1.2	The effect of the text produced so far on the writing process	93

1.3 Working memory and writing	94
1.4 Hypotheses	96
2 Method	99
2.1 Participants	101
2.2 Design	101
2.4 Procedure	104
2.5 Dependent variables	107
3 Analyses	109
4 Results	110
4.1 General findings	110
4.2 The effect of mode of error presentation (speech vs. non-speech)	111
4.3 The effect of error span on cognitive effort	112
4.4 The effect of input mode on cognitive effort	116
4.5 The effect of lexicality on cognitive effort	117
5 Conclusions and discussion	118
6 Further research	121
Acknowledgements	124
Appendix 1	125
Appendix 2	126

6 Isolating the effects of writing mode from error span, input mode, and lexicality when correcting text production errors: A multilevel analysis

1 Introduction	130
2 Hypothesis	132
3 Method	135
3.1 Participants	136
3.2 Materials	136
3.4 Dependent variables	137
4 Data analysis	137
4.1 Characteristics of participants and sentences	139
4.2 Models	140
5 Results	142
5.1 Effect of mode of error presentation on the interaction with correct TPSF	144
5.2 Effect of mode of error presentation on the interaction with incorrect TPSF	145
5.3 Effect of error span	148
5.4 Effect of input type	151
5.5 Effect of lexicality	152
6 Conclusions and discussion	154
Acknowledgements	158
Appendix	159



Section III

Error correction strategies of professional speech recognition users writing business texts

7	The text produced so far in business texts	
	Error correction strategies of professional speech recognition users	
1	Introduction	166
1.1	Reflection on the text produced so far in speech recognition	169
1.2	Error correction strategies in writing with speech recognition	176
1.3	Interaction with the text produced so far	177
2	Method	178
2.1	Participants	178
2.2	Task	179
2.3	Data collection	179
2.4	Procedure	184
3	Data analyses	185
3.1	Product analysis	185
3.2	Process analysis	185
3.3	Protocol analysis	188
4	Results	191
4.1	General	192
4.2	Writer groups	199
4.3	Case studies	206
4.5	Postpone (revision) profile: Ethan	212
4.6	Postpone (technical problems) profile: Ben	214
5	Conclusions and discussion	217
5.1	Writing profiles	218
5.2	Comparison with previous studies	221
5.3	Methodological considerations	222
	Acknowledgements	223



Section IV

Logging writing processes in a Windows environment

8	Inputlog	
	A research tool for observing and analyzing multimodal writing processes in a Windows environment	
1	Introduction	228
2	Characteristics of Inputlog	229
2.1	Word processor independency	230

2.2	Parsing	230
2.3	XML structure of output files	231
2.4	Speech recognition	231
3	Technical description	233
3.1	Programming language	235
3.2	Structure of Inputlog	235
3.3	Structure of IDF-files	236
3.4	Program settings	237
4	Functional description	238
4.1	Entry screen	239
4.2	Record	239
4.3	Generate	240
4.4	Integrate	243
4.5	Play	246
4.6	Help file	246
5	Applications	246
5.1	Case study: pausing and revision behavior in writing bad news letters	247
5.2	Experimental study: the text produced so far and the use of working memory	248
5.3	Applications on a broader level	250
6	Data analysis	250
6.1	Factor analysis	250
6.2	Progression analysis and GIS	253
7	Further development	254
7.1	Revision analysis	254
7.2	Subtitling via speech recognition	255
7.3	Typing tests	256
7.4	Searchable output files	256
8	Conclusion	257
	Notes	257
	Acknowledgements	258
	Appendix 1	259
	Appendix 2	260
	Summary	271
	References	281
	Samenvatting	289
	Curriculum Vitae	301

1

Introduction

It is snowing, salt from a clogged shaker
but however sparsely snow falls
it bares the skin of the world

...

[Het sneeuwt als uit een verstopt zoutvat.
Maar hoe karig sneeuw ook valt,
ze legt het vel van de wereld bloot.]¹
Bernard Dewulf

Would we not all want to write like the columnist that we admire or like the novelist whose next book we can not wait to see in print? In most cases, our writing does not even need to be best-selling prose; we would be very happy if we could write our routine business texts without too much effort. But let us face up to it: writing is difficult. In the words of the Dutch writer Kees van Kooten: “Writing is sitting put till you’ve got it down: a fair hand.”²

Ever since the origins of writing, mankind has been trying hard to make writing as easy as possible. On the one hand, products have been developed to facilitate the writing process; on the other hand, products have been developed to support the underlying subprocesses. An example of the efforts to facilitate writing can be found in the ease with which children use the computer as a writing instrument nowadays. Various tools like spelling checkers allow them to recognize and avoid spelling errors: wavy lines under incorrect words show the mistakes. This contrasts sharply with the limited and labour-intensive instruments which our forefathers had at their disposal. If we take a step back in history then we notice that before the alphabet was invented, messages were carved in figures and symbols. The ease with which we make adaptations to our text nowadays would be unthinkable if we still used the same writ-

¹ Text of columnist Bernard Dewulf entitled ‘Sneeuwvel’, *De Morgen*, February 13, 2006.

² Fragment from: Kooten, K. & E. Spieker (2003). *Letterlust*. Uitgeverij Manteau/De Harmonie Antwerpen/Amsterdam, p.120. [Original Dutch text: Schrijven is blijven zitten tot het er staat: schoonschrift]

ing instruments. The transition from pen-and-paper writing to keyboard-based word processing was equally decisive. If I had to write this text now with pen-and-paper, I would feel forced to think much better beforehand on the exact formulation of my words. In keyboard-based word processing my fingers regularly rest on the keyboard while I pre-test formulations, think and adjust; meanwhile, my eyes keep focused on the text produced so far on the screen to help me formulate my sentences. I can not even begin to imagine what strategy that I would use if I needed to carve this text in a rock. In sum, efficiency and comfort are two pillars in the evolution of written communication.

It is the quest for efficiency that has led to the development of speech recognition³. Speech is considered the most natural mode of discourse production. Compared to speaking, writing is far more difficult. Even nowadays people all over the world are not able to master this skill⁴. In other words, most people can speak properly, but not all of them can write properly. From a historical point of view written language has become necessary to store and distribute information. However, the basis is spoken language, as Crossman says:

From a Darwinian perspective, written language is a 6,000-to-10,000-year-old bridge that humanity has been using to walk from our First Golden Age of oral culture to our Second. We undertook this journey to survive as species. Six thousand to ten thousand years ago, lacking the ability to store and retrieve by memory the growing sum of survival information, our species faced two options: develop new storage-retrieval technology or self-destruct. That's when and why we created the written-language bridge. (Crossman, 2004, p. 21)

Writing can be seen as a visual representation of spoken language: a combination that speech recognition takes full advantage of. When using speech recognition writers dictate their texts to the computer, the computer transforms the spoken input into written text that appears on the screen. Perfect, one would say. Yes, but only if speech recognition is 100% accurate. In this thesis we will show that writing with speech recognition is not errorless and may even bring in a new type of error. Today's speech recognition software is not advanced enough to recognize every word that writers dictate. When the speech recognizer does not recognize a word, it chooses a similar word from its lexicon, with the highest probability rate within the written context. A typical speech recognition error is that 'editing' is recognized as 'edit in'. The difference with the familiar typing errors is that the writer cannot always control the error. Consequently, re-reading the text produced so far is turned into quite a challenge. Furthermore, since speech recognition is much faster than any other writing tool, the task of correcting errors is approached differently.

In this thesis we describe the changes in the organization of speech recognition based writing processes. In the field of writing research, speech recognition is a new

³ Dutch is one of the languages available in speech recognition because a major player in speech recognition was Lernout and Hauspie. They were situated in 'Flanders Language Valley' in the Dutch-speaking part of Belgium. The Dutch and Flemish governments have put Language and Speech Technology on the agenda and have given financial support to the 'Nederlandse Taalunie' (<http://taalunieversum.org/taalunie/>).

⁴ 17% Of the world population is illiterate: in some parts of Africa this percentage is even higher than 50.

writing instrument that may cause a shift in writing process research because the underlying processes are changing. In addition to this, we would like to take advantage of one of the weak points of speech recognition, namely the misrecognitions in the text. As it is, speech recognition visualizes how writers deal with errors in the text produced so far. In addition, speech recognition may cause a different view on writing processes because the processes are easier to reveal. Speech recognition will bare the skin of writing processes in general.

In this chapter we will describe the main topic of this thesis from two perspectives. On the one hand, we will describe the relation and the structure of the different research topics that are presented; on the other hand, we will survey the methodological choices that were made in the studies. We will end with an overview on the formal organization of the thesis.

1 Writing and speech technology as a research object

In the late 80's and early 90's several studies were conducted to describe writing processes in the 'new' writing mode keyboard based word processing (Gould, 1981; Haas, 1989a, 1989b). Several studies directly compared this writing mode with pen and paper writing (for a review see: Goldberg, Russell, & Cook, 2003; Van Waes, 1991; Van Waes & Schellens, 2003). The main conclusion was that writers when writing with a computer organize their writing processes more recursively than when writing with pen and paper. Moreover, keyboard based word processing shifted attention to the more formal aspects of texts (Bangert-Drowns, 1993; Bridwell & Duin, 1985; Van Waes & Schellens, 2003). These studies not only showed that computer based word processing influences the organization of writing processes they also introduced the use of word processors as a research instrument to reveal those writing processes.

The emergence of speech recognition as a new technology for writing created a new writing mode, which may in turn influence the organization of writing processes. Comparable to research in computer based word-processing our interest in speech recognition also has a double agenda: on the one hand, we are interested in the organization of speech recognition based writing process – speech recognition as a research object – and on the other hand we are interested in speech recognition as a research instrument. This double approach is demonstrated in the various research projects in this thesis.

In the year 2000 little was known about the cognitive processes of writers who use speech dictation software. Therefore, the focus of the first research project in this thesis was on the adaptation strategies of novice speech recognition users. Since keyboard based word processing caused a shift in writing processes, we also expected speech recognition to have an effect on the organization of writing processes.

We opted to observe the writing processes of two writer groups in an exploratory study at various moments in their learning process. On the one hand, we were interested in the adaptation processes of writers who already had previous dictating experience, the professional group of lawyers. On the other hand we expected that the adaptation to a new writing mode of writers, who were not familiar with classical dictating, might be somewhat different. Therefore, we opted to observe the writing processes of another group of professional writers: academic staff members. Our main interest in this part was to gain insight in how different users learn to write with a new writing mode; when they switch between different writing modes and writing (sub)processes; and how they adapt their planning, formulating and reviewing behavior to speech recognition.

The common denominator in the study is that the speech recognition mode creates a writing environment that is open for different writing styles. Neither writing experience nor learning style imposes a very strict writing profile on the writers. In other words, the speech recognition mode itself does not seem to force writers to adopt a specific writing style, as opposed to what happens when writers have to adjust to write their texts in either the classical dictating or the word processor mode.

We also noticed that novice users of speech recognition frequently need to correct errors in the text produced so far (TPSF), because of (technical) misrecognitions by the software that can lead to text that shows a deficient resemblance to the writer's intention. Speech recognition causes misrecognitions like when 'writing' is represented as 'right thing', or 'dual' as 'jewel' on the screen. These new types of errors seem to disrupt the writers' focus on text production. However, writers seem to vary in the way they deal with error correction during text production: they develop various strategies. Since error correction is known to be a writing subprocess that increases the cognitive load during writing, the focus of our research shifted to error correction of specific speech recognition errors.

In short, speech recognition causes unfamiliar errors in the text produced so far and writers use various strategies to solve the errors. Also, different error types seem to lead to different error correction strategies. Since, the errors in writing with speech recognition are different from keyboard based word processing errors they require different correction strategies. Previous research has shown that non-existing words are less demanding to correct (Hacker, Plumb, Butterfield, Quatham, & Heineken, 1994; Larigauderie, Gaonac'h, & Lacroix, 1998). Semantic errors are more demanding since the context of the text produced so far needs to be taken into account. It is inherent in the underlying technology and its use of predefined lexica for speech recognition causes more semantic errors in the text. This characteristic makes it more difficult for the writer to benefit from the main strength of speech recognition (speed of composition) and complicates the revision process (Honeycutt, 2003).

To find out in greater detail what the influence of the text produced so far is on the cognitive load and the writing behavior, we set up a controlled reading-writing experiment. So we opted to focus more explicitly on error correction. We analyzed

the correction behavior of writers related to four different error types: large speech recognition errors, small speech recognition errors, small speech recognition errors that could also occur in keyboard based writing and errors that could only occur in keyboard based writing since they resulted in non-existing words. We also hypothesized that vocalization could be an important discriminating characteristic of writing with speech recognition. When writers produce a text via speech recognition they also hear themselves dictating the text produced so far. The addition of an auditory channel that is created by voicing the TPSF may influence the writer's error correction strategies. This auditory information enables writers to reprocess the information and may help to preserve it in working memory (until this information is replaced by new information). Pilotti, Chodorow, & Thornton (2004) found that familiarity with auditory feedback improved reading to correct errors, both in speed and accuracy. However, if we translate this result to writing processes then hearing the dictated text may focus the writers on continuing text production. This focus on production might cause a burden to switch subprocesses. In theory, writers do not need to read the TPSF to continue text production and could stick to a more linear pattern like in classical dictating (Gould, 1978; Schilperoord, 1996).

In our study we isolated the effects of writing modes (auditory feedback versus no auditory feedback) from various error types. The error types cause a different cognitive load on writers and also the accuracy of solving error types differed. Again, various writers use different strategies in correcting errors immediately or delaying error correction.

As stated before, speech recognition can cause more meaning based errors since the words that are inserted are correctly spelled (surface) but the meaning in the context may be inadequate. Correcting these types of errors has been shown to be more difficult for 'child, adolescent, young adult, and older adult revisers, for topic and linguistic novices and experts, for texts written by the revisers and texts written by someone else, [...]' (Hacker, 1994, pp. 143). Since error correction in general appears to be such a difficult task for various writers, of various expertise levels, we are interested in the strategies used by writers who are familiar with speech recognition based errors. Professional speech recognition users may have developed successful error correction strategies for dealing with the specific errors that are generated by speech recognition software.

Therefore, in a third study we focused on error correction strategies of professional writers and speech recognition users while they were writing a strategic and quite complex business text report. Not only detecting errors places demands on cognitive resources, but also solving a diagnosed problem is cognitive demanding. Writers must constantly compare the text on the screen with what they had in mind. Furthermore, these comparisons need to be related to the more fundamental knowledge about the text, the goal, the audience etc. This combination of reading and writing – and complex balancing between subprocesses – is essential for high quality text production.

2 Converging research methods

In this thesis we have used a combination of exploratory versus hypothesis based research, qualitative versus quantitative data approaches, ethnographic oriented research versus experimentally controlled research, aggregated data analysis versus multilevel analysis, general descriptions versus case analysis. In other words, we have opted for converging, complementary research methods to describe writing processes in great detail. Figure 1 shows the relation between the four research projects that are presented in this thesis.

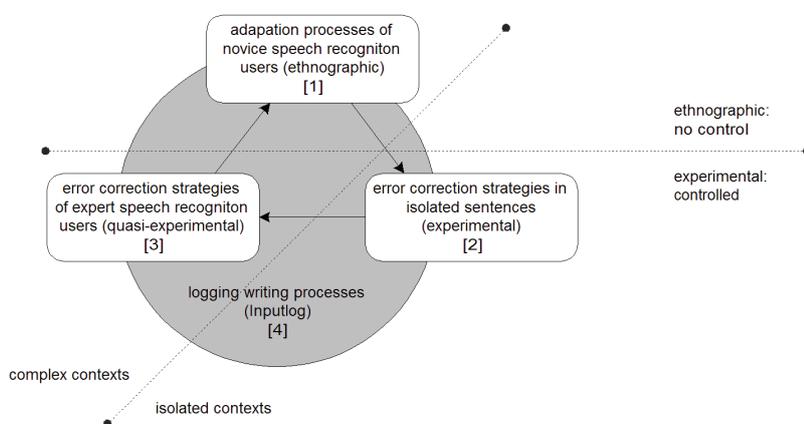


Figure 1. Relation between four research projects.

In the first study we have described the adaptation processes of novice speech recognition users. Two writer groups (lawyers and academics) were observed at five moments in their own professional contexts: offices of lawyers and academics. To guarantee a high ecological validity the writers were observed both in their own offices as well as during writing sessions in which they wrote their own professional business texts (varying from a short routine letter to a complex summons). The advantage of this ethnographic approach is the ecological validity of the data observed. However, this resulted in a wide range of text types and genres, differing for instance in length and structure which makes it difficult to compare the participants' writing strategies.

Besides, the study aimed at developing an appropriate methodology for writing process research in which speech recognition was used as the main writing medium. Because of specific characteristics of the speech recognition mode, we developed a categorization model and a notation system to describe the writing process data. We collected both product and process data. The product data provided information about the length of the texts and the duration of the observation. The process data provided information about the gradual development of the writing process.

The process data were collected by an on-line screencam that is called Camtasia⁵, a sound recorder QuickRecord, and myself as an observer. Of course, logging via key-stroke logging would have been more precise and would have provided us with more detailed information on the writing processes. However, at that moment none of the existing logging tools could log in a professional Windows environment and nor could they log speech recognition.

In the second study we shifted to a controlled experimental study. Sixty college students were invited to participate in a one-hour experiment during which they had to take two short initial tests and completed two sets of reading-writing tasks in two different modes, that is, one with a spoken script before the text produced so far appeared on the screen and one without such a script. The writers were asked to detect errors and complete causal statements.

Since many processes during writing are not apparent in the final product, methods are necessary to uncover them online. We have introduced a new technique for experimentally isolating the impact of error type. A program especially developed for this purpose controlled the experimental flow and also recorded the participants' main input. It presented writing contexts and the TPSF during the writing task. Since we were interested in the cognitive demands which the various error types posed on the writers, we integrated a secondary task technique. During the primary writing task the program also presented an auditory probe as a secondary task, and recorded interference reaction time. A logging program called Inputlog (cf. infra) recorded the writing process and collected the writers' online strategy choices. These online measures provide insight in writing behavior (e.g. pauses) and more specifically strategy choice, such as, whether the writer chooses to correct an error before text production, or to continue text production and revise it later.

In this controlled experiment the writing process was limited to single sentences; error correction strategies were observed in isolated sentences. This highly controlled experimental setup allows focusing on fine-grained issues like the influences of different errors in the TPSF on working memory (via e.g., secondary tasks). This would be impossible to accurately describe in ecologically valid studies. Therefore, the combination of both research projects is valuable. The combination of starting with an exploratory study and pursue the line of research with a controlled experimental study complements one another (also Figure 1).

So far, the studies have generated insight in how writers used speech recognition for writing business texts, how they solved (technical) problems, how they interacted with the text produced so far, but they have not answered the question *why* writers made some strategic choices about error correction during their writing processes. To answer this question, we have opted to use converging research methods in the final study.

⁵ Camtasia is a product of TechSmith, the company that also produces Morae.

This enabled us to clarify the writers' underlying motives in choosing writing strategies. We have gathered product, process and protocol data.

The professional speech recognition users wrote a business report of about one-and-a-half page. The data of the writing process are logged by the logging program Inputlog, the speech recognizer Dragon Naturally Speaking 8.1, and the usability program Morae. The logging of the writing process data did not interfere with the writing process. In addition, we observed the writers and the writing processes concurrently from another room. After the participants had finished their writing tasks they were asked questions in a retrospective thinking-aloud protocol. On the basis of the findings of the concurrent observation, we used active prompts during the stimulated retrospective interview. The findings from the product, process and protocol analyses corroborated each other.

The quasi-experimental study on error correction strategies of expert speech recognition users combines the most and the most diverging research methods in writing research: from keystroke logging data, visual screen data in combination with a webcam, to retrospective interviews. What is new about this approach is that writers are observed simultaneously and that they can review their writing processes in combination with their own actions after their writing session (Lindgren, 2005; Matsushashi, 1982).

In the above-mentioned studies we have combined a quantitative data approach with a qualitative data approach. The case studies, in which we selected two to three writers, are the most explicit elaboration of this approach. Via the case studies we have illustrated general findings on writing strategies in greater detail.

Finally, the last part of this thesis is connected to and puts together the three other parts. After the first research project, in which we have coded the revisions, writing mode operations and pauses manually, we have started developing a logging tool called Inputlog in 2003. Of course, keystroke logging already existed at that moment, but it could not be applied in professional writing environments like Microsoft Word or the logging of speech recognition. Inputlog can easily be used in various Windows environments and since 2006 it also logs speech recognition data from Dragon Naturally Speaking 8.1. Since the logging is based on timestamps, the data can also be merged with macro oriented observation tools like Morae.

Inputlog is a practical tool that enables researchers to log writing processes on (almost) every computer that has Windows as an operating system. Researchers can use it independently, but they can also opt to combine it with usability programs (Morae) and eyetrackers (Eyelink, Tobii) or integrate it in custom-made research tools. These integrations open up new perspectives for further analysis. Since speech recognition can be logged, this writing medium can also be used as a research instrument to log also, for instance, thinking-aloud protocols.

3 Practical organization of this thesis

To conclude we would like to state that the coming of age of speech recognition will further stimulate research on the effect of this new technology on the writing processes of both professional writers and writers with learning disabilities. Speech recognition contains the best of natural speech (VIVO), and furthermore provides visual feedback. In other words, speech input facilitates the formulation of new text; the text produced so far is visible on the screen and the simultaneous use of keyboard and mouse makes multimodal interaction with the text possible. However, more important than a hypothetical evaluation of writing comfort, the hybrid character of speech recognition enables us to bring characteristics of writing processes to the surface that were previously less explicit. By studying the particular ways in which writers use speech recognition, we can gain insight into the cognitive processes underlying writing. We see speech recognition as a magnifying glass or clogged shaker to bare the skin of writing processes.

This thesis addresses three main research questions:

1. What is the influence of speech recognition on cognitive writing processes and the organization of writing?
2. How do errors in the text produced so far influence text production (when writing with speech recognition)?
3. What (combination of) research methods are most adequate to observe and - quantitatively - analyze the writing processes in different writing modes without interfering in the writing process?

The thesis consists of 7 articles. These 7 articles are related to the four main research projects as described above (see Figure 1). The first project was on the influence of speech recognition on the adaptation processes and writing processes of novice speech recognition users (chapters 2, 3 and 4). In this first section we approach the collected data from three different perspectives: adaptation strategy, learning style and previous (classical) dictating experience. Chapter 2 focuses mainly on the adaptation strategies and learning styles of writers with the same writing background (lawyers that already had previous dictating experience). Chapter 3 extensively takes this difference in previous writing experience into account and focuses on the adaptation processes and writing processes. Next to this, the chapter describes the classification model in greater detail. Chapter 4 focuses on the error correction strategies of novice speech recognition users. Chapters 2 and 4 have been published as book sections and chapter 3 as an article in an international journal. Since they are based on the same research project, similarities in the theoretical framework might occur. Furthermore there is overlap in the methodological sections. However, chapter 3 provides the most elaborate framework, both on the research method and on the classification of speech recognition based writing processes.

In section 2 we describe the research project on error correction strategies in isolated sentences (chapters 5 and 6). Chapter 5 describes the theoretical framework and the research methodology in great detail. The analyses performed in this chapter are based on aggregated data. In chapter 6 we describe a more detailed re-analysis of the dataset via multilevel analyses. Writing researchers can also read this chapter as an illustration of conducting this kind of analysis⁶.

The third section describes a research project on error correction strategies of professional speech recognition users (chapter 7).

Finally in section 4, we wind up this thesis with an article on the logging program Inputlog (chapter 8). This chapter provides state-of-the-art information about Inputlog (April 2007). The latest version of Inputlog can be found on the Inputlog website (www.inputlog.net). The website also contains extra documentation on Inputlog.

An English summary of the four research projects described in this thesis can be found in chapter 9.

⁶ A practical guide can be found in Quené and Van den Bergh (2004) and De Maeyer and Rymenans (2004).

Section I

Adaptation and writing processes of novice
speech recognition users in their professional
working environments



2

How do writers adapt to speech recognition software?

The influence of learning styles on writing processes in speech technology environments

Abstract: This paper describes the adaptation and learning processes of writers who have started using speech recognition systems for writing business texts. We observed writers during five set moments in their daily work. The data from these observation sessions were used to describe the adaptation strategies during the learning process. In a case study we analyzed the learning processes of two writers with similar writing experience (classical dictating), but with a different learning style: accommodator versus diverger style (based on a taxonomy of Kolb). The participants of the case study differed mainly on (a) the amount of time they spent in the speech recognition mode, and (b) the mode they used to solve 'technical problems' caused by the speech recognition software. The results show that both participants have a different adaptation process, which evolves differently and seems to be driven by the learning styles of the participants.

Keywords: Speech recognition, writing processes, adaptation processes, learning strategies, research method, writing modes.

This chapter has been published as Leijten, M. (2007). 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.

1 Introduction

Speech recognition software enables writers to write their texts by ‘talking to the computer’. By adding a toolbar to the normal on-screen working environment (see Figure 1), the software enables normal keyboard & mouse commands to be voice-activated. For example: dictating text, navigating through programs or texts, emboldening words, and opening or closing files. (Williams et al. (2004) provide an extensive description of the practical impact of speech recognition software; Honeycutt (2003), gives a historical overview of speech recognition).



Figure 1. Speech recognition toolbar (Lernout & Hauspie Voice Xpress).

Until a few years ago research on speech recognition applications focused on qualitative (technical) improvement. More recent research has dealt with speech recognition as a writing tool (Halverson, Horn, Karat, & Karat, 1999; Hartley, 2007; Hartley, Sotto, & Pennebaker, 2003; C. Karat, Halverson, Horn, & Karat, 1999; J. Karat, Horn, Halverson, & Karat, 2000; Quinlan, 2004).

These studies mainly focus on the usability of speech recognition as a writing mode. In the study by Karat et al. (1999) initial and professional users tested the usability of three speech recognition systems. The initial users were given two kinds of writing tasks, one they had to perform with speech recognition, and the other with keyboard & mouse. The results showed that users struggled with speech driven error correction and that they spent more time making corrections than the researchers had expected. Perhaps this can be explained by (a) the goal of the study (usability and error correction), (b) the fact that the participants were observed all the time, and (c) the fact that they had just received extensive training in speech based correction methods.

The use of speech recognition software by the learning- and physically-disabled has also received some attention. Quinlan (2004) studied the writing processes and products of forty 11 to 14-year-old children. Twenty of these children had writing difficulties. The children composed a series of four narratives in one of two main writing conditions, handwriting and speech recognition, with a further two conditions for each mode, with or without advance planning. Quinlan hypothesized that speech recognition would provide cognitive benefits to children with writing difficulties, and that advance planning would be supportive for the real-time planning process. The results showed that the children with writing difficulties produced longer texts when using speech recognition than when using handwriting. Speech recognition seemed to reduce transcription-related interference, enabling the children to produce more fluently written – and hence longer – texts. For the fluent writers, however, composing with speech recognition did not lead to improved narratives. Advance planning had a significant positive effect on text quality for both groups.

Hartley et al. (2003) compared the academic correspondence produced by an experienced writer, initially using keyboard & mouse and then using speech recognition. The writing products differed only slightly in their average sentence lengths, the use of especially long sentences and the first pronoun (see also Hartley, 2007).

In a previous study we described a similar case study (Leijten & Van Waes, 2003), in which we focused on writers with dissimilar writing experiences, that is, those with previous classical dictating experience versus those. This study showed that speech recognition does not impose a specific writing style on the user, contrary to previous findings stating that pen & paper writers had to adapt to the keyboard & mouse environment first.

Speech recognition can be described as a hybrid writing mode, because it combines elements of dictating and computer aided writing. Traditional dictating is characterized by a high degree of linear text production (Gould, 1978; Gould & Alfaro, 1984; Schilperoord, 1996), whereas writers using speech technology receive immediate visual feedback from their computer screen. This allows them to review the text at all stages of the writing process, thereby opening the gates to non-linearity. Indeed, a high degree of non-linearity is typical of computer writing processes (Lee, 2002; Severinson Eklundh, 1994; Severinson Eklundh & Kollberg, 2003; Van Waes & Schellens, 2003). The constant feedback from the screen allows writers to revise continuously, without losing an overview of the final text.

Learning

An important characteristic of this study is that the participants had to learn how to produce texts in a different way. Therefore, we will pay attention to different learning styles. In this study we used Kolb's definition of learning, namely "the process whereby knowledge is created through the transformation of experience" (Kolb, 1984, p. 38). In this definition we would like to emphasize two aspects of the learning process which are especially important from our perspective. The first one is the emphasis on the processes of adaptation and learning as opposed to content or outcomes. The second one is that knowledge is a transformation process; it is continuously created and recreated. The knowledge people have of working with speech recognition is constantly transformed by the experience they build up while working with the program.

Kolb describes the learning process as a four-stage cycle involving four adaptive learning modes: concrete experience, reflective observation, abstract conceptualization, and active experimentation.

In his Learning Style Inventory he evaluates the relative preferences an individual holds for each of these four learning modes. The Learning Style Inventory is an objective, self-scoring instrument that reveals four statistically prevalent learning styles: diverger, converger, assimilator, and accommodator.

People with a converger or assimilator learning type prefer a higher level of abstract conceptualization. Convergents prefer to learn via the direct application of

ideas and theories and have been described as somewhat unemotional, preferring to work on their own. Assimilators are good at taking in a wide range of information and reducing it to a more logical form. They tend to prefer theoretical models and deductive reasoning; this leads to a greater interest in abstract concepts and ideas than in interaction with other people.

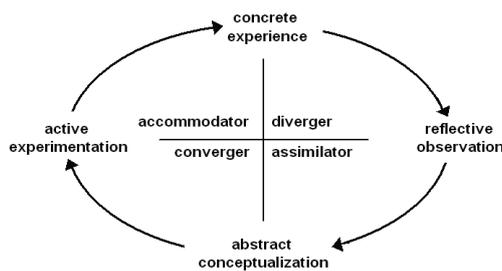


Figure 2. Learning styles (Kolb, 1984).

Divergers and accommodators rely heavily on concrete experiences when learning. Because they also prefer active experimentation, accommodators are described as having the ability to carry out plans and get things done. They get involved quickly in new material by way of trial and error learning. Divergers are identified by their ability to look at a learning situation from multiple points of view. They often have a hard time making a decision and mostly prefer to observe rather than to participate (Kolb, 1984; Terrel, 2002).

In this study we have opted to observe writers in their professional environments. The texts they were writing were part of their 'normal' work; the only difference was that we asked them to use speech recognition as a writing medium. Kolb predicts that people in technology and information science careers generally benefit from a converger or an assimilator learning style. Consequently, we expected that those participants who were characterized by either converger or assimilator learning styles, would outperform other participants. But of course, participants whose 'preferred' learning style is less suitable for learning to work with a new writing mode, will also be able to adapt their learning style to meet the requirements of this new mode, perhaps less efficiently. In this study we will take a closer look at the concrete strategies participants develop in learning to work with speech recognition.

In this chapter we focus on the adaptation and learning processes of writers learning to write their business texts in the new medium of speech technology. We are mainly interested in the general cognitive processes that characterize the writing process and will become more visible by focusing on a new writing medium: How will the writers adapt their writing to the new medium? Will their writing strategy change over time? What is the influence of learning style on the initial use of speech recognition?

2 Description of the study

In this section we will first describe the participants involved in this study, then we will elaborate on the design and procedure of the study, and finally, we will briefly describe the writing materials that the participants produced.

In the case study presented here, we will focus on the writing and adaptation processes of two participants, Frederik and Bart. Both participants only had to adapt to the speech software because they were experienced in working with computers and dictating devices. Frederik, an associate in a large law firm, had 5 years of work experience and about 6 years of experience working with computers and classical dictating devices. Bart had been working as a lawyer for 9 years, but had a comparable level of experience in working with computers and dictating devices. However, the reason to select them for this case study was that they were characterized by a different learning style (cf. Learning Style Inventory of Kolb, 1984, as described in Figure 2).

According to Kolb's taxonomy, Frederik is an accommodator. He scored high on active experimentation and concrete experience. Kolb's model states that people with an accommodative orientation tend to solve problems in a trial-and-error manner. They also rely heavily on other people for information rather than trust their own analytic ability. People with this learning style are sometimes seen as impatient and 'pushy'. As Kolb predicted, Frederik's learning style is determined by his job to the extent that he has to make a lot of decisions in uncertain and unpredictable circumstances.

Bart is a so-called diverger. He scored high on concrete experience and reflective observation. In this learning style the emphasis is on adaptation by observation rather than by action. A person of this type performs better in situations that call for generation of alternative ideas and implications. Furthermore, they tend to be imaginative and they prefer to reflect on situations. Characteristics of Bart's job that relate to his learning style are personal relationships and effective communication with other people.

Both participants have another learning style than Kolb predicted to be most adequate for learning to deal with (new) technology. However, since learning styles are adaptable to various tasks, we can predict that both participants will be – in some way – influenced by the task requirements of writing in a new medium. The active orientation of both participants will guide them to learn certain tasks and to improve active skills.

Design and procedure

The participants were required to complete a Dutch translation of Kolb's Learning Style Inventory (Kolb, 1984), which provided information about differences in learning styles. In a post hoc analysis we used this information to determine the learning styles of the participants.

Before the participants started using speech recognition for the first time they watched an introductory video about the use of the speech technology program. This video was provided by the software company. The participants were then informed about the procedures of the study in more detail. They were asked to use the speech recognition system during their day-to-day work for at least three hours a week. The participants could decide for themselves how to use the software and they were not restricted to the exclusive use of speech input. Keyboard & mouse could also be used as complementary input devices. In total we observed the participants five times while they were writing in their own environment, respectively after 1, 3, 6, 9 and 12 hours of working with speech recognition.

The data were collected using an online camera (Camtasia™) and a sound recorder (QuickRecord™). Because of the combination of the different input modes (keyboard, mouse and speech) we were not able to use existing logging programs. We observed the participants during each writing session and took notes about specific writing circumstances that could not be registered in any other way (see Figure 3). These recordings and notes enabled us to reconstruct the writing process in detail.

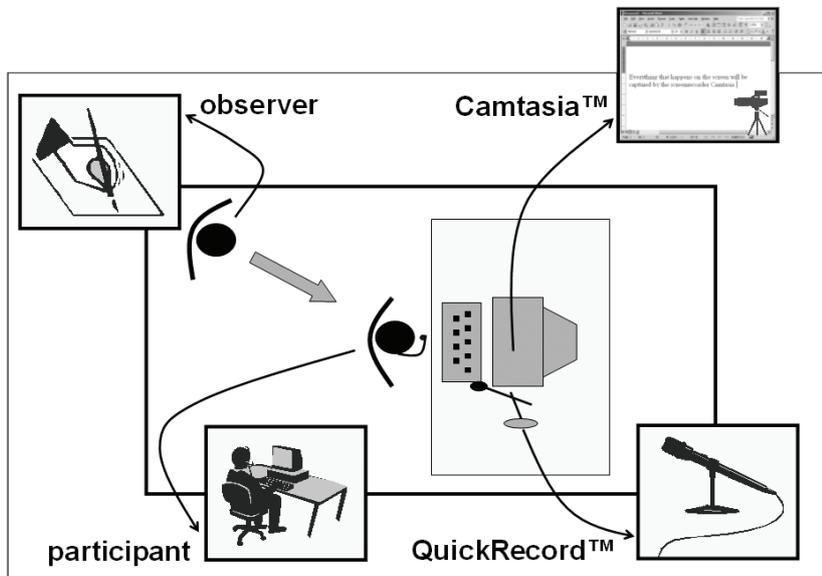


Figure 3. Observation setting.

3 Results

To describe the product and process data of the study a categorization model was developed that takes the complexity of the hybrid writing mode into account.

This model also made the enormous amount of process data accessible for further research (for a full description see Kollberg, 1998; Leijten & Van Waes, 2005b, chapter 3; Severinson Eklundh, 1994; Severinson Eklundh & Kollberg, 1996; Van Waes & Schellens, 2003).

First we describe the product data - the material - to contextualize the writing activity. In the second section we describe the use of the speech recognition writing mode and focus on the input of the three possible writing modes: namely speech, keyboard & mouse. Finally, we describe the repairs. We have adapted the concept of repairs from conversation analysis, because it provides a valuable addition to the traditional concept of reviewing. Moreover, it enabled us to incorporate a broader view of reviewing, taking into account the specificity of the reviewing process in the speech recognition mode. Repairs could either refer to 'technical problems' or to 'revisions'. 'Technical problems' in this study are things that go wrong because of the speech mode (C. Karat et al., 1999). For example:

- misrecognitions (dictated text is misinterpreted and a non-intended text appears, e.g., 'eye' instead of 'I');
- command misrecognition (for example, instead of executing the command 'end of sentence', the text 'end of sentence' appears on the screen);
- zero recognition (the computer does not respond).

In the repair analysis we distinguished these technical problems from 'revisions'. Revisions are changes in the text after an evaluation of the previously written text. These changes are not initiated by technical problems but by the writer. They are designed to change the content, formulation or appearance of the text.

3.1 Material

The product data informed about the length of the texts, and the duration of the observations. The different tasks that the participants carried out were job-related and part of their normal writing activities, for example letters, e-mails or reports. Because they were familiar with these kinds of texts, the tasks could be performed structurally and routinely.

Table 1. Characterization of writing product in 5 observations

	Task	Length observation	Number of words in final text	Mean number of words per minute
Frederik				
1	letter	4'08"	92	22.27
2	petition	22'17"	449	20.15
3	conclusion	21'23"	419	19.59
4	conclusion	23'33"	247	10.49
5	letter	19'46"	305	15.43

	Task	Length observation	Number of words in final text	Mean number of words per minute
Bart				
1	e-mail	6'00"	62	10.33
2	conclusion	24'33"	586	23.86
3	petition	22'13"	520	23.42
4	report	17'25"	419	24.05
5	petition	37'00"	685	18.52

The observation session differed in length because the participants were observed while working on a task they had chosen themselves. The first observation of both participants lasted about 5 minutes and the mean time of the remaining observations was 23'31" (Frederik: $M = 18'05''$, Bart: $M = 21'18''$). During this time Frederik produced an average of 17.76 words per minute whereas Bart produced 20.33 words per minute. Because of the variable length of the observation sessions, the texts also differed in length. In the first observation the participants explored the program and wrote a very short text; during the subsequent observations their texts were between 247 and 685 words per session (Frederik: $M = 355$ words, Bart: $M = 552.5$ words). Table 1 gives a description of the five observation sessions.

3.2 Writing modes

The mode analysis shows that both participants use the potential of speech recognition quite differently (Table 2). Frederik uses the speech input and the keyboard & mouse input almost just as much. The part of the speech mode drops gradually from the first observation session (47%) to the final session (34%). On the other hand, Bart uses the speech input almost twice as much as Frederik. For over 80% of his writing tasks, he uses the speech mode. This way of writing adheres to Bart's traditional dictating habits and he hardly changes this strategy over the different observation sessions.

Table 2. Mean use of writing modes in percentages (5 observation sessions)

	Frederik		Bart	
	%	SD	%	SD
Speech	42,5	8,9	81,1	7,6
Keyboard	44,4	13,9	13,8	7,2
Mouse	13,1	9,4	5,1	2,2

If we compare the use of writing modes in the first and second part of the writing process, it can be seen that Bart prefers to write his texts with speech input and that his choice of writing mode stays the same over the two writing parts. Speech is his preferred writing mode, both in the formulation and revision writing subprocesses.

On the contrary, Frederik uses speech input in the first part of the writing process for more than 50% of the time. In the second part he clearly prefers to write with keyboard & mouse. The use of speech input to dictate text segments drops to a mean of

32.7%. In the first observation session Frederik used speech for 40% of the process time in the second part of the writing process. However, in the last session, the speech input almost completely disappeared from the writing process. He hardly uses it to finish his text (cf. *infra*). In Bart's writing process, changes in the use of writing modes are hardly seen. His use of speech recognition remains relatively constant (80%) in the different writing sessions and in both parts of the writing process.

In summary, it can be said that the mode analyses confirm to a large extent the learning styles of both participants. Frederik, the accommodator, actively explores in the first sessions the possibilities of the speech recognition mode, but gives in after some trial and error. His use of speech recognition is rather restricted. Bart, on the other hand, as a diverger, uses the speech recognition system much more intensively and he explores the possibilities in several phases of the writing process. Table 1 shows that these explorations do not have a negative effect on the writing 'performance' Bart produces even more words per minute than Frederik (mean per session: Bart 20.33 ($SD = 5.66$) vs. Frederik: 17.76 ($SD = 4.72$) words per minute).

3.3 Repairs

In total we observed more than 500 repairs (technical problems and revisions). Frederik interrupts the linearity of his writing process 306 times in five observation sessions and Bart has 223 recursive actions (Frederik, $M = 61.20$ per session vs. Bart, $M = 44.60$). If we correct these data for the time differences between the observation sessions, this results in 3.45 repairs per minute for Frederik, as opposed to 2.09 repairs per minute for Bart. These numbers may seem high, but they are in line with Levy & Ransdell (2001) in their study of 'concurrent memory loads', they found that over frequent concurrent task presentation (comparable to interruptions of e.g. technical problems) may lead the research participant to devote his or her full attention to the secondary task, thereby reducing the opportunities for proficient text generation. They concluded that critical events for secondary tasks should occur about four times a minute, a conclusion which compares favorably to our observation of the number of times Frederik and Bart are willing to interrupt their writing process.

Table 3. Writing mode used before a repair in percentages (5 observation sessions)

	Frederik		Bart	
	%	SD	%	SD
Speech	6.63	5.24	55.01	29.98
Keyboard & mouse	88.13	8.62	31.48	22.74
Combination	5.24	3.75	13.51	11.04

The data in Table 3 also show that both participants deal with repairs quite differently. As we have seen in the mode analyses, Frederik hardly uses the speech mode to solve technical problems or to revise (6.6%), while Bart tries to use speech recognition in 55% of the cases for recursive text interventions. If we add the repairs that are made

with a combination of writing modes, including speech, the difference is even more significant. Frederik does 90% of the repairs exclusively with keyboard & mouse. In Bart's case, this percentage drops to 30. He chooses to explore the possibilities of speech recognition. In 30 to 40% of the cases during the first observation sessions Bart needs more than one attempt to solve a (technical or textual) problem. During the course of the observation session the solutions taking more than one attempt decline to less than 10%. Frederik, however, needs no more than one attempt to solve a problem in most cases, because he prefers to use keyboard & mouse for text repairs. As such, he seems to be less eager to explore the possibilities of speech recognition.

3.3.1 Technical problems

If we compare the proportion of technical problems then it becomes clear that the first category is the most extensive one (Frederik: 81.4% technical problems vs. Bart: 91.1% technical problems). This pattern is typical of the adaptation process, as proved by the low number of technical problems in the last observation session (Frederik: 70.7% vs. Bart: 79.8%).

The analyses of the technical problems show a different pattern for both writers. Frederik struggles more with the new technology than Bart does. If we compare the technical problems per 100 words, we see that Frederik has to deal with almost one third as many technical problems than Bart (Frederik: $M = 16.75$ problems per 100 words vs. Bart: $M = 10.43$).

We can again relate this result to the learning styles of both participants. Frederik is fairly impatient and he is primarily focused on productivity. As a result, he keeps writing in his traditional way: dictating larger text segments with only short interruptions. He verifies the result of his dictations on the computer screen rather superficially. Consequently, he is not able, or not willing to adjust his use of the speech recognition tool. He repairs his technical problems mostly in the second writing phase when he rereads the produced text, again mostly without the use of speech recognition. On the other hand, Bart uses the text that appears on the screen more consistently during the formulating phase. He reflects on the possible causes of the problems that occur (misrecognitions) and consequently tries to anticipate and to avoid these problems. This strategy results in a lower problem ratio.

3.3.2 Revisions

It is striking that both professional writers revise very little in the five observation sessions. This may be partly explained by the fact that both writers performed routine tasks. Furthermore, the writing strategy of dictating writers is also relevant: these writers mentally preformulate and therefore already partly revise their text before one word is dictated. A large part of the dictators' revisions of takes place in the writers' mind. This limits the on screen revisions and places them higher in the text hierarchy (Frederik: 43.21% of the revisions are above the word level vs. Bart: 73.87%; see Table 4). Both writers partly maintain this writing style in the new writing mode, but we still see that the representation of the text starts to play an active role in the writing

process. Contrary to the writing process that is typical for traditional dictators (Gould & Alfaro, 1984; Schilperoord, 1996) we now observe that recursivity becomes more apparent in the writing process of these writers.

Nevertheless, both writers also show a different revision behavior. Frederik revises 3.41 words per 100 words versus Bart: 0.93. If we assume that the mental revision is also reflected in the pausing behavior of the participants, then we can incorporate this factor in our explanation. Bart's pausing time takes up about 60% of his writing process, while for Frederik this amount drops to 30% of the total time. This distribution of pausing and writing remains constant for both writers over the five observation sessions. The difference in pausing behavior is also in line with the learning styles of both participants: experimenting vs. reflective.

Table 4. Distribution of the revisions per level and per writing phase (5 observation sessions)

	Frederik		Bart	
	%	<i>SD</i>	%	<i>SD</i>
Level				
Word	56.79	13.52	26.13	26.94
Sentence	35.90	5.73	35.80	44.67
Higher	7.31	11.08	18.07	31.35
Writing phase				
Phase 1	42.95	30.01	75.29	43.31
Phase 2	57.05	30.01	4.71	10.52

Moreover, the distribution of the revisions in the two writing phases differs for both participants (Table 4). Frederik is a writer who – in accordance with his previous dictating experience – mainly revises in the second phase of the writing process, after he has finished a first draft of his text. Approximately 60% of his revisions are in the second phase. Bart, however, hardly revises in the second writing phase, only 5%. This kind of revising behavior closely matches the writing profile of the professional writer who uses a word processor to write his texts. These writers also prefer to revise their texts during the writing process of the first draft, leaving only few revisions for the rereading phase.

4 Conclusion

In this case study we have proposed an analyses of two professional writers who, while equally experienced in classical dictating, started working for the first time with speech recognition as a dictating device. Both writers are characterized by a different learning profile (accommodator vs. diverger). We wanted to learn more about the cognitive aspects of learning to write with a new writing mode. How do writers adapt to a new writing mode? And how do learning styles influence the initial use of speech recognition?

The mode analysis illustrated that Frederik explores the possibilities of the new

writing mode actively in the first writing sessions, but that he opts rather quickly for a selective and modest use of the speech mode. He uses speech recognition mainly to dictate his text and does not use it to solve technical problems or to revise his text. This pattern confirms the behavior characteristic of accommodators (trial-and-error and get things done). Moreover, during the completion of the text Frederik 'evolves' in his use of speech recognition to a writing pattern that is comparable to the writing processes of writers using classical dictating devices. He dictates his text and then adjusts his text in the second writing phase by using keyboard & mouse. The text on the screen has a more passive, monitoring function and not a guiding function.

Bart explores the different possibilities of the new writing mode more systematically and more patiently. His writing process is very reflective and is characterized by long pauses. During these pauses Bart actively rereads the text produced so far on the screen and when necessary he repairs it. When possible he tries to work with speech input. Bart uses the speech mode during 80% of his writing time, twice as much as Frederik. In Bart's case speech recognition is definitely not an elevated dictating device, but more a hybrid writing tool that combines elements of classical dictating devices with the classical computer mode (keyboard & mouse). In Bart's adaptation process, quantitative movements between the writing sessions were rare. His profile remained stable over the different sessions. As a 'diverger' he explores the possibilities of speech recognition and develops a satisfactory way of working with speech recognition (22 words per minute) through reflection and observation. In studies by Karat (2000) and Halverson (1999) the initial users produced 13.6 words per minute (keyboard & mouse $M = 32.5$ words per minute) and the four experienced users who worked for 30 hours with speech recognition produced 25.1 words per minute.

These descriptions show that both participants have a different adaptation process, which evolves differently over time and seems to be driven by the learning styles of the participants. Consequently the speech input mode has a different effect on the writing behavior of both writers. The characteristics of both writers are summarized in Table 5.

Table 5. Characteristics of the adaptation and writing process of Frederik and Bart

	Frederik	Bart
Learning style	accommodator	diverger
Use of speech mode	active exploration (initially) selective use (mostly formulating)	systematic exploration dominant use (formulating and revising/repairs)
Adaptation process	gradual evolution of hybrid use to a more traditional dictating process	continue adaptation and hybrid use
Revision behavior	mainly in the second writing phase	mainly in the first writing phase

5 Discussion and further research

In this chapter we described a case study and found that the participants hardly adapted their personal writing strategies to speech recognition.

To obtain a more complete picture it would be interesting to compare the writing modes of speech recognition to keyboard & mouse in a more controlled experimental setting. In the present study the participants produced different text types, varying from a short routine letter to a complex petition. This observation was not considered problematic, because we wanted this study to have high ethnographic validity; the participants worked in their own environments and developed their own strategies. However, to compare the writing and adaptation strategies of writers in both the keyboard and mouse mode and the speech recognition mode, a follow-up experiment with a Latin square design with counter-balanced texts and modes of writing would have to be carried out.

In such a study we would also like to include classical dictating devices as a third writing mode. Because, as was mentioned earlier, the focus of this overall study was not on the speech recognition itself but rather on its effect on different aspects of writing processes. The methods of analysis used in this study show that focusing on a new writing mode enables us to gain an insight into fundamental cognitive aspects of the writing processes. Or as Haas stated:

As is often the case, new situations – in this case new technological contexts and situations for writing – bring to the fore aspects of writing that may have been there all along, but that have not been previously noticed. [...] – something with which writers have presumably been operating all along – becomes obvious in such a new situation or context: the changed technological context of composing with word processing. (Haas, 1996, p. 117)

So, by observing and contrasting the writing process in the speech recognition, keyboard & mouse and classical dictating mode, we would also hope to gain insights into certain aspects of the writing process in general.

From a more methodological point of view we would like to raise two points. Firstly, for the purpose of this study we designed a categorization model analyzing writing modes, technical problems and revisions (for a full description see Leijten & Van Waes, 2003b; 2005b, also chapter 3). The variables of this categorization model were useful in studying the writing process at case level. However, if we want to analyze a larger set of participants these categorization models are too time-consuming to work with. Thus we need a more automated way of collecting and analyzing writing process data.

Consequently, a point of particular interest with this kind of research lies in the automated logging of the writing processes. For this study we did not make use of any of the existing logging tools. Software like Trace-it (Kollberg, 1998; Severinson Eklundh, 1994; Severinson Eklundh & Kollberg, 1996) and Scriptlog (Holmqvist, Johansson, Strömqvist, & Wengelin, 2002) make it possible to register, reconstruct and analyze detailed online writing processes on several levels (including pause-

analyses). However, these existing logging tools have been developed for specific applications, and are hardly adapted to the current Windows environment and commercial word processors. Moreover, none of these logging tools have integrated the logging of the speech recognition mode.

As such we did not use any of the above-mentioned software. Because of the speech input we were forced to broaden the existing methods. The manual way of analyzing the data was very time-consuming. In the near future we would like to integrate the use of speech recognition software with the logging program Inputlog that is currently being developed¹ (also chapter 8). This would facilitate data collection, description and analysis.

Acknowledgements

We would like to thank both participants for their willingness to learn how to work with speech recognition and for letting us observe them while doing so. We are grateful to Lernout & Hauspie for the free license they granted for VoiceXpress Legal™ during the course of this study. We would also like to thank Tom Van Hout for proof-reading this chapter.

¹ More information on the development of the logging tool Inputlog can be found on the website of the program: www.inputlog.net.

3

Writing with speech recognition

The adaptation process of professional writers with and without dictating experience

Abstract: This paper describes the adaptation and writing process of writers who have started using speech recognition systems for writing business texts. The writers differ in their previous writing experience. They either have previous classical dictating experience or they are used to writing their texts with a word processor. To gather the process data for this study we chose complementary research methods. First the participants were asked to fill in a questionnaire and given instruction about the speech recognition system. Then they were observed five times using the speech recognition system during their day-to-day work. Finally, they also filled in a logging questionnaire after each task.

The quantitative analysis of the use of the writing mode shows that those participants who had no previous dictating experience, tend to use the voice input more extensively, both for formulating and reviewing. This result is confirmed in the more detailed case analysis. The other analyses in the case study – i.e. repair, revision, and pause analysis – refine the differences in the organization of the writing process between the writers, and show that the speech recognition mode seems to create a writing environment that is open for different writing profiles.

Keywords: Speech recognition, writing processes, dictating, adaptation processes, research method, writing modes, writing experience, writing profiles.

*This chapter has been published as Leijten, M., & Van Waes, L. (2005). Writing with speech recognition: The adaptation process of professional writers with and without dictating experience. *Interacting with Computers*, 17(6), 736-772.*

1 Introduction

Until a few years ago, research on speech recognition focused on the technical improvement of the technology. Research on the applications of speech recognition in a business context was difficult, because of major technical shortcomings. However, continued research efforts have achieved a significant breakthrough in developing commercial speech products (Lernout & Hauspie¹, Philips, Dragon and IBM) that make it possible to dictate fluently to a computer with an acceptable error margin (after practice). Since the late nineties, some usability studies have been conducted on the use of speech recognition systems (Halverson et al., 1999; Karat et al., 1999) but, so far, little is known about the cognitive processes of writers who are using speech input devices (Honeycutt, 2003).

Previous research has shown that next to several social and individual factors, also the writing mode (such as pen and paper versus computer) influences the organization of writing processes (Hayes, 1996). The emergence of speech technology as a new input mode for writing processes has created a new writing mode, which may again influence the organization of writing processes. Writing with speech recognition systems develops a hybrid modus between dictating and writing with keyboard & mouse. Contrary to traditional dictating processes, speech recognition systems offer the possibility of immediate text feedback on the screen as is the case when writing with keyboard & mouse. This seems to be the biggest advantage of speech recognition over classical dictating. The inability of writers to keep track of the external representation of the text produced so far is a common problem in classical dictating. To monitor their text, dictators must maintain a mental representation of the text in their memory, or if this is not sufficient, relisten to the tape (Reece & Cumming, 1996). Speech recognition, however, does offer this visual external representation of the text, which makes it possible for writers to (re)read their text from the screen.

The goal of this article is not to stress the technical aspects of the current speech technology systems, but to show how users adapt their writing to this new mode and what strategies they develop in using speech recognition systems in their day-to-day work. (In the study reported here, we used L&H Voice Xpress Legal™ version 4) The scope of this paper is threefold. Firstly, we focus on the (cognitive) aspects that characterize the writing processes of people writing in this new writing mode, that is using speech technology. The other focus in this study is the adaptation process and how initial users adjust their writing strategies to a new writing mode: we would like to investigate how different users learn to write with a new writing mode; how they solve problems; how they use a combination of different writing modes for different writing activities; when they switch between different modes; how they adapt their planning, formulating and reviewing behavior. In answering these questions we will also pay attention to the way in which these kinds of research questions can be answered, by describing a categorization model that we had to develop, in order to

¹ Lernout & Hauspie is nowadays Nuance.

take into account the specific characteristics of speech recognition data. Finally, we will focus on the methodology we developed to observe and analyze the writing processes with the speech recognition mode.

In section two we will first situate this study in a broader context of research about writing processes and speech recognition. Then we will give a description of the research project this study is based on. Section four gives a description of the analyzing methods we used and the categorization model we developed. The categorization model consists of four topics: writing modes, repairs, revisions and pauses. These topics are individually categorized at several levels to make the data accessible for analysis. For writing modes we study the use of the different writing modes and focus on their part in the writing process. In the repair analysis we study the way in which the participants deal with technical problems and revisions while writing. We have adapted the notion of 'repair' from research in the field of conversation analysis. Among other things, it is used to refer to corrections of misunderstanding and mishearing in natural speech. The concept enables us to take into account the specificity of the reviewing process in the speech recognition mode. Repairs in this study are categorized in type of technical problems, difference between intended text and outcome, cause of the problem, number of attempts to solve the problem, preferred writing mode to solve technical problems, level and remoteness of correction, direction of corrections and the temporal location of the repairs. Although revisions are part of the repairs, we will also describe them separately as an important subprocess in the writing process. The categorizations of the revisions are to a large extent the same as the ones we use in the general repair analysis. The last categorization concerns pauses, because they also create a perspective to gain insight in cognitive processes. We categorized the length of the pauses, the number of pauses and the temporal location of pauses. We also give a short description of a notation model we developed to make the data accessible and to illustrate specific observations. The notation model enables us to transcribe short fragments of the writing process, and to point at interesting findings. In the fifth section the results of the writing mode analysis are presented. We end with a more detailed analysis of a case study in which the writing process of a participant with previous (classical) dictating experience is compared with a participant who did not have this experience. The participants are both lawyers with comparable working experience and computer experience. Their writing process is described from the different perspectives in the categorization model. Using the transcription model, we also transcribe two short fragments as a detailed illustration of the analyses, and to make the results of the case study more profound. Conclusions are drawn in section seven, and in section eight we discuss these conclusions in the broader perspective of the research project, related research and further research.

2 Related research: writing processes

Writing media have been the topic of several studies on writing processes (Flower & Hayes, 1981; Haas, 1996; Van Waes & Schellens, 2003). These studies show that

the use of a specific writing medium often creates complementary perspectives to describe some of the cognitive processes that support writing in more detail². By observing the writing process in the speech recognition mode, we also hope to gain insight into some writing processes in general. Or as Haas stated:

As is often the case, new situations – in this case new technological contexts and situations for writing – bring to the fore aspects of writing that may have been there all along, but that have not been previously noticed. [...] – something with which writers have presumably been operating all along – becomes obvious in such a new situation or context: the changed technological context of composing with word processing. (Haas, 1996, p. 117)

The effects of the writing medium are wide ranging; the writing experience, the writing process, and the resulting texts are all influenced by the medium. Several studies have been conducted to compare writing processes in handwriting and in computer-based writing (see Van Waes & Schellens, 2003 for a review). These studies show that the writing medium has an effect on the writing process, especially on the planning, rereading and revision strategies. For example, the initial planning in computer-based writing is significantly less than in handwriting (Haas, 1989a). Most studies also show that use of a word processor tends rather to encourage revision of formal aspects (spelling, punctuation, etc.) and revisions below the level of the sentence (e.g. Bridwell & Duin, 1985; Daiute, 1986; Joram, Woodruff, Lindsay, & Bryson, 1990; Lee, 2002; Van Waes, 1991). The risk-taking behavior expected because of the ease with which the text on-screen can be manipulated (Haas, 1989b, 1996) does not, according to those studies, lead to a greater number of revisions at a higher level (revision of meaning). On the contrary, the greater attention paid to revision at lower levels (revision of form) apparently distracts the writer's attention away from the possibility of revision at higher levels. The studies cited also show that writers shift attention from the writing task to the medium. An aspect which might be expected in the speech recognition mode, because the interference of the writing medium is even bigger - especially in the beginning - than with word processing.

In the revised writing model of Hayes (1996) the composing medium arises in the task environment. In the former model (Flower & Hayes, 1980) the 'text produced so far' already played a significant role. This part of the task environment also influences our study because the writers are able to reread the text they have produced so far on the screen, contrary to what they normally do when dictating with classical dictating devices. The text remains unseen until the dictated text is typed out. With speech recognition the writer can constantly refer to the text, keep track of his progress, and revise already written text segments. This difference in writing medium can shed a light on the role of the physical environment in the organization of the writing process. Speech recognition can be an appropriate writing tool to gain insight in the role of the composing medium because of its hybrid characteristics.

Speech recognition can be described as a hybrid writing mode, because it is more

² When examining writing processes for dictating, writing patterns occur that differ from computer writing processes.

or less a combination of dictating and computer writing. Traditional dictating is characterized by a high degree of linearity in the production of text. Only few revisions are made. Sentences or phrases are dictated one after the other. The only revising usually taking place is a mental revision before the text is dictated to the recorder. In contradiction to the traditional dictating mode, writers using speech technology get immediate written feedback on the computer screen. This creates the possibility to review the text in all stages of the writing process, opening the gates to non-linearity. A high degree of non-linearity is typical of computer writing processes (Severinson Eklundh, 1994; Van Waes & Schellens, 2003). Most computer writers consider the paragraph, or even a sentence, as a unit that is planned, formulated, reviewed and revised in short recursive episodes (Van den Bergh & Rijlaarsdam, 1996). The constant feedback on the screen offers them the possibility to revise a lot, without losing the overview of the final text (Haas, 1989b; Honeycutt, 2003).

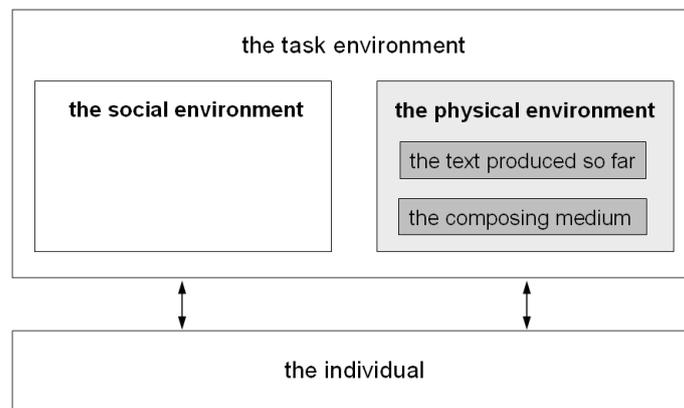


Figure 1. Synthesis of the writing model of Hayes (1996, p. 4).

Different components of the writing process require evaluation of the text produced so far. This is a very complex task, even more so if a text is not at all visible as it is in the dictating mode. But also in the computer mode text evaluation is difficult. A study by Wood et al. (2002) showed the importance of the physical availability of the text produced so far. They examined how academics use computer technology when writing academic papers. They suggested that hard copy probably offers cognitive support that may not be available in computer writing³ (Wood et al., 2002). And next to findings of Haas (1996), they show the advantages of capturing the ‘sense’ of a document. Computer technology might not sufficiently support the memory and organizational demands that the writers need to have of a text. These demands, however, differ for each writer. Consequently, we assume that writers with different

³ Wood et al. (2002) state that revisions are better traced in paper versions of a text. However, they do not mention the revising tools that are available in Word processors these days.

previous experience in writing business texts, may require different ways to facilitate their writing process.

2.1 Speech recognition and writing

With the term 'speech recognition' we refer in this article to the use of dictating software. Texts are written by talking to the computer. The speech recognition software just adds a toolbar to a normal working environment on screen. The programs are often called continuous speech recognizers. This means that the software has an input of fluent speech, and that it is not necessary to spell words discretely. By talking to the computer via a headset with a microphone, users see their text appear on the screen. Every action that used to be done by keyboard & mouse can be done with speech input; dictate text, navigate through programs or texts, apply functions like making words bold, and opening or closing files. Speech recognizers are speaker-dependent. The user has to train the software to teach the system his or her voice. Then the software adapts dynamically to the particular voice characteristics of the user. When a person first begins to use the speech software, the recognition and performance is still poor. The performance of the speech recognition software improves as usage and training time increase (Rodman, 1999).

Speech recognition enables writers to talk to the computer at a normal speaking rate. This could cause the production rate of these texts to be much higher, because we can speak faster than most of us can handle the keyboard to type texts. However, especially in the adaptation stage the text production is not faultless, because speech recognition systems always require (extensive) training. Furthermore, the errors that are caused by speech recognition are easy to miss, because most of the time they are spelt correctly (Honeycutt, 2003). A short sentence like 'this is a test' can be recognized by the speech recognizer as 'it is a test'. This sentence is of course not what the writer intended to write, but because of its grammatical correctness and its orthographic resemblance the 'misinterpretation' is easy to miss. This slows down the writing and especially the reviewing process, because writers have to be extra careful to avoid this kind of mistakes appearing in their final text.

Whereas most research in the field of speech technology has dealt with the technical improvement of speech recognition and the role of phonology, only a few studies have focused on the use of speech recognition as a writing tool (Halverson, Horn, Karat, & Karat, 1999; Karat, Halverson, Horn, & Karat, 1999). These studies were mainly focused on the usability of speech recognition as a writing mode. In the study of Karat et al. (1999), for instance, initial and professional users tested the usability of three speech recognition systems. The initial users were given two kinds of text-creating tasks. They had to perform one task with speech recognition, the other with keyboard & mouse. Results showed that users experienced a great deal of difficulty correcting errors with speech input and subjects tended to stay in the speech modality much longer during a correction than the researchers had expected. Perhaps this can

be explained by the goal of the study (usability and error correction) and the fact that the participants were observed all the time and had just received extensive training in speech based correction methods.

Also the use of speech recognition by the learning and physically disabled has received some attention. Quinlan (2002) studied the writing process and products of forty 11 to 14 year-old children. Twenty of these children experienced writing difficulties. The children composed a series of four narratives in four writing conditions, via handwriting and speech recognition, with advanced planning and without. Quinlan hypothesized that speech recognition provides cognitive benefits to children with writing difficulties and that advanced planning is supportive for the real-time planning process. Results showed that dysfluent writers produced longer texts when using speech recognition. Speech recognition seems to reduce transcription-related interference enabling children to produce more fluently written – and thus longer – texts. For the fluent writers, composing with speech recognition did not lead to improved narratives. Perhaps the writing process of fluent writers is relatively more interfered by the speech recognizer. For them, handwriting is relatively effortless, but speech recognition may represent a new source of interference. Advance planning had a significant positive effect on text quality for both groups.

Hartley et al. (2001) focused on differences between the writing products of keyboard & mouse and speech recognition. They compared the writing products produced in different writing technologies of three expert writers over a period of thirty years. Although the writing styles of the participants differed, the individual styles remained remarkably constant over time. In another study, Hartley et al. (2003) compared the writing products of two experienced writers, one using keyboard and the other using speech recognition. The writing products of both writers differed in sentence length, the number of long sentences and the use of the first pronoun.

In the relatively short history of speech recognition as a writing mode, no specific studies have been conducted on the effect of this new writing mode on the writing process. A study that perhaps relates most to this subject is by Reece & Cumming (1996) on the Listening Word Processor (LWP). In their experiment participants dictated to a hidden typist. This study had the benefit that the text appearing on the screen had no word-recognition errors, as opposed to the text that appears when working with a speech recognizer. In their experiment with 30 young children (10-12 years old) the authors found very promising results for the LWP. The children had to write a text in three different writing modes: handwriting, dictation and LWP. Texts written in the LWP condition were generally superior to those produced by the other two methods. Reece and Cumming state that the Listening Word Processor fostered a different composing process to that seen with dictation:

If long pauses are assumed to represent planning episodes, then there were indications that the LWP encouraged planning in a fashion not seen with dictating. We interpret this as follows: because the LWP provides the writer with a visual record of the text, the

writer has no need to allocate the substantial portion of working memory to maintaining some form of ongoing representation of the composition. Instead, writers can turn their attention to a consideration of higher level aspects of the composition task. (Reece & Cumming, 1996, p. 375)

2.2 Research questions

In our study we observed participants who worked with speech technology as a writing device for their (professional) texts for the first time. In a previous article (Leijten & Van Waes, 2006b, chapter 2) we focused on the adaptation and writing processes of two participants with different learning styles. In this study another important aspect is focused upon. Because the participants in the study were either familiar with word processing or classical dictating, we were able to thematize this dissimilarity in their initial writing experience. Therefore, the main research question in this follow-up study is whether previous writing experience (i.c. classical dictating vs. word processing) influences the adaptation process of learning to write with speech recognition? Subquestions are: How do the writers combine the different writing modes (speech, keyboard and mouse) in their writing process? Do they adapt their writing style to the new medium? Does their writing strategy and their organization of the writing process change over time?

To answer these questions, we had to develop a research method that enabled us to describe the writing processes of persons using speech technology in their writing activities, taking into account the specific characteristics of this writing mode. Therefore, the feasibility of the methodology itself is also an important aspect of this study. In other words, is the methodology we have developed suitable for the research objectives we put forward, both from a quantitative and a qualitative perspective?

Finally, we would like to emphasize that we are not only interested in particularities of writing with speech recognition. We hope that our research will also enable us to gain a clearer understanding of cognitive writing processes in general. We think that because of the characteristics of writing with speech technology, certain aspects of the writing process might be easier to observe and analyze than before. The interaction with the text produced so far, is certainly such a subprocess (cf. *infra*).

3 Description of the research project

In this section we describe the method we developed to reveal the elements that characterize the adaptation strategies of the writers involved in the study. First we give a description of the participants, and explain the design and procedure of the research study. Then the materials the participants produced are presented. Finally, we describe the selection of the data for the quantitative analysis and the case study.

3.1 Participants

The data we discuss in this study are taken from a larger set of observations we collected in the context of an NRI research project⁴. In this research project we observed the writing processes of the participants five times during a period of about one month. This resulted in 40 observation sessions of about 20 minutes (10 participants). We chose to observe two different groups of writers: lawyers and academics. Most of the lawyers that participated in the study were used to dictating their texts with classical dictating devices. The academics on the other hand were not familiar with dictating. This difference in writing preferences enabled us to take into account previous dictating experience in the description of the adaptation processes.

3.2 Design and procedure

Before we started the experiment the participants had to fill in a questionnaire. The questionnaire focused on information about the participants' prior knowledge of speech technology, their attitude towards writing with a computer, their learning strategies and writing profile (Levy & Ransdell, 1996; Sharples, 1999; Sharples & Pemberton, 1992; Van Waes & Schellens, 2003). It also addressed their experience in writing business texts and using a computer or a dictating system.

Before the participants started using speech recognition for the first time, they watched an introductory video about the use of the speech technology program⁵. The software company provided this video with each software package and advised users to watch this video before they started using speech recognition. The participants were then informed about the procedure of the study in more detail. They were asked to use the speech recognition system during their day-to-day work for about at least three hours a week. The participants could decide for themselves how to use the software and were not restricted to using speech input only. In total we observed the participants five times, after 1, 3, 6, 9 and 12 hours, while writing in their own environment. In the hours in between the participants worked for themselves with the speech recognition software.

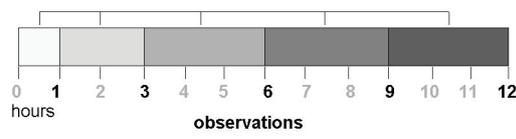


Figure 2. Timeline observation sessions.

⁴ NRI research project on the influence of speech recognition on the writing process (Research grant of the University of Antwerp 2000-2002).

⁵ We opted for a general video introduction because we did not want to influence the use of the technology in one way or another. We wanted to give the participants optimal opportunities to develop their own strategies on the basis of a technical introduction to the possibilities the program offers. Therefore, we did not provide an operational instruction session, as in the study of Karat et al. (1999). We think it is important not to direct the writing and correction behavior in a preconditioned way.

We collected the writing process data with an on-line screen camera⁶ (Camtasia™) and a sound recorder (Quickrecord™)⁷. The on-line screen camera did not interfere with the writing process of the writers. Apart from a small icon in the tray bar, the recorder was not visible when it recorded all the writer's actions. Because of the combination of the different input modes (keyboard, mouse and speech) we could not use existing logging programs. In addition to the digital observation, we also observed the participants on site during each writing session and took notes about specific writing circumstances that could not be registered otherwise (see Figure 3). These recordings and notes enabled us to reconstruct the writing process in detail.

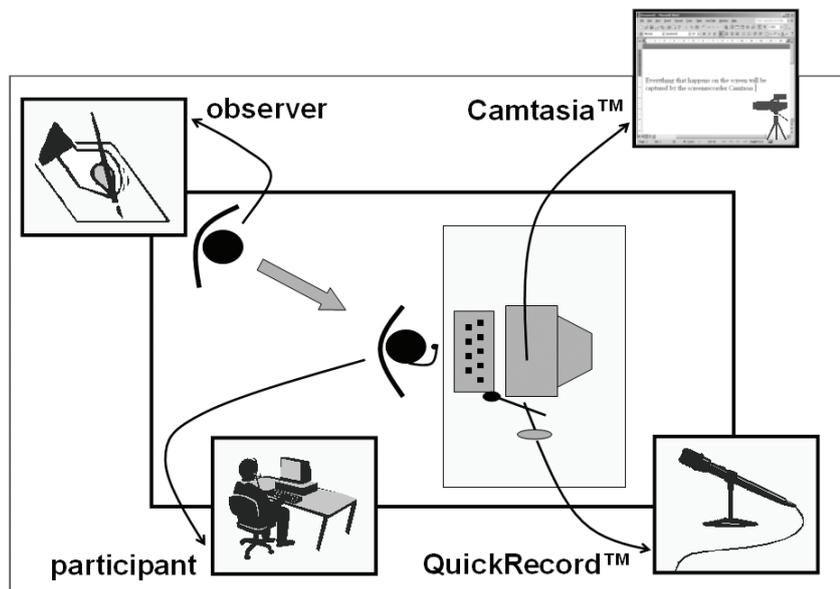


Figure 3. Observation setting.

Between the observation sessions the participants were asked to fill in a logging questionnaire after each task they had performed with speech recognition software. This logging questionnaire was mainly meant to track information about what tasks were performed and which specific functions of the program were explored. With this questionnaire we wanted to gather as much additional information as possible because the participants also worked with the program and developed new strategies between the different observation sessions.

⁶ Camtasia Recorder allows to capture cursor movements, menu selections, pop-up windows, layered windows, typing, and everything else that is visible on the computer screen.

⁷ Camtasia enables users to record audio in real time. We could not use this function, because the soundcard of the computer was already in use by the speech recognition software. Therefore, we opted for a separate sound recorder connected to a laptop. Subsequently we mixed the avi file and the wav file to one.

3.3 Materials

The different tasks the participants conducted were job-related and part of their normal writing activities, for example letters, e-mails or reports. During the observation sessions we collected both product and process data. The product data gave us information about the length of the texts (e.g. number of words, mean words per sentence), and the duration of the observation. The process data - collected by the on-line screen camera, the sound recorder, and the observer - provided us with data about the gradual development of the writing process (e.g. mode switches, pauses and revisions).

In total we observed the 20 writers for about 100 hours and they produced approximately 80 pages (about 24.000 words) of professional business texts.

3.4 Selection of data

For the analyses we divided the participants of the study in two groups: participants who had previous dictating experience and participants who did not. In this article we will describe a selection of the research data from two perspectives (see Table 1). In a quantitative study we will focus particularly on the use of the different writing modes. The results of this analysis will be used as reference data for a case study, in which we will compare the characteristics of the writing processes in the new writing mode of a 'non-dictator' with those of a 'dictator'. The data of the quantitative analysis will be used to further explore the influence of previous writing experience in more detail by relating them to the other process characteristics, in particular in relation to the repair, revision and pausing behavior. This approach should also allow us to better evaluate the feasibility of the methodology.

Table 1. Number of observation sessions in the quantitative study and case study

	+ Dictating experience	- Dictating experience
Quantitative study (10 participants * 5 sessions)	25	25
Case study (2 participants * 5 sessions)	5	5

The data of the participants that we selected for the quantitative study had to meet two criteria: the availability of the data⁸ and the production rate of the observation session. We selected the participants with the highest production rates. Moreover, because of the exploratory nature of the first observation session, we decided to eliminate the data of this observation from further analysis. We are aware that the first observation sessions provide a lot of relevant information about the adaptation process, but in our opinion these sessions are too different from the other sessions. In order to avoid too much interference of the results of these sessions we decided to eliminate these data. The sessions are to be seen as a first interaction with the speech

⁸ A few sessions could not be used because they were not complete (sound, video or both were not intact or of poor quality).

recognition system, not as an interaction with speech recognition as a writing tool. Our results are thus based on data from observations two to five. For the quantitative study this led to 20 sessions for each group (with previous dictating experience and without previous dictating experience). For technical reasons (quality of the recording) one observation in each group had to be removed, resulting in 19 observation sessions for each group. Finally, we selected two writers for the case study, one from each group. Because we focus on the difference in writing experience in this study we started from a lawyer who did not have previous dictating experience. In the group of experienced dictators we selected a lawyer with the same amount of working experience and the same learning style⁹.

4 Analysis

For the analysis of the data different methods were used. The analysis of the product data was aimed at contextualizing the writing activity. However, the focus of this study is on the process data. We will describe the adaptation process on three levels:

1. Current writing process related to previous writing experience;
2. Writing process in observation session 2 related to observation session 5;
3. Writing process in first half of the session related to second half of the session.

To describe the adaptation and writing process data, we developed a categorization and a transcription model that takes the complexity of the hybrid writing mode into account and makes the enormous amount of process data accessible for further research.

4.1 Categorization model

The categorization model was developed to describe and categorize the different aspects of the learning and writing process. The model takes the complexity of the hybrid writing mode into account and makes the enormous amount of process data accessible for further research. We developed the model while analyzing several observation sessions in detail in combination with existing literature on the different taxonomies for the analysis of writing processes (Hayes & Hayes, 1980; Karat et al., 1999; Severinson Eklundh, 1994; Van Waes, 1991). First, we will pay attention to the multimodal aspect of the writing process, or the use of different writing modes. This aspect will be studied in the quantitative study as well as in the case study. Then, we will describe the analyses that are used in the case study to further explore the characteristics of the writing processes observed. The problem-solving process of the participants using speech recognition is our first extra category. In this analysis we use

⁹The questionnaire contained a Dutch translation of Kolb's Learning Style Inventory (Kolb, 1984), which offered us information about differences in learning styles. We used this information to study the influence of learning styles on adaptation processes (Leijten & Van Waes, 2006b).

the term 'repair' to refer to technical problems and revisions that occur in the writing processes. In the next category we isolate the revisions from the other repairs that are made in the text. Finally, we describe the instrumentation we used to analyze the pausing behavior during the writing sessions.

4.1.1 Writing modes

Speech recognition software allows writers to conduct a writing session only with the use of speech. But, will writers choose this input as their main writing mode, or will they for example switch frequently to keyboard or mouse? To describe the multimodality, we calculated the number of mode switches and the time writers spend in a specific writing mode. We were also interested in the effect that mode switches have on the cognitive writing activity. The 'writer' can decide to change the writing mode at any time in the writing process. It was our expectation that especially when problems arise in the use of the speech technology or when the writer decides to revise the text produced so far, a switch from the speech mode to the keyboard or mouse mode could occur.

To analyze the mode switches we categorized each mode separately, but for the repairs we also combined the modes. We did not take the order of the combined modes into account:

- speech recognition,
- keyboard,
- mouse.

4.1.2 Repairs

The characteristic of classical dictating is that the 'text produced so far' is invisible and imposes a cognitive load on the writer. But using speech recognition and being able to see the text written so far may also have a negative effect on the cognitive load, because of incorrect text that sometimes appears on the screen due to technical shortcomings of the speech recognition software. It is our hypothesis that an incorrect representation of the text on the screen could also make it difficult for the writer to build a coherent representation of the text and will require a certain degree of cognitive accommodation.

The problem that the computer does not correctly 'hear' or 'understand' the input, and consequently takes action in a way that is different from the message spoken by the user. This phenomenon is comparable with an ordinary 'speaker-hearer' situation in which problems of misunderstanding also occur, for instance because of 'noise'. In that case the speaker or hearer of an utterance has to repair the problem in order to regain common understanding. In conversation analysis the concept of repairs is used to refer to a wide range of conversational phenomena, including correction of misunderstanding and mishearing, word recovery or self-editing (cf. Schegloff, Jefferson, & Sacks, 1977). In repairs we can distinguish self-repairs and other-repairs. Self-repairs refer to cases in which speakers repair errors or correct the appropriateness of their own speech. Other-repairs refer to the repairs of dysfluencies of the speaker by the

hearer. In both cases the speaker is the person who initiates the error or indistinct phrasing (cf. *supra*).

For speech recognition (or other human-computer interaction) we can add a third category: recognition repair (Stifelman, 1993). In this case the hearer, i.c. the speech recognizer, is more or less responsible for the error (see Figure 4). Of course, the user can also still be responsible for the error, for example because of a lack of knowledge about the speech recognizer (cf. *infra*).

		Responsible for error	
		Speaker	Hearer
Repairs error	Speaker	self-repair (revision)	recognition repair (technical problem)
	Hearer	other-repair	

Figure 4. Taxonomy of repairs (Stifelman, 1993, p. 3).

According to the contribution model of Clark and Schaeffer, a conversation consists of a series of presentations and acceptances. A participant of a conversation who is presenting an utterance wants to know whether his utterance has been accepted by the hearer. If we apply this theory to writing with speech recognition, we see that this is exactly what writers frequently (can) do when writing with speech recognition. After one or several utterances writers (can) search for evidence that their utterances have been accepted by the speech recognizer by rereading the text on the screen.

The concept of repairs – adapted from conversation analysis – is a valuable addition to the traditional concept of reviewing. It enabled us to incorporate a broader view of reviewing in this study, taking into account the specificity of the reviewing process in the speech recognition mode. Repairs are not limited to traditional revisions in the text (such as deletions and additions), but also refer to the correction of misrecognitions or technical problems.

Another aspect of repairs that is taken into account, is the distance of the repair in relation to the last point of utterance. It can either be done after one word or after several utterances. However, not every problem leads to a repair. Sometimes problems or flaws are ignored, for whatever reason (Mazeland, 2003). For this study we adopt the concept repairs in the context of writing with speech recognition. In order to do this, we focus only on the ‘provoker’ of the error. In this study repairs refer to both ‘technical problems’ and ‘revisions’. The technical problems are – in most cases – caused by the speech recognizer (hearer) and have to be solved by the user (speaker). The revisions are both initiated and conducted by the speaker.

In this section we will first describe the technical problems. We classified every speech recognition error and user strategy for repair. In section 4.1.2.2 we will discuss the categorization of revisions separately.

4.1.2.1 Technical problems

We categorized an event as a technical problem when it could be attributed to one of the following categories (taxonomy partially based on Karat et al., 1999):

1. single misrecognition (dictated text is misrecognized and a single word appears differently than was intended);
2. multiple misrecognition (dictated text is misrecognized and multiple words appear differently than intended);
3. command misrecognition (a command is misrecognized and the computer performs another action. For example, instead of executing the command 'end of the sentence' correctly, 'end of the sentence' appears on the screen);
4. dictated text as command;
5. hum or cough as text;
6. unaccountable errors;
7. no recognition (the computer does not respond);
8. computer crash.

Intention & outcome

As mentioned earlier in the text the speech input sometimes led to something different from what was intended by the writer. Therefore, we compared the intention of the writer with the actual outcome of the speech recognition system. We can distinguish errors at two levels: errors at the same level (word a = word b) and errors at a different level (word = symbol).

Intention	Outcome
word	word
symbol	symbol
function key	function key
navigation	navigation
no (intended) input	no output

The string 'word - word' is categorized as a misrecognition when the intended word differs from the outcome on the screen. For example the writer intended to formulate the word 'stay', but instead the word 'say' was recognized by the speech recognizer and represented on the screen. The string 'function key - word' means that a command has been misinterpreted and appeared in the text as a text segment. For instance, when the writer dictates the command 'end of the line' no cursor movement is realized, but the four words 'end of the line' appear as text on the screen.

Cause

We especially focused on technical speech recognition problems because we think that the way in which the participants deal with those problems and try to solve them reveals important aspects of the writing and learning process. To value the technical problems in the way in which the writers solve them, it is interesting to know what caused the (technical) problem. The problems can be either caused by the user, the

speech recognizer, or the environment. It is also possible that the cause cannot be logically explained.

1. user - ignorance (the user does not know how to perform an action);
2. user - inaccurate (the user knows how to perform an action, but does not perform that action accurately);
3. user - unknown (a problem caused by the user, other than 1 or 2);
4. environment - noise (the computer responds to a sound in the environment other than the writer, e.g. the sound of a ringing telephone);
5. software - ignorance (the computer is not trained sufficiently to know a word or a command);
6. software - illogical (the computer performs an action that is hard to explain, e.g. the computer makes a connection to the internet after the command 'underline that');
7. software - unknown (a problem caused by the computer, other than 5 or 6).

Number of attempts

We counted the number of attempts to solve the problem, because when using keyboard & mouse, you can solve a problem at once. This is not self-evident using speech technology. Repairs could either be done by keyboard & mouse or by speech input.

Writing mode

We also wanted to gather information about the mode the participants used to solve a problem. Do they hold to the same mode or do they switch writing modes to solve a problem? When writing with a classical dictating device, writers first produce the text and then make most corrections in a second phase. In this phase they probably use another mode, pen & paper or the word processor, to make these changes. Would they continue this pattern when writing in the speech recognition mode? Therefore we analyzed the three different writing modes (speech recognition, keyboard & mouse and all possible combinations between these modes, cf. infra).

Level & remoteness of correction

We categorized the levels of the corrections and distinguished between: character, word, sentence segment, sentence, paragraph and punctuation¹⁰. In addition, we compared the remoteness of the repairs in relation to the point of utterance, and distinguished the following categories: within the sentence, within the paragraph or outside the paragraph.

¹⁰ An extraordinary characteristic of speech recognition is that it is sometimes more time-consuming to correct just one character instead of a whole sentence or a sentence segment. When a writer wants to correct, for instance, the word 'eye' in the sentence 'Eye have been there', it is easier to dictate the complete sentence, than to navigate to the misinterpreted word, select it, change it, and reposition the cursor (all using speech).

Direction of correction

Because we were also interested in the distance of the corrections, we distinguished between backward and forward movements. Backward corrections start before the point of inscription and forward revisions occur in the text after the point of inscription.

Absolute time and interval

We noted the absolute time of the technical problems in the writing process and afterwards we calculated the position of the problems in the writing process by dividing the session into ten equal parts.

4.1.2.2 Revisions

In the repair analysis we distinguish technical problems from revisions. Revisions are changes in the text after an evaluation of the previously written text. These changes are not initiated by technical conflicts caused by the program but are meant to change the content, formulation or appearance of the text. Revision has already been studied over the years in different writing modes (e.g. pen and paper versus the word processor). Matsuhashi defines revision as follows:

A revision is an episode in which the writer stops the pen's forward movement and makes a change in the previously written text. The writer may then either resume the pen's forward movement or make another change at another location. (Matsuhashi, 1987, p. 208)

For writing with the word processor one could state we would have to replace pen with keyboard & mouse. It is obvious that the shift of writing with pen and paper to a word processor caused a shift in revising behavior (cf. review above). Therefore we are interested in the revising behavior of writers using speech recognition. Would this new writing medium also cause a shift in revision behavior?

Research about dictation as a writing method often focuses on text production. Texts written by dictating are composed more rapidly and are often longer. Dictation does not seem to affect text quality (for a description of research on dictating see Reece & Cumming, 1996). Studies do show that dictation is more appropriate for writing simple sequential text, rather than texts with a more complex structure. This is why writers do not revise very much when dictating and when they do revise, it is at a local level (Gould, 1978; Gould & Alfaro, 1984).

We categorized the revisions on four different levels: writing mode, level and remoteness of revision in written text produced so far, direction of revision, and absolute time and writing phase. Below we will give a full description of the different levels.

Writing mode

We categorized the three different writing modes, namely, speech recognition, keyboard & mouse and all possible combinations between these modes (cf. infra).

Level and remoteness of revision

We categorized the revisions on two levels: a first level in which the writer actually wants to revise, and another level in which the writer actually revised. We categorized the levels: character, word, sentence segment, sentence, paragraph and punctuation. In addition, we compared the remoteness of the repairs in relation to the point of utterance, and distinguished the following categories: within the sentence, within the paragraph or outside the paragraph.

Direction of revision

We distinguished between backward and forward revisions. Backward revisions start before the level of inscription and forward revisions occur in the text after the point of inscription.

Absolute time and writing phase

We calculated the absolute time of the revision in the writing process and afterwards we calculated the position of the revision in the writing process. Furthermore, we distinguished between revisions that were made within the first writing phase (first draft) and revisions that were made in the second writing phase and later.

4.1.3 Pauses

The analysis of pausing behavior is the last perspective we used to look at the data of the case study. Characteristics of the writers' pausing behavior enable us to gain a better insight in the cognitive planning processes. Or, to quote Matsuhashi (1982, p. 270):

Moments of scribal inactivity during writing reflect time for the writers to engage in cognitive planning and decision-making behavior. (...) [They] offer clues to cognitive planning processes during written discourse production.

Of course, we are aware that cognitive processes - might - also occur during speaking and writing. Because of the hybrid character of the speech recognition mode, we want to pay attention to pauses that are related to characteristics of the writing medium. Levelt states that the flow of speech is immediately interrupted when a problem is detected (Levelt, 1983). This phenomenon is also found in dictation processes (Schilperoord, 1996). Therefore, one might expect that this would also hold for speech input with speech recognition. Our hypothesis is that in writing processes in which speech input is used, pauses will also mark the choice of writing mode. Especially when writers are confronted with technical problems caused by the speech recognition software (cf. supra), and are forced to revise their texts technically, they seem to hesitate whether or not to switch from the speech mode to the keyboard or mouse mode. Because we observed initial users of the technology, the cognitive activity preceding this decision is probably more explicit and takes more time. We think that during the adaptation process the participants will develop new strategies to deal with technical problems and will be better able to anticipate (and avoid) this kind of problem. For the pause analyses we categorized the following elements.

Duration of pauses

Absolute length of every pause longer than 1 second¹¹.

Number of pauses

The total number of pauses longer than 1 second.

Temporal location of pauses

We identified the time stamp of every pause, and also attributed every pause to an interval by dividing each writing process in ten equal intervals. Intervals 1-5 and 6-10 are also referred to as 1st and 2nd half of the writing process respectively.

The above categories of writing modes, technical problems, revisions and pauses enabled us to describe the writing process data. But next to this we wanted to be able to illustrate the data by examples. Therefore, we had to develop a transcription model. This model will be described in the next paragraph.

4.2 Transcription model

We developed a (preliminary) transcription model to represent the writing process in a multi-layered linear representation. This model is partly based on the S-notation as developed by Kollberg (1998). The S-notation is a format for representing revisions during the writing process (Severinson Eklundh & Kollberg, 1992). An important goal of the S-notation, for this study, is that the notation should be non-redundant and consistent. In other words, it has to be possible to derive the written text from the notation in an unambiguous way.

As in the S-notation we use the following symbols for insertions and deletions: in the S-notation 'i' is the sequential number of the revised text. As we have showed in the categorization model, in our study we focus on technical problems and revisions. Therefore the 'i' in this study stands for sequential number of corrections of (technical) problems caused by the speech recognition software and the textual revisions made by the writer.

{inserted text} ⁱ	an insertion immediately following break i
[deleted text] ⁱ	a deletion immediately following break i
ⁱ	the break with sequential number i

In the S-notation all writing actions can be read in a single text layer. This is impossible for reporting a writing process with speech recognition. Therefore we had to develop a multi-layered transcription model. We also chose to give a linear representation of the text. In layer 1 the spoken or written input are transcribed, including textual

¹¹ We excluded very short pauses of < 1 second from our data, because we were not interested in the more physical pauses (Schilperoord, 1996).

commands. To represent these different types of input we used different fonts: sanserif for spoken input and serif for keyboard input. The command actions are indicated by tag brackets < >. Because the spoken input does not necessarily lead to a correct output we added a special layer in the model which shows what actually appears on the computer screen. This can be seen in layer 2. The third layer provides information about the writing modes used and the pausing times. Appropriate symbols indicate which writing mode is used and what movements the writers make through the text. To give an indication of the time in the writing process, time stamps are put in a different column (for an example of the notation model see Figure 6). Figure 5 gives a complete overview of the symbols used for the transcription model.

Notation system	
Spoken text	
<i>Typed text</i>	
i	start repair
/so\	pause time
Mode switch	
Indicated by the movement of the switch:	
☞	speech
⌨	keyboard
☞	mouse
☞	selection by mouse
a.b.c.	spelling mode
Movement by cursor	
←	cursor movement to the left
↑	cursor movement one line up
→	cursor movement to the right
↓	cursor movement one line down
☒	deletion with cursor to the left
☓	deletion with cursor to the right

Figure 5. Notation system for writing processes with speech recognition.

The example below (Figure 6) shows a short segment of a business letter (for the 'original text' see Figure 7). The writer starts in the speech recognition mode with the opening of the letter: 'dear mrs. baker'. The punctuation 'comma' and 'full stop' are also spoken. Therefore, these commands are also shown in the topmost row 'data of what the writer actually says'. Two line spaces are inserted with the voice command 'new line'. Then two sentences are dictated, with a short pause of 4 seconds after the first sentence. After a pause of 7 seconds the writer switches to a mouse-input to select the first misrecognition 'and least'.

With speech input the correct text 'employees who leave' is inserted. Now the text on the screen looks like 'Employees who leave leave'. Therefore the writer selects 'leave' by voice input and switches to keyboard to delete the selection and an extra interspace. The writer switches back to speech recognition to select the second misrecognition and corrects it to 'those'. Then he positions his mouse after the word 'employee'. After this word he types an 's' to make the word plural and he continues in the keyboard & mouse mode to delete the word 'sweeter'. To repair it to the correct word he switches again to speech input and dictates 'who leave the'. Finally he switches to the mouse mode to select the fourth misrecognition 'amply' and changes it with speech input to 'employee'.

In Figure 7 the final text of this short transcription is shown.

Dear Mrs. Baker,

Employees who leave the company with ten years of employment are entitled both to the company contributions and the retirement benefit payment deductions. Those employees who leave the company with less than ten years of employment will receive employee pay check contributions made to their retirement accounts.

Figure 7. 'Original' text of the transcription.

At the moment the elaboration of this short text segment is very time-consuming, because the transcriptions cannot be generated automatically. Consequently, the notation is merely meant to illustrate certain findings in the case study and not to describe full writing sessions. The linear representation of the notation also makes it difficult to interpret longer texts because of the recursive complexity.

5 Quantitative study: writing modes

For a quantitative analysis of the use of the different writing modes we selected 10 participants from our corpus: 5 participants with previous dictating experience (+D) and 5 without previous dictating experience (-D). Because the observation sessions differed somewhat in length, we present our analyses mainly in percentages, or per minute, in order to make the data more comparable.

The mode analysis shows that both groups use the possibility of speech recognition quite differently (Table 2). Writers with no previous dictating experience use significantly more speech than those with dictating experience (ANOVA $F(1, 36) = 6.30, p < .05$). Dictators use the speech input and the keyboard & mouse input respectively for about 60% and 40% of their writing time. The proportion of the speech mode drops gradually over time, from an average of 68% in the second observation session to about 56% in the final session. Non-dictators on the other hand, use the speech significantly more than dictators, that is for about 72% of their total writing time.

Table 2. Mean use of writing modes in percentages (4 observation sessions)

	+ Dictating experience		- Dictating experience	
	%	SD	%	SD
Speech (<i>n</i> = 19)	59.19		72.28	
Keyboard & mouse (<i>n</i> = 19)	40.81	20.90	27.72	8.93

If we compare the use of writing modes of the first half of the writing process with the second (Table 3), we notice that non-dictators prefer to write their texts mainly with speech input and that this choice of writing mode stays quite homogeneous over the two writing halves. Speech is their preferred writing mode, also in the writing phase where not formulating, but revising is an important subprocess. Dictators, on the other hand, use speech input in the first half of the writing process for about 65% of the time, but in the second half this amount drops to an average of 54%. In the second observation session dictators used speech on average for about 66% of the process time in the second half of the writing process. In the last session the speech input drops to 41%. Participants who have no previous dictating experience are more constant in their use of speech recognition. Speech input in this group remains quite constant (about 72%) in the different writing sessions and in both halves of the writing process.

Table 3. Mean proportion of speech mode in both halves of the writing process in percentages (4 observation sessions)

	+ Dictating experience		- Dictating experience	
	%	SD	%	SD
1st half	64.97	21.01	72.50	10.47
2nd half	53.99	26.40	71.93	12.34

Overall, participants without dictating experience use more speech than participants with dictating experience. However, in the first part of the writing session both groups still use a comparable amount of speech input (ANOVA $F(1, 36) = 1.95, p = > .05$). The difference between the groups arises in the second half. In that stage of the writing process the participants without dictating experience use significantly more speech than the group with dictating experience (ANOVA $F(1, 36) = 7.20, p < .05$).

The differences in the use of the speech mode are mainly determined by the beginning and the end of the writing process. As described above, we divided the writing sessions in 10 equal intervals. If we focus on the first and the last two intervals, we see that both groups use more speech in the first part than in the last part (GLM¹² $F(1, 36) = 6.48, p < .05$) of the writing process. The general observation that participants without previous dictating experience use significantly more speech than

¹² GLM = General Linear Model, Repeated Measures (i.e. a multivariate analysis with 'part' as a within-subjects variable and 'dictating experience' as a between-subjects variable).

participants with previous dictating experience also holds for these parts of the writing process (GLM $F(1, 36) = 5.37, p < .05$).

This is most explicit at the end of the writing process (intervals 9&10): dictators 46% versus non-dictators 65% ($t(36) = 2.34, p < .05$).

Table 4. Mean part of speech mode in both parts of the writing process in percentages (4 observation sessions)

	+ Dictating experience		- Dictating experience	
	%	SD	%	SD
1st part: interval 1-2	60.98	23.18	69.99	13.95
2nd part: interval 9-10	46.22	30.10	64.78	17.10

In Table 5 we present the data that describe the number of times the participants switch between the speech mode and the other writing modes (from speech to keyboard and from speech to mouse). The total number of switches over the four observation sessions does not differ for the two groups.

Table 5. Mean number of switches per minute from speech mode to the other writing modes during the writing process (4 observation sessions)

Speech > keyboard & mouse	+ Dictating experience		- Dictating experience	
	M	SD	M	SD
1st half	1.58	0.67	1.42	0.62
2nd half	1.15	0.50	1.42	0.69
Overall	1.35	0.50	1.41	0.52

Although the total number of switches from speech to another writing mode is comparable for both groups (ANOVA $F(1, 36) = 0.14, p = >.05$), their writing behavior over the four sessions seems to differ, revealing differences in the adaptation process. For the group with dictating experience the amount of switches remains quite constant over the different observation sessions, but for the group without dictating experience the number of switches rises from 1.19 switches per minute in session 2 to 1.61 switches per minute in session 5 (see Figure 8, top). To get a more detailed picture we divided the writing process in two equal halves again. The number of switches for inexperienced dictators increases parallel in the first half and the second half of the writing process. The number of switches for the experienced dictators drops considerably in the second half of the writing process ($t(18) = 2.87, p < .05$). This result can be partially explained by the drop in the use of speech input in the second half of the writing process by participants with dictating experience (Figure 8, bottom).

In sum, the choice for preferred writing mode differs for both experience groups. Participants without previous dictating experience use more speech than participants with previous dictating experience. This difference is to be found in the last half

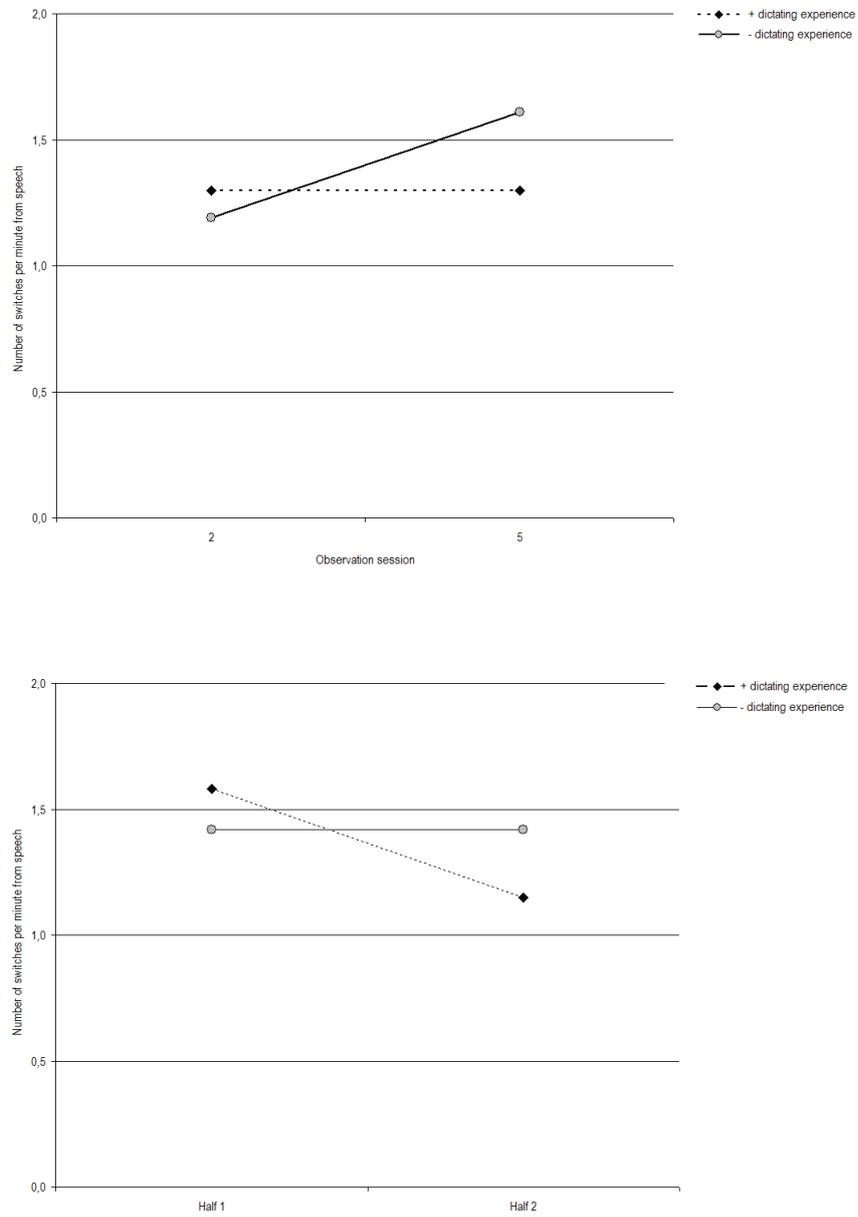


Figure 8. Number of switches per minute from speech in the first and second half (bottom) – in the second and fifth writing session (top).

and part of the writing process. Both writing groups are comparable for the amount of switches from speech to another writing mode, but the adaptation process that becomes visible when comparing the different observations and the two halves of the writing process, shows a different evolution. The group without dictating experience develops a pattern in which they steadily switch more between writing modes, both in the first and the second writing half. The group with dictating experience, on the other hand, switches less in the second writing half and also makes less use of the speech mode in that stage of the writing process. They hardly change this interaction pattern over time.

6 Case study

The results from the mode analysis reveal certain differences in the organization of the writing process and the use of the different writing modes between the group of dictators and non-dictators. Because these differences are sometimes difficult to interpret solely from the mode perspective, we present a case study in which the writing processes of two participants are described in more detail and from different perspectives. At the end of this analysis a short fragment of each participant's writing process is transcribed. In this transcription the data are illustrated on a more detailed and elaborated level, and can therefore function both as an introduction and as a summary to the case study as a whole.

In the case study we further explore the data, and compare the writing and adaptation processes of Frederik and Steven, two participants who differed in previous dictating experience and are representative for the groups described in the quantitative mode analysis. Frederik had 6 years of experience with classical dictating devices whereas Steven had no experience at all in dictating.

Frederik, associate in a large law firm, has 5 years of work experience and about 6 years of experience working with computers. He has a comparable experience in writing with classical dictating devices (dictaphone). In the questionnaire he described himself as not very patient when working with a computer. He also indicated that he does not really 'study' new computer programs, but that he does put extra effort and time into learning to work with them. Frederik expected that writing with speech recognition software would be easier and faster than his traditional way of writing texts (dictating combined with reviewing his text in a word processor with keyboard & mouse).

Steven also had 5 years of work experience as a lawyer and about 6 years of computer experience. He had no previous experience of writing with classical dictating devices. Steven reported in the questionnaire that he does not really like working with a computer, he is fairly impatient and gets irritated reasonably quickly. When he starts working with a new computer program he typically does not really take the time to study the possibilities, but he is patient enough to try things again. Steven's expectations on writing with speech recognition software were that it would be less tiring and more pleasant than writing with keyboard & mouse.

Both participants were also asked to describe their writing style for business texts (cf. design and procedure). Frederik indicated that he hardly plans his text in advance, but during the writing process he pauses shortly to plan the text. In the beginning of the writing process he plans a lot, but the further he progresses in the writing process, the less his writing process is fragmented by pauses. He rereads paragraphs and then decides whether or not to revise. Steven indicated that before he starts to write a text, he plans what he is going to write. He prefers a first-time final draft. During the writing process he does not make a lot of changes to the text produced so far. Revisions are made after the first draft. When asked to describe the relative time they devote to each of these different writing activities, they reported the percentages shown in Table 6.

Table 6. Percentage of time attributed to the different subprocesses in writing texts

	Frederik (+D)	Steven (-D)
	%	%
Planning	40	80
Writing	20	15
Revising	40	5

As described earlier, we observed five writing sessions of each participant. The observation sessions differed in length because the participants were observed while working on a task they had chosen themselves. The mean time of the observations in the case study is 22'55" (Frederik: $M = 21'06''$, Steven: $M = 24'33''$). In this time Frederik produced an average of 16.83 words per minute and Steven 13.51 words per minute. Because of the variable length of the observation sessions, the texts produced also differed in length. In the first observation the participants explored the program and wrote a very short text; during the subsequent observations their texts were between 247 and 449 words per session (Frederik: $M = 355$ words, Steven: $M = 330$ words). Table 7 gives a description of the five observation sessions.

Table 7. Summary details of the 5 observations

Observation session	Task	Length	Number of words in final text	Mean number of words per minute
Frederik (+D)				
1*	Letter	07'04"	92	13.07
2	Petition	21'58"	449	20.81
3	Conclusion	21'18"	419	19.78
4	Conclusion	21'48"	247	11.50
5	Letter	20'01"	305	15.24
Mean		21'06"	355	16.08

Steven (-D)				
1*	Letter + conclusion	30'47"	300	9.85
2	Letter	24'33"	404	16.61
3	Letter	21'53"	274	12.73
4	Letter + conclusion	27'27"	373	13.68
5	Petition	24'01"	267	11.12
Mean		25'33"	330	12.80

* Remark: Because of the exploratory nature of the first observation the data of the first observation session are eliminated from further analysis.

6.1 Writing modes

The mode analysis shows that both participants use the possibility of speech recognition quite differently (Table 8). Frederik uses the speech input and the keyboard & mouse input almost just as much. The part of the speech mode drops gradually from the second observation session (54%) to the final session (35%). Steven, on the other hand, uses the speech input almost twice as much as Frederik. For over 75% of his writing time, he uses the speech mode to construct his text. These results are in line with the results of the quantitative analyses reported above.

Table 8. Mean use of writing modes in percentages (4 observation sessions)

	Frederik (+D)		Steven (-D)	
	%	SD	%	SD
Speech	40.90	8.40	75.30	7.39
Keyboard & mouse	59.10		24.70	

If we compare the use of writing modes between the first half of the writing process with the second (Table 9), we notice that Steven prefers to write his texts with speech input and that this choice of writing mode remains quite homogeneous over the two writing parts. Speech is his preferred writing mode, also in the writing phase where not formulating, but revising is an important subprocess. Frederik, on the other hand, uses speech input in the first half of the writing process for about 50% of the time, but in the second half he clearly prefers writing with keyboard & mouse. The use of speech input to dictate text segments drops to a mean of 27.10%.

In the first observation session Frederik used speech for 34% of the process time in the second half of the writing process. In the last session, however, speech input was hardly used anymore in the last phase of the writing process. Frederik hardly uses it anymore to finish his text (cf. *infra*). In Steven's writing process, on the other hand, we hardly see any changes in the use of writing modes over the four sessions we observed. His use of speech recognition remains quite constant (75%) in the different writing sessions and in both halves of the writing process.

Table 9. Mean proportion of speech mode in both halves of the writing process in percentages (4 observation sessions)

	Frederik (+D)		Steven (-D)	
	%	SD	%	SD
Half 1	53.99	20.05	75.80	10.47
Half 2	27.10	20.00	73.92	12.30

If we zoom in on the beginning and the end of each writing process and isolate the data of the intervals 1-2 and 9-10, we see that the differences in use of speech are even greater. Both participants use more speech in the beginning (Frederik: $M = 62\%$, Steven: $M = 76\%$) of the writing process. Steven's use of speech at the end of the sessions still is 63% of the time, but Frederik prefers keyboard & mouse in 90% of his time in this writing phase.

In Figure 9 we present the data of the number of times the participants switch from the speech mode to the other writing modes. As in the quantitative mode analysis the average number of switches does not differ very much. They both switch from speech input to one of the other writing modes at least one time per minute (Frederik: 1.09 switches from speech per minute, Steven: 1.16 per minute).

The mean number of switches from speech to another writing mode is fairly comparable for both participants; the same goes for the distribution of switches over the two halves. Frederik switches more in the first half ($M = 1.42$) of the writing process, than Steven does ($M = 1.56$). The pattern in Frederik's switching behavior is consistent with the findings of the quantitative study; Steven's is a bit more divergent. The mean number of switches per minute is also somewhat lower for the participants of the case study compared to the average in the quantitative study (+dictating, $M = 1.35$ vs. -dictating, $M = 1.41$).

If we take a closer look at the evolution over the different writing sessions, we see that in the second session Frederik switches writing modes almost twice as much as Steven does (resp. 1.46 vs. 0.86). In the last session, however, Steven switches almost twice as much as Frederik does (resp. 1.12 vs. 0.60). If we compare this with the results from the quantitative study, the evolution of Steven's switching behavior shows to be consistent with those of the larger group of non-dictators. Frederik, on the other hand, displays a somewhat divergent pattern. His very modest use of the speech mode in the second half of the writing process is partly due to this (speech input was only used in 27% of the writing time). If we refine this to the last part of the writing process (interval 9 and 10), we see that his use of speech even drops to 9% at the end of the writing process.

The mode analysis of the case study to a large extent confirms - and strengthens - the findings of the quantitative study showing that dictators make less use of the speech mode while writing than non-dictators, especially in the second writing half. In the next paragraphs we will try to complement and interpret these findings by analyzing the repair, revision and pausing behavior of the two participants.

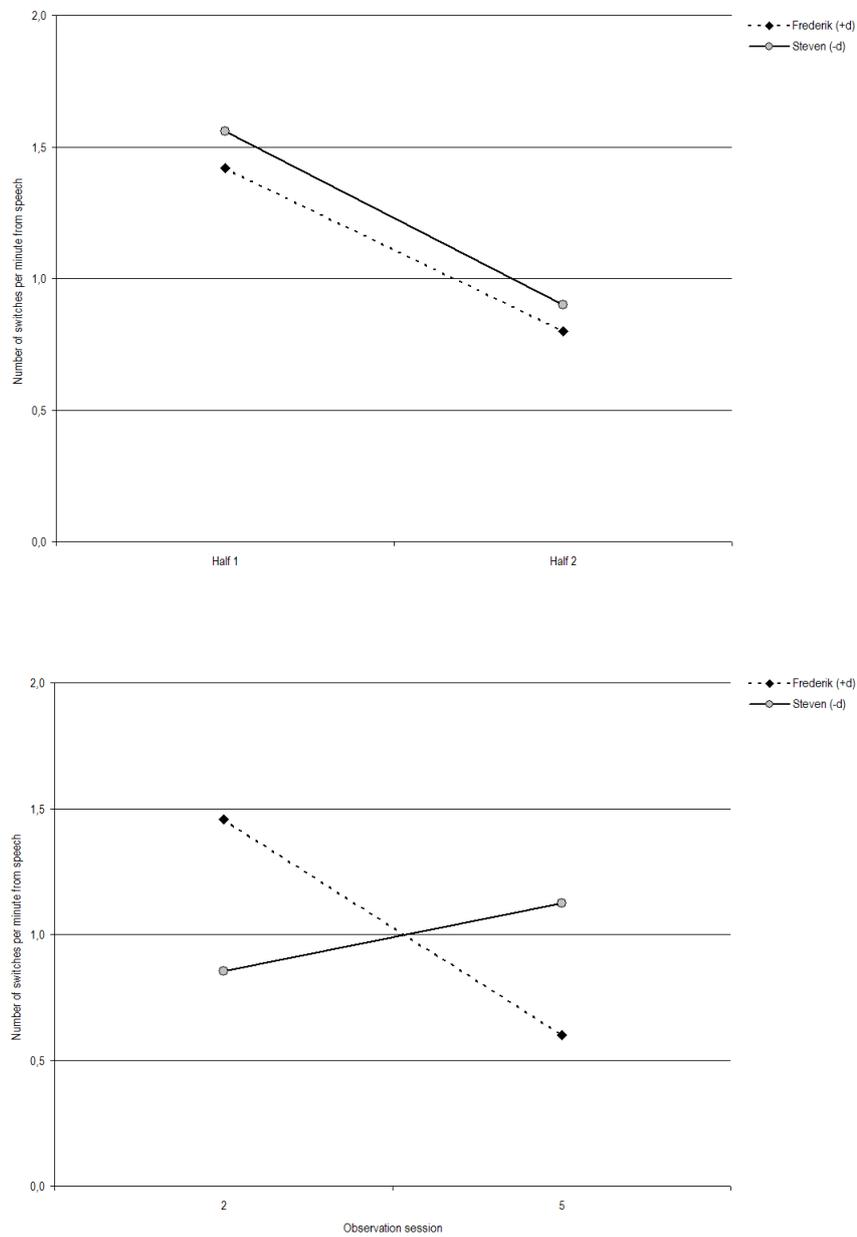


Figure 9. Number of switches per minute from speech in the first and second half (top) – in the second and fifth writing session (bottom).

6.2 Repairs

In this case study we coded 580 repairs in the four last sessions in which the participants either solved a technical problem in their text or made a revision (Frederik: $M = 72.50$ per session and Steven: $M = 72.45$ per session). In Table 10 the number of repairs per minute is presented. The table shows that Frederik executes 3.42 repairs per minute on average, while Steven carries out about 2.32 repairs per minute. This seems a lot, but these results are more or less in line with the findings of a study by (Levy & Ransdell, 2001) on writing and concurrent memory loads.

They state that presenting a concurrent task (more or less comparable to interruptions by technical problems) too frequently could lead to a situation in which the research participants devote their full attention to the secondary task, reducing the opportunities for proficient text generation. They determined that critical events for secondary tasks could occur with a maximum of about four times a minute which seems to be in line with our observation.

Table 10. Mean number of repairs, technical problems and revisions per minute and problems and revisions as a percentage of number of repairs (4 observation sessions)

Frederik (+D)	Repairs per minute	Technical problems per minute	Revisions per minute	Technical problems (% of repairs)	Revisions (% of repairs)
2	4.77	3.57	1.20	75	25
3	3.68	3.12	0.57	85	15
4	3.17	3.03	0.14	96	04
5	2.05	1.45	0.60	71	19
<i>M (SD)</i>	3.42 (1.13)	2.71 (0.92)	0.63 (0.44)	81 (11)	19 (11)
Steven (-D)					
2	2.06	1.40	0.66	68	32
3	2.74	2.46	0.28	90	10
4	2.53	2.20	0.38	87	13
5	1.96	1.33	0.62	68	32
<i>M (SD)</i>	2.32 (0.38)	1.85 (0.57)	0.47 (0.20)	78 (12)	22 (12)

About 80% of the repairs concerns problems that are caused by an inadequate response of the dictating software. The share of these problems in the total amount of repairs hardly drops throughout the sessions (see Table 10). Of course, we should keep in mind that the writers are novice users who are adapting to and experimenting with the new technology. Most of these misunderstandings concern misrecognitions of content dictated (Frederik: 66.9% vs. Steven: 44.1%). Steven's problem analysis also reveals a significantly higher percentage of functional and navigational misrecognitions (Frederik: 5% vs. Steven: 24.8%) giving extra support to the previous finding that he uses and explores the speech technology in a more varied and complete way than Frederik does.

If we further refine the analysis and take a look at the direction of the repairs (backward vs. forward, cf. *infra*) we notice that about 88% of Steven's repairs are backward movements in the text as opposed to 44% for Frederik. This proportion of backward and forward repairs remains quite stable over the different observation sessions too, with an exception for Frederik's fourth session. In this session he had to deal with severe technical problems (illogical problems, cf. *supra*) which he had to solve immediately, disturbing his normal way of writing. In other words, Frederik basically opts to repair and revise his text after a paragraph has been completed, or even waits until the first draft of the text is completed. This organization of the writing process corresponds to research findings that report a more linear development of the writing process of dictators in contrast to a more recursive development of the non-dictators' writing processes (cf. review above).

These findings can easily be related to the participants' uses of speech during the writing process (cf. mode analysis). Steven, as a non-dictator, prefers a first-time final draft and therefore he tries to repair errors in his text almost immediately, resulting in more backward and less forward repairs. Because he does not need to switch too often between writing modes for these different processes, his total number of switches is not very high. He does not wait until the end of the text to make these changes, resulting in a percentage of switches that is even higher in the first half than in the second half. Moreover, a percentage of the switches is not related to a switch between writing subprocesses. For instance, in the beginning of his texts he often has to write down several names and addresses, because he anticipates possible misrecognitions of the speech software. These events also seem to trigger mode switches.

6.3 Revisions

Table 10 shows that revisions take a 20% share of all the repairs. In absolute numbers, only a few revisions were made in every writing session (min. 3 – max. 26). Perhaps this is due to the large number of technical problems the writers were confronted with, as noted above. As a result they could have been distracted, causing an extra cognitive load. More than half of the revisions are situated at the sentence level, about 10% within the paragraph and about 35% beyond the paragraph level. So, contrary to the finding that revisions in computer-based writing are generally concerned with lower-level aspects of the texts (see Van Waes & Schellens, 2003 for a review), we observed quite a high percentage of higher-level revisions. This can be explained by the high proficiency of the writers, but also by the fact that most paragraphs were quite short (2 to 3 sentences).

Still it shows that the writers really do make use of the text on the screen, and do not only rely on their mental representation of the text produced so far - as is the case in traditional dictating environments. The analyses show that the revisions are not equally spread over the writing process.

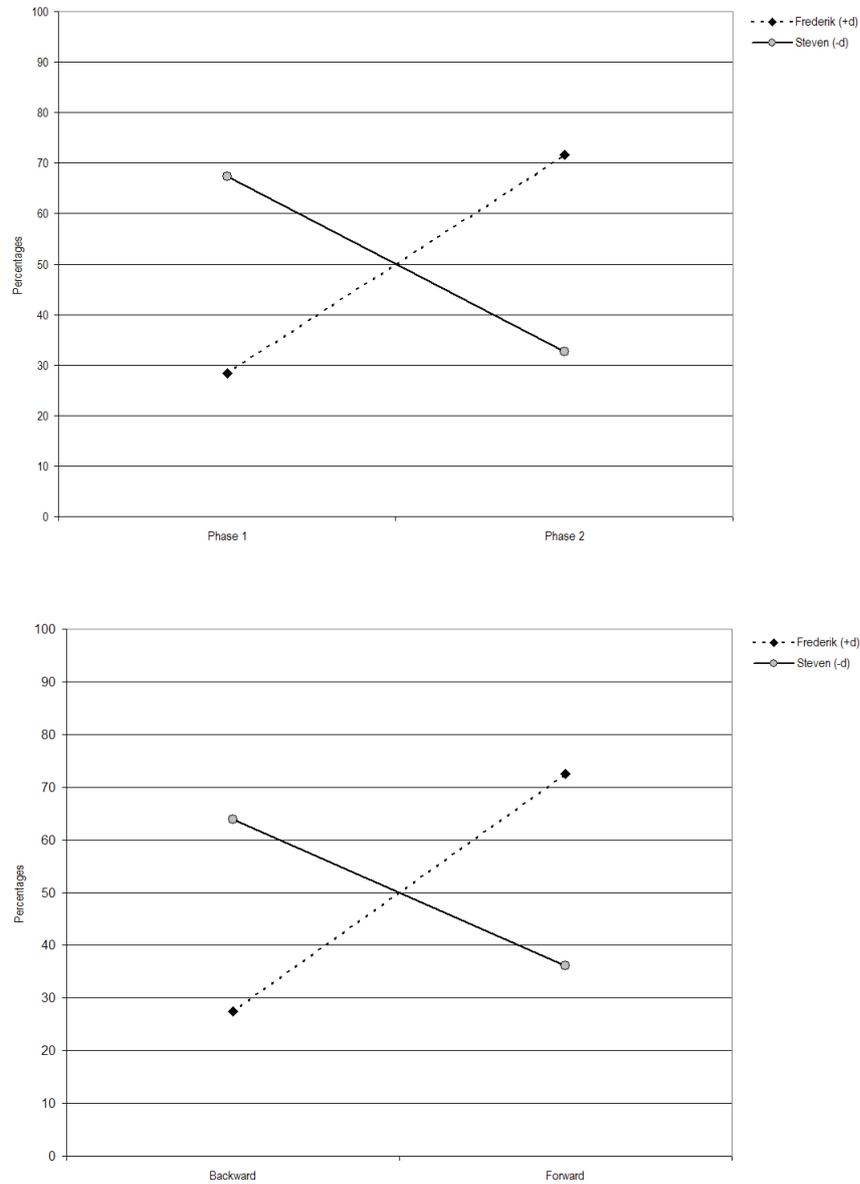


Figure 10. Distribution of the revisions over the writing phases (top) – direction of the revisions (bottom).

For instance, if we compare the number of revisions in the two writing phases (completion of first draft and later), we notice that in Frederik’s writing process 74% of the revisions are situated in the second phase. Steven, on the other hand, carries out only

32% of the revisions in the second draft of his texts, making two-thirds of his revisions in the first phase (see Figure 10, top).

The analysis of the remoteness of the revision in which backward and forward revisions are distinguished, confirms this observation: 73% of the revisions in Frederik's writing process are executed forward, as opposed to only 36% in Steven's (see Figure 10, bottom). This difference illustrates again that both writers actively use the text on their screen, but in different ways. Frederik prefers not to interrupt his formulating process too often and postpones his revision activity to the second phase. Steven, on the other hand, almost always revises his text immediately and then returns to formulating new texts. These strategies are in line with how they were used to writing their texts, dictating versus word processing.

6.4 Pauses

The characteristics of the two writers' pausing behavior enable us to gain a better insight into the cognitive planning processes. All pauses longer than 1 second were timed using the on-line video reconstruction of the writing process. In total 1465 pauses were registered: 495 pauses were located in Frederik's four writing sessions; 970 in Steven's session (see Table 11).

Table 11. Mean number and length of pauses over 4 observations

	Number of pauses		Length of pauses (s)	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Frederik (+D) (<i>n</i> = 495)	123.75	23.00	3.88	1.31
Steven (-D) (<i>n</i> = 970)	242.50	51.41	3.92	0.60

For both participants, the distribution of the pauses was quite homogeneous over the four writing sessions. A closer look at the data also shows a consistent distribution of pauses within every writing session. However, Frederik tended to pause more often in the first half of the writing session than in the second half (60% vs. 40%), whereas Steven spread his pauses over the process almost equally (52% vs. 48%).

We also calculated the mean length of the pauses. This analysis does not show any difference between the two participants: Frederik $M = 3.88s$ ($SD = 1.31$) - Steven $M = 3.92s$ ($SD = 0.60$). Frederik's total pausing time is 32% of the total production time; Steven's pausing time amounts to 62% of the production time, almost twice the percentage of the time Frederik uses for planning.

From these analyses we can conclude that the mean length of the pauses for both participants is comparable, but that Frederik's writing process is characterized by fewer pauses than Steven's, indicating a more fluent development of the text. To illustrate this, we calculated a fragmentation ratio representing the total amount of pauses divided by the total writing time. The higher this ratio, the higher the degree of fragmentation of the writing process. The fragmentation ratio of Frederik's writing

process is 9.4 and Steven's is almost double: 17.35. So, Steven interrupts his writing process almost twice as much as Frederik.

These results are in line with our expectations that Frederik, an experienced dictator, would adhere more explicitly to the habits he developed in the traditional dictating writing mode, whereas Steven would stick more to a fragmented writing process that is a typical profile for a writer using a word processor with keyboard & mouse only (Van Waes & Schellens, 2003). In describing the possible relationship between pauses and mode switches, we have indicated that pauses probably mark the choice of writing mode. If Steven's fragmented writing process were related to the choice of writing mode, we would expect that his writing process was characterized by many more mode switches than Frederik's. However, this was not the case.

The same holds for the relation between repairs and mode switches. In classical dictating and speech, the flow of speech is immediately interrupted when a problem is detected (Levelt, 1983; Schilperoord, 1996). We expected that especially when problems arise or when the writer decides to revise, a switch from speech to another writing mode would occur. If mode switches were directly related to the repairs, then Frederik would have switched more often from speech to another writing mode than Steven, because he made more repairs per minute than Steven. This was not the case either.

How else can we relate these observations from the case study to the difference in switching behavior between writers with previous dictating experience versus writers without previous dictating experience? The group of experienced dictators switched less from speech to another mode in the second half of the writing process.

If we combine this with the drop of speech input at the end of the writing process, we could state that this writing behavior resembles their behavior in classical dictating: first the text is dictated using a classical dictating machine, then it is handed over to a secretary who writes down the text with a word processor, and finally the text is revised using pen & paper or a word processor. This writing process is quite different from the writing process of writers who are used to writing their text with keyboard & mouse. Although the basis of their writing process is characterized by the same subprocesses, it is much more recursive and consists of short recursive cycles. Moreover, writers who use a word processor have the habit of completing a text in one writing mode. When writing with speech recognition they also try to complete the full writing process in one preferred writing mode, that is to say speech.

In the last part of this section we transcribe two short fragments as a detailed illustration of the analyses above.

6.5 Transcriptions

To make the results of the case study more profound, two short fragments of each participant's writing process are transcribed in this section. The notation system

described above was used to represent the different layers of the multimodal input and output. The first fragment in Figure 11 is from Frederik (observation session 2).

The original Dutch transcripts are presented in the Appendix.

0.00.16.12	For this reasons <new line> May it please the court <new line><new line> To hear the legal claim of the requestor to be declared admissible For this reasons J May it please the court JJJ To hear the legal claim of the requestor to be declared admissible
0.00.16.24	and legitimate <new line><new line> To conclude that this matter is not amenable to serious argument and therefore and legitimate JJJ The courtment ¹⁹ that this matter is not amenable to severe ⁶⁰ argument and therefore
0.00.16.33	needs to be dealt with at the introductory court session <new line><new line> Defendants joint <comma> at least in solidum <comma> needs to be dealt with at the introductory court session JJJ Defendants joint, at least insolidum ⁶¹ ,
0.00.16.46	one in absence of the other <comma> to see and hear sentence to pay the requestor the following sums one in absence of the other, to see and hear sentence two ⁶² pay the requestor the following sums
0.00.16.46	<colon><new line><new line><dash> as far as the repair concerns including the : JJJ - as far as the repair concerns including the ⁶³ ⁶⁴ ⁶⁵ ⁶⁶ ⁶⁷ ⁶⁸ ⁶⁹ ⁷⁰ ⁷¹ ⁷² ⁷³ ⁷⁴ ⁷⁵ ⁷⁶ ⁷⁷ ⁷⁸ ⁷⁹ ⁸⁰ ⁸¹ ⁸² ⁸³ ⁸⁴ ⁸⁵ ⁸⁶ ⁸⁷ ⁸⁸ ⁸⁹ ⁹⁰ ⁹¹ ⁹² ⁹³ ⁹⁴ ⁹⁵ ⁹⁶ ⁹⁷ ⁹⁸ ⁹⁹ ¹⁰⁰ ¹⁰¹ ¹⁰² ¹⁰³ ¹⁰⁴ ¹⁰⁵ ¹⁰⁶ ¹⁰⁷ ¹⁰⁸ ¹⁰⁹ ¹¹⁰ ¹¹¹ ¹¹² ¹¹³ ¹¹⁴ ¹¹⁵ ¹¹⁶ ¹¹⁷ ¹¹⁸ ¹¹⁹ ¹²⁰ ¹²¹ ¹²² ¹²³ ¹²⁴ ¹²⁵ ¹²⁶ ¹²⁷ ¹²⁸ ¹²⁹ ¹³⁰ ¹³¹ ¹³² ¹³³ ¹³⁴ ¹³⁵ ¹³⁶ ¹³⁷ ¹³⁸ ¹³⁹ ¹⁴⁰ ¹⁴¹ ¹⁴² ¹⁴³ ¹⁴⁴ ¹⁴⁵ ¹⁴⁶ ¹⁴⁷ ¹⁴⁸ ¹⁴⁹ ¹⁵⁰ ¹⁵¹ ¹⁵² ¹⁵³ ¹⁵⁴ ¹⁵⁵ ¹⁵⁶ ¹⁵⁷ ¹⁵⁸ ¹⁵⁹ ¹⁶⁰ ¹⁶¹ ¹⁶² ¹⁶³ ¹⁶⁴ ¹⁶⁵ ¹⁶⁶ ¹⁶⁷ ¹⁶⁸ ¹⁶⁹ ¹⁷⁰ ¹⁷¹ ¹⁷² ¹⁷³ ¹⁷⁴ ¹⁷⁵ ¹⁷⁶ ¹⁷⁷ ¹⁷⁸ ¹⁷⁹ ¹⁸⁰ ¹⁸¹ ¹⁸² ¹⁸³ ¹⁸⁴ ¹⁸⁵ ¹⁸⁶ ¹⁸⁷ ¹⁸⁸ ¹⁸⁹ ¹⁹⁰ ¹⁹¹ ¹⁹² ¹⁹³ ¹⁹⁴ ¹⁹⁵ ¹⁹⁶ ¹⁹⁷ ¹⁹⁸ ¹⁹⁹ ²⁰⁰ ²⁰¹ ²⁰² ²⁰³ ²⁰⁴ ²⁰⁵ ²⁰⁶ ²⁰⁷ ²⁰⁸ ²⁰⁹ ²¹⁰ ²¹¹ ²¹² ²¹³ ²¹⁴ ²¹⁵ ²¹⁶ ²¹⁷ ²¹⁸ ²¹⁹ ²²⁰ ²²¹ ²²² ²²³ ²²⁴ ²²⁵ ²²⁶ ²²⁷ ²²⁸ ²²⁹ ²³⁰ ²³¹ ²³² ²³³ ²³⁴ ²³⁵ ²³⁶ ²³⁷ ²³⁸ ²³⁹ ²⁴⁰ ²⁴¹ ²⁴² ²⁴³ ²⁴⁴ ²⁴⁵ ²⁴⁶ ²⁴⁷ ²⁴⁸ ²⁴⁹ ²⁵⁰ ²⁵¹ ²⁵² ²⁵³ ²⁵⁴ ²⁵⁵ ²⁵⁶ ²⁵⁷ ²⁵⁸ ²⁵⁹ ²⁶⁰ ²⁶¹ ²⁶² ²⁶³ ²⁶⁴ ²⁶⁵ ²⁶⁶ ²⁶⁷ ²⁶⁸ ²⁶⁹ ²⁷⁰ ²⁷¹ ²⁷² ²⁷³ ²⁷⁴ ²⁷⁵ ²⁷⁶ ²⁷⁷ ²⁷⁸ ²⁷⁹ ²⁸⁰ ²⁸¹ ²⁸² ²⁸³ ²⁸⁴ ²⁸⁵ ²⁸⁶ ²⁸⁷ ²⁸⁸ ²⁸⁹ ²⁹⁰ ²⁹¹ ²⁹² ²⁹³ ²⁹⁴ ²⁹⁵ ²⁹⁶ ²⁹⁷ ²⁹⁸ ²⁹⁹ ³⁰⁰ ³⁰¹ ³⁰² ³⁰³ ³⁰⁴ ³⁰⁵ ³⁰⁶ ³⁰⁷ ³⁰⁸ ³⁰⁹ ³¹⁰ ³¹¹ ³¹² ³¹³ ³¹⁴ ³¹⁵ ³¹⁶ ³¹⁷ ³¹⁸ ³¹⁹ ³²⁰ ³²¹ ³²² ³²³ ³²⁴ ³²⁵ ³²⁶ ³²⁷ ³²⁸ ³²⁹ ³³⁰ ³³¹ ³³² ³³³ ³³⁴ ³³⁵ ³³⁶ ³³⁷ ³³⁸ ³³⁹ ³⁴⁰ ³⁴¹ ³⁴² ³⁴³ ³⁴⁴ ³⁴⁵ ³⁴⁶ ³⁴⁷ ³⁴⁸ ³⁴⁹ ³⁵⁰ ³⁵¹ ³⁵² ³⁵³ ³⁵⁴ ³⁵⁵ ³⁵⁶ ³⁵⁷ ³⁵⁸ ³⁵⁹ ³⁶⁰ ³⁶¹ ³⁶² ³⁶³ ³⁶⁴ ³⁶⁵ ³⁶⁶ ³⁶⁷ ³⁶⁸ ³⁶⁹ ³⁷⁰ ³⁷¹ ³⁷² ³⁷³ ³⁷⁴ ³⁷⁵ ³⁷⁶ ³⁷⁷ ³⁷⁸ ³⁷⁹ ³⁸⁰ ³⁸¹ ³⁸² ³⁸³ ³⁸⁴ ³⁸⁵ ³⁸⁶ ³⁸⁷ ³⁸⁸ ³⁸⁹ ³⁹⁰ ³⁹¹ ³⁹² ³⁹³ ³⁹⁴ ³⁹⁵ ³⁹⁶ ³⁹⁷ ³⁹⁸ ³⁹⁹ ⁴⁰⁰ ⁴⁰¹ ⁴⁰² ⁴⁰³ ⁴⁰⁴ ⁴⁰⁵ ⁴⁰⁶ ⁴⁰⁷ ⁴⁰⁸ ⁴⁰⁹ ⁴¹⁰ ⁴¹¹ ⁴¹² ⁴¹³ ⁴¹⁴ ⁴¹⁵ ⁴¹⁶ ⁴¹⁷ ⁴¹⁸ ⁴¹⁹ ⁴²⁰ ⁴²¹ ⁴²² ⁴²³ ⁴²⁴ ⁴²⁵ ⁴²⁶ ⁴²⁷ ⁴²⁸ ⁴²⁹ ⁴³⁰ ⁴³¹ ⁴³² ⁴³³ ⁴³⁴ ⁴³⁵ ⁴³⁶ ⁴³⁷ ⁴³⁸ ⁴³⁹ ⁴⁴⁰ ⁴⁴¹ ⁴⁴² ⁴⁴³ ⁴⁴⁴ ⁴⁴⁵ ⁴⁴⁶ ⁴⁴⁷ ⁴⁴⁸ ⁴⁴⁹ ⁴⁵⁰ ⁴⁵¹ ⁴⁵² ⁴⁵³ ⁴⁵⁴ ⁴⁵⁵ ⁴⁵⁶ ⁴⁵⁷ ⁴⁵⁸ ⁴⁵⁹ ⁴⁶⁰ ⁴⁶¹ ⁴⁶² ⁴⁶³ ⁴⁶⁴ ⁴⁶⁵ ⁴⁶⁶ ⁴⁶⁷ ⁴⁶⁸ ⁴⁶⁹ ⁴⁷⁰ ⁴⁷¹ ⁴⁷² ⁴⁷³ ⁴⁷⁴ ⁴⁷⁵ ⁴⁷⁶ ⁴⁷⁷ ⁴⁷⁸ ⁴⁷⁹ ⁴⁸⁰ ⁴⁸¹ ⁴⁸² ⁴⁸³ ⁴⁸⁴ ⁴⁸⁵ ⁴⁸⁶ ⁴⁸⁷ ⁴⁸⁸ ⁴⁸⁹ ⁴⁹⁰ ⁴⁹¹ ⁴⁹² ⁴⁹³ ⁴⁹⁴ ⁴⁹⁵ ⁴⁹⁶ ⁴⁹⁷ ⁴⁹⁸ ⁴⁹⁹ ⁵⁰⁰ ⁵⁰¹ ⁵⁰² ⁵⁰³ ⁵⁰⁴ ⁵⁰⁵ ⁵⁰⁶ ⁵⁰⁷ ⁵⁰⁸ ⁵⁰⁹ ⁵¹⁰ ⁵¹¹ ⁵¹² ⁵¹³ ⁵¹⁴ ⁵¹⁵ ⁵¹⁶ ⁵¹⁷ ⁵¹⁸ ⁵¹⁹ ⁵²⁰ ⁵²¹ ⁵²² ⁵²³ ⁵²⁴ ⁵²⁵ ⁵²⁶ ⁵²⁷ ⁵²⁸ ⁵²⁹ ⁵³⁰ ⁵³¹ ⁵³² ⁵³³ ⁵³⁴ ⁵³⁵ ⁵³⁶ ⁵³⁷ ⁵³⁸ ⁵³⁹ ⁵⁴⁰ ⁵⁴¹ ⁵⁴² ⁵⁴³ ⁵⁴⁴ ⁵⁴⁵ ⁵⁴⁶ ⁵⁴⁷ ⁵⁴⁸ ⁵⁴⁹ ⁵⁵⁰ ⁵⁵¹ ⁵⁵² ⁵⁵³ ⁵⁵⁴ ⁵⁵⁵ ⁵⁵⁶ ⁵⁵⁷ ⁵⁵⁸ ⁵⁵⁹ ⁵⁶⁰ ⁵⁶¹ ⁵⁶² ⁵⁶³ ⁵⁶⁴ ⁵⁶⁵ ⁵⁶⁶ ⁵⁶⁷ ⁵⁶⁸ ⁵⁶⁹ ⁵⁷⁰ ⁵⁷¹ ⁵⁷² ⁵⁷³ ⁵⁷⁴ ⁵⁷⁵ ⁵⁷⁶ ⁵⁷⁷ ⁵⁷⁸ ⁵⁷⁹ ⁵⁸⁰ ⁵⁸¹ ⁵⁸² ⁵⁸³ ⁵⁸⁴ ⁵⁸⁵ ⁵⁸⁶ ⁵⁸⁷ ⁵⁸⁸ ⁵⁸⁹ ⁵⁹⁰ ⁵⁹¹ ⁵⁹² ⁵⁹³ ⁵⁹⁴ ⁵⁹⁵ ⁵⁹⁶ ⁵⁹⁷ ⁵⁹⁸ ⁵⁹⁹ ⁶⁰⁰ ⁶⁰¹ ⁶⁰² ⁶⁰³ ⁶⁰⁴ ⁶⁰⁵ ⁶⁰⁶ ⁶⁰⁷ ⁶⁰⁸ ⁶⁰⁹ ⁶¹⁰ ⁶¹¹ ⁶¹² ⁶¹³ ⁶¹⁴ ⁶¹⁵ ⁶¹⁶ ⁶¹⁷ ⁶¹⁸ ⁶¹⁹ ⁶²⁰ ⁶²¹ ⁶²² ⁶²³ ⁶²⁴ ⁶²⁵ ⁶²⁶ ⁶²⁷ ⁶²⁸ ⁶²⁹ ⁶³⁰ ⁶³¹ ⁶³² ⁶³³ ⁶³⁴ ⁶³⁵ ⁶³⁶ ⁶³⁷ ⁶³⁸ ⁶³⁹ ⁶⁴⁰ ⁶⁴¹ ⁶⁴² ⁶⁴³ ⁶⁴⁴ ⁶⁴⁵ ⁶⁴⁶ ⁶⁴⁷ ⁶⁴⁸ ⁶⁴⁹ ⁶⁵⁰ ⁶⁵¹ ⁶⁵² ⁶⁵³ ⁶⁵⁴ ⁶⁵⁵ ⁶⁵⁶ ⁶⁵⁷ ⁶⁵⁸ ⁶⁵⁹ ⁶⁶⁰ ⁶⁶¹ ⁶⁶² ⁶⁶³ ⁶⁶⁴ ⁶⁶⁵ ⁶⁶⁶ ⁶⁶⁷ ⁶⁶⁸ ⁶⁶⁹ ⁶⁷⁰ ⁶⁷¹ ⁶⁷² ⁶⁷³ ⁶⁷⁴ ⁶⁷⁵ ⁶⁷⁶ ⁶⁷⁷ ⁶⁷⁸ ⁶⁷⁹ ⁶⁸⁰ ⁶⁸¹ ⁶⁸² ⁶⁸³ ⁶⁸⁴ ⁶⁸⁵ ⁶⁸⁶ ⁶⁸⁷ ⁶⁸⁸ ⁶⁸⁹ ⁶⁹⁰ ⁶⁹¹ ⁶⁹² ⁶⁹³ ⁶⁹⁴ ⁶⁹⁵ ⁶⁹⁶ ⁶⁹⁷ ⁶⁹⁸ ⁶⁹⁹ ⁷⁰⁰ ⁷⁰¹ ⁷⁰² ⁷⁰³ ⁷⁰⁴ ⁷⁰⁵ ⁷⁰⁶ ⁷⁰⁷ ⁷⁰⁸ ⁷⁰⁹ ⁷¹⁰ ⁷¹¹ ⁷¹² ⁷¹³ ⁷¹⁴ ⁷¹⁵ ⁷¹⁶ ⁷¹⁷ ⁷¹⁸ ⁷¹⁹ ⁷²⁰ ⁷²¹ ⁷²² ⁷²³ ⁷²⁴ ⁷²⁵ ⁷²⁶ ⁷²⁷ ⁷²⁸ ⁷²⁹ ⁷³⁰ ⁷³¹ ⁷³² ⁷³³ ⁷³⁴ ⁷³⁵ ⁷³⁶ ⁷³⁷ ⁷³⁸ ⁷³⁹ ⁷⁴⁰ ⁷⁴¹ ⁷⁴² ⁷⁴³ ⁷⁴⁴ ⁷⁴⁵ ⁷⁴⁶ ⁷⁴⁷ ⁷⁴⁸ ⁷⁴⁹ ⁷⁵⁰ ⁷⁵¹ ⁷⁵² ⁷⁵³ ⁷⁵⁴ ⁷⁵⁵ ⁷⁵⁶ ⁷⁵⁷ ⁷⁵⁸ ⁷⁵⁹ ⁷⁶⁰ ⁷⁶¹ ⁷⁶² ⁷⁶³ ⁷⁶⁴ ⁷⁶⁵ ⁷⁶⁶ ⁷⁶⁷ ⁷⁶⁸ ⁷⁶⁹ ⁷⁷⁰ ⁷⁷¹ ⁷⁷² ⁷⁷³ ⁷⁷⁴ ⁷⁷⁵ ⁷⁷⁶ ⁷⁷⁷ ⁷⁷⁸ ⁷⁷⁹ ⁷⁸⁰ ⁷⁸¹ ⁷⁸² ⁷⁸³ ⁷⁸⁴ ⁷⁸⁵ ⁷⁸⁶ ⁷⁸⁷ ⁷⁸⁸ ⁷⁸⁹ ⁷⁹⁰ ⁷⁹¹ ⁷⁹² ⁷⁹³ ⁷⁹⁴ ⁷⁹⁵ ⁷⁹⁶ ⁷⁹⁷ ⁷⁹⁸ ⁷⁹⁹ ⁸⁰⁰ ⁸⁰¹ ⁸⁰² ⁸⁰³ ⁸⁰⁴ ⁸⁰⁵ ⁸⁰⁶ ⁸⁰⁷ ⁸⁰⁸ ⁸⁰⁹ ⁸¹⁰ ⁸¹¹ ⁸¹² ⁸¹³ ⁸¹⁴ ⁸¹⁵ ⁸¹⁶ ⁸¹⁷ ⁸¹⁸ ⁸¹⁹ ⁸²⁰ ⁸²¹ ⁸²² ⁸²³ ⁸²⁴ ⁸²⁵ ⁸²⁶ ⁸²⁷ ⁸²⁸ ⁸²⁹ ⁸³⁰ ⁸³¹ ⁸³² ⁸³³ ⁸³⁴ ⁸³⁵ ⁸³⁶ ⁸³⁷ ⁸³⁸ ⁸³⁹ ⁸⁴⁰ ⁸⁴¹ ⁸⁴² ⁸⁴³ ⁸⁴⁴ ⁸⁴⁵ ⁸⁴⁶ ⁸⁴⁷ ⁸⁴⁸ ⁸⁴⁹ ⁸⁵⁰ ⁸⁵¹ ⁸⁵² ⁸⁵³ ⁸⁵⁴ ⁸⁵⁵ ⁸⁵⁶ ⁸⁵⁷ ⁸⁵⁸ ⁸⁵⁹ ⁸⁶⁰ ⁸⁶¹ ⁸⁶² ⁸⁶³ ⁸⁶⁴ ⁸⁶⁵ ⁸⁶⁶ ⁸⁶⁷ ⁸⁶⁸ ⁸⁶⁹ ⁸⁷⁰ ⁸⁷¹ ⁸⁷² ⁸⁷³ ⁸⁷⁴ ⁸⁷⁵ ⁸⁷⁶ ⁸⁷⁷ ⁸⁷⁸ ⁸⁷⁹ ⁸⁸⁰ ⁸⁸¹ ⁸⁸² ⁸⁸³ ⁸⁸⁴ ⁸⁸⁵ ⁸⁸⁶ ⁸⁸⁷ ⁸⁸⁸ ⁸⁸⁹ ⁸⁹⁰ ⁸⁹¹ ⁸⁹² ⁸⁹³ ⁸⁹⁴ ⁸⁹⁵ ⁸⁹⁶ ⁸⁹⁷ ⁸⁹⁸ ⁸⁹⁹ ⁹⁰⁰ ⁹⁰¹ ⁹⁰² ⁹⁰³ ⁹⁰⁴ ⁹⁰⁵ ⁹⁰⁶ ⁹⁰⁷ ⁹⁰⁸ ⁹⁰⁹ ⁹¹⁰ ⁹¹¹ ⁹¹² ⁹¹³ ⁹¹⁴ ⁹¹⁵ ⁹¹⁶ ⁹¹⁷ ⁹¹⁸ ⁹¹⁹ ⁹²⁰ ⁹²¹ ⁹²² ⁹²³ ⁹²⁴ ⁹²⁵ ⁹²⁶ ⁹²⁷ ⁹²⁸ ⁹²⁹ ⁹³⁰ ⁹³¹ ⁹³² ⁹³³ ⁹³⁴ ⁹³⁵ ⁹³⁶ ⁹³⁷ ⁹³⁸ ⁹³⁹ ⁹⁴⁰ ⁹⁴¹ ⁹⁴² ⁹⁴³ ⁹⁴⁴ ⁹⁴⁵ ⁹⁴⁶ ⁹⁴⁷ ⁹⁴⁸ ⁹⁴⁹ ⁹⁵⁰ ⁹⁵¹ ⁹⁵² ⁹⁵³ ⁹⁵⁴ ⁹⁵⁵ ⁹⁵⁶ ⁹⁵⁷ ⁹⁵⁸ ⁹⁵⁹ ⁹⁶⁰ ⁹⁶¹ ⁹⁶² ⁹⁶³ ⁹⁶⁴ ⁹⁶⁵ ⁹⁶⁶ ⁹⁶⁷ ⁹⁶⁸ ⁹⁶⁹ ⁹⁷⁰ ⁹⁷¹ ⁹⁷² ⁹⁷³ ⁹⁷⁴ ⁹⁷⁵ ⁹⁷⁶ ⁹⁷⁷ ⁹⁷⁸ ⁹⁷⁹ ⁹⁸⁰ ⁹⁸¹ ⁹⁸² ⁹⁸³ ⁹⁸⁴ ⁹⁸⁵ ⁹⁸⁶ ⁹⁸⁷ ⁹⁸⁸ ⁹⁸⁹ ⁹⁹⁰ ⁹⁹¹ ⁹⁹² ⁹⁹³ ⁹⁹⁴ ⁹⁹⁵ ⁹⁹⁶ ⁹⁹⁷ ⁹⁹⁸ ⁹⁹⁹ ¹⁰⁰⁰ ¹⁰⁰¹ ¹⁰⁰² ¹⁰⁰³ ¹⁰⁰⁴ ¹⁰⁰⁵ ¹⁰⁰⁶ ¹⁰⁰⁷ ¹⁰⁰⁸ ¹⁰⁰⁹ ¹⁰¹⁰ ¹⁰¹¹ ¹⁰¹² ¹⁰¹³ ¹⁰¹⁴ ¹⁰¹⁵ ¹⁰¹⁶ ¹⁰¹⁷ ¹⁰¹⁸ ¹⁰¹⁹ ¹⁰²⁰ ¹⁰²¹ ¹⁰²² ¹⁰²³ ¹⁰²⁴ ¹⁰²⁵ ¹⁰²⁶ ¹⁰²⁷ ¹⁰²⁸ ¹⁰²⁹ ¹⁰³⁰ ¹⁰³¹ ¹⁰³² ¹⁰³³ ¹⁰³⁴ ¹⁰³⁵ ¹⁰³⁶ ¹⁰³⁷ ¹⁰³⁸ ¹⁰³⁹ ¹⁰⁴⁰ ¹⁰⁴¹ ¹⁰⁴² ¹⁰⁴³ ¹⁰⁴⁴ ¹⁰⁴⁵ ¹⁰⁴⁶ ¹⁰⁴⁷ ¹⁰⁴⁸ ¹⁰⁴⁹ ¹⁰⁵⁰ ¹⁰⁵¹ ¹⁰⁵² ¹⁰⁵³ ¹⁰⁵⁴ ¹⁰⁵⁵ ¹⁰⁵⁶ ¹⁰⁵⁷ ¹⁰⁵⁸ ¹⁰⁵⁹ ¹⁰⁶⁰ ¹⁰⁶¹ ¹⁰⁶² ¹⁰⁶³ ¹⁰⁶⁴ ¹⁰⁶⁵ ¹⁰⁶⁶ ¹⁰⁶⁷ ¹⁰⁶⁸ ¹⁰⁶⁹ ¹⁰⁷⁰ ¹⁰⁷¹ ¹⁰⁷² ¹⁰⁷³ ¹⁰⁷⁴ ¹⁰⁷⁵ ¹⁰⁷⁶ ¹⁰⁷⁷ ¹⁰⁷⁸ ¹⁰⁷⁹ ¹⁰⁸⁰ ¹⁰⁸¹ ¹⁰⁸² ¹⁰⁸³ ¹⁰⁸⁴ ¹⁰⁸⁵ ¹⁰⁸⁶ ¹⁰⁸⁷ ¹⁰⁸⁸ ¹⁰⁸⁹ ¹⁰⁹⁰ ¹⁰⁹¹ ¹⁰⁹² ¹⁰⁹³ ¹⁰⁹⁴ ¹⁰⁹⁵ ¹⁰⁹⁶ ¹⁰⁹⁷ ¹⁰⁹⁸ ¹⁰⁹⁹ ¹¹⁰⁰ ¹¹⁰¹ ¹¹⁰² ¹¹⁰³ ¹¹⁰⁴ ¹¹⁰⁵ ¹¹⁰⁶ ¹¹⁰⁷ ¹¹⁰⁸ ¹¹⁰⁹ ¹¹¹⁰ ¹¹¹¹ ¹¹¹² ¹¹¹³ ¹¹¹⁴ ¹¹¹⁵ ¹¹¹⁶ ¹¹¹⁷ ¹¹¹⁸ ¹¹¹⁹ ¹¹²⁰ ¹¹²¹ ¹¹²² ¹¹²³ ¹¹²⁴ ¹¹²⁵ ¹¹²⁶ ¹¹²⁷ ¹¹²⁸ ¹¹²⁹ ¹¹³⁰ ¹¹³¹ ¹¹³² ¹¹³³ ¹¹³⁴ ¹¹³⁵ ¹¹³⁶ ¹¹³⁷ ¹¹³⁸ ¹¹³⁹ ¹¹⁴⁰ ¹¹⁴¹ ¹¹⁴² ¹¹⁴³ ¹¹⁴⁴ ¹¹⁴⁵ ¹¹⁴⁶ ¹¹⁴⁷ ¹¹⁴⁸ ¹¹⁴⁹ ¹¹⁵⁰ ¹¹⁵¹ ¹¹⁵² ¹¹⁵³ ¹¹⁵⁴ ¹¹⁵⁵ ¹¹⁵⁶ ¹¹⁵⁷ ¹¹⁵⁸ ¹¹⁵⁹ ¹¹⁶⁰ ¹¹⁶¹ ¹¹⁶² ¹¹⁶³ ¹¹⁶⁴ ¹¹⁶⁵ ¹¹⁶⁶ ¹¹⁶⁷ ¹¹⁶⁸ ¹¹⁶⁹ ¹¹⁷⁰ ¹¹⁷¹ ¹¹⁷² ¹¹⁷³ ¹¹⁷⁴ ¹¹⁷⁵ ¹¹⁷⁶ ¹¹⁷⁷ ¹¹⁷⁸ ¹¹⁷⁹ ¹¹⁸⁰ ¹¹⁸¹ ¹¹⁸² ¹¹⁸³ ¹¹⁸⁴ ¹¹⁸⁵ ¹¹⁸⁶ ¹¹⁸⁷ ¹¹⁸⁸ ¹¹⁸⁹ ¹¹⁹⁰ ¹¹⁹¹ ¹¹⁹² ¹¹⁹³ ¹¹⁹⁴ ¹¹⁹⁵ ¹¹⁹⁶ ¹¹⁹⁷ ¹¹⁹⁸ ¹¹⁹⁹ ¹²⁰⁰ ¹²⁰¹ ¹²⁰² ¹²⁰³ ¹²⁰⁴ ¹²⁰⁵ ¹²⁰⁶ ¹²⁰⁷ ¹²⁰⁸ ¹²⁰⁹ ¹²¹⁰ ¹²¹¹ ¹²¹² ¹²¹³ ¹²¹⁴ ¹²¹⁵

Frederik does not interrupt his writing process very often. He prefers to dictate larger text segments at a rather high speed. He does not pause very often, nor does he interrupt his stream of production to make repairs or revisions. In this transcription he omits four errors (59-62) in his text. The fifth error (nr. 63) is corrected immediately but that does not trigger him to start correcting the previous errors. These errors are only corrected in the second phase of the writing process. Frederik prefers to correct most errors after having finished his first draft. He then returns to the beginning of the text and corrects most errors (59, 60, 62) by keyboard & mouse. Notable is that the errors that he does repair immediately are also repaired by keyboard & mouse. He took no notice of error 61.

For this reasons
May it please the court
To hear the legal claim of the requestor to be declared admissable and legitimate
To conclude that this matter is not amenable to serious argument and therefore needs to be dealt with at the introductory court session
Defendants joint, at least insolidum, one in absence of the other, to see and hear sentenced to pay the requestor the following sums:
- as far as the repair concerns including the forks: 1.117.872 BEF
....

Figure 12. Final text of fragment Frederik.

Steven's fragment is taken out of the fourth observation session.

Steven prefers a first time final draft. Therefore, he interrupts his writing process very often to pause and to make repairs. Because he repairs technical errors and revisions in his text immediately, a lot of his repairs are identified as backward movements. He prefers to stay in the speech mode to produce and to repair his text. Therefore, he has enough patience to try two to three times to correct the same error. For the word 'conviction' (error 18-20) he ultimately changes his text to remain in the speech mode.

In this transcription a remarkable strategy is used, that is more common in speech recognition. Instead of only repairing the error, a larger text segment is repaired. Error 16 is a good example of this strategy. Only the date 'June 26 1998' is misrecognized by the speech recognizer. But instead of repairing this date Steven repairs a larger text segment, probably because he anticipates this strategy to cost less effort.

0.05.14	Since by the first requestor a writ of summons was issued on June 26 1998 <dot> Since by the first requestor a writ of summons was issued on June 20 and not in 8 and 9. ¹⁶ /0.05.16-0.05.25 \ <undo ¹⁶ >
0.05.43	a writ of summons was issued on June 26 1998 1998 <dot> That October 7 1999 a conviction a writ of summons was issued on June 26 90 98 ¹⁷ '98. That October 7 '99 of action ¹⁸ <undo ¹⁷ >
0.06.21	a conviction a conviction revision ¹⁹ complex ion ²⁰ <undo ¹⁸ > .0.06.23-0.07.20 \ <undo ¹⁹ > <undo ²⁰ > .0.07.20-0.07.34
0.07.35	judgement was passed on with the second requestor being admitted for certain to the examination of witnesses judgements ²¹ was passed on with the 2nd requestor being admitted four certain ²² to the examination of witnesses <undo ²² >
0.07.57	<dot><new paragraph> In the meantime two decrees were deposited after the examination of witnesses by first requestor as well as +JJ In the meantime to ²³ decrees were deposited after the examination of witnesses by first requestor as well as . /0.08.00-0.08.12 \
0.08.35	by second requestor <dot><new paragraph> As a consequence the case is ready for and requestors ²⁴ by 2nd requestor. +JJ As a consequence the case is ready for and requestors ²⁴ <undo ²⁴ >
0.09.01	and a day in court can be set<dot><two lines down> and a day in court can be set. ++ / - / 0.10.53. ²¹ / 11.09. ²³

Figure 13. Transcription of fragment writing process Steven (4).

Since by the first requestor a writ of summons was issued on June 26 '98. That October 7 1999 judgement was passed on with the 2nd requestor being admitted to the examination of witnesses.

In the meantime two decrees were deposited after the examination of witnesses by first requestor as well ass by 2nd requestor.

As a consequence the case is ready for and a day in court can be set.

Figure 14. Final text of fragment Steven.

7 Conclusions

The question we raised in the introduction to this paper was whether the writers' previous writing experiences (word processing or dictating) would influence their adaptation to writing with speech recognition. Both the quantitative analysis and the case study reveal a difference in the use of the speech mode between the writers with previous dictating experience and those who do not have this experience. The mode analysis shows that dictators make less use of the speech mode than non-dictators, especially in the second writing half. Also the number of switches between writing modes for the experienced dictators drops considerably in the second half of the writing process.

The results of the pause analyses show that Frederik, an experienced dictator, adhered more explicitly to the habits he developed in the traditional dictating writing mode, whereas Steven stuck more to a fragmented writing process that is more typical for a writer using a word processor. Frederik's writing process is characterized by a mental planning in which relatively long text episodes are developed without being interrupted by any pauses. Frederik hardly ever uses the visual feedback on the screen in his planning activities and seems to rely heavily on his mental concept of the text produced so far. The problems of misunderstanding and mismatching that arise are often neglected at first - even when the text is quite seriously distorted - and postponed to a later writing phase in which he rereads and revises his text.

Steven's writing process, on the other hand, is very often interrupted, about twice as much as Frederik's. In the protocols we often see that sentences are formulated in chunks of two or three words with a small pause in between. Steven is also more influenced by the visual feedback of his dictating process on the screen and almost always corrects the mismatches immediately (percentage of repairs backwards; Steven = 88% and Frederik = 49%). This results in a high degree of recursivity, also typical for the dominant writing profiles of persons using a word processor.

The analyses of the revision data reinforce these findings. About three quarters of the revisions in Frederik's writing process are characterized as forward revisions. This means that he prefers not to interrupt his writing process too often to make changes to the text produced so far, but postpones the revisions to a rereading phase in which

he corrects his text top down. Steven's writing process can be characterized as highly recursive when compared to the more linear organization of Frederik's process. The larger number of backward revisions in Steven's process reveals a writing style in which revisions are often carried out immediately at the point of insertion. An extension of the remoteness analysis to all the repairs, confirms these findings.

To conclude we can say that the results of this case analysis suggest that the speech recognition mode creates a writing environment that is open for different writing styles. Both participants explicitly use the speech mode as an input device, but hold to their previous writing profile throughout the adaptation process and hardly seem to feel the need to deviate from the style that they reported in the questionnaire. Frederik maintained the writing habits he developed by using traditional dictating devices, while Steven holds to his word processing writing style and relies heavily on the visual feedback of his dictation that appears as typed text on the screen. In other words, the speech recognition mode itself does not seem to trigger writers to adapt a specific writing style, as opposed to what happens when writers have to adapt to write their texts in either the dictating or the word processor mode.

8 Discussion and further research

In this paper we have described writing data produced with speech recognition from a more quantitative and a more qualitative perspective. The data have been selected from a larger corpus of observations. The study confirms the potential hybrid character of speech recognition as a writing mode. One of the main differences between classical dictating and dictating with speech recognition is that the writer gets feedback on the screen almost immediately. The case analysis presented here also confirms the fact that the interaction with 'the text produced so far' is an important aspect in the organization of the writing process (cf. also Hayes' model). It certainly influences the writing process, but the extent to which writers use this feedback while dictating differs.

The interaction with the text on the screen can lead to a highly recursive writing process in which every error is repaired almost immediately, but it can also lead to a less recursive writing process in which repairs are made at the end of a paragraph or a text. In an experimental follow-up study we would like to further explore the repair behavior of the participants and more specifically the interaction with the (defective) 'text produced so far'. We would like to identify different characteristics of the (defective) 'text produced so far', and select material from our corpus as input data. In a controlled experiment we would like to present this material to participants in a writing task, in order to be better able to describe the interaction with the 'text produced so far', and evaluate the cognitive load caused by defective representations of it (see also chapter 5 & 6).

The detailed description of the case study shows that the chosen observation instruments and analyzing methods enable us to analyze and describe the specific

aspects of the writing processes in which speech recognition was used as a writing device. But, as stated above, the analyses of the data were very time-consuming. Therefore, we are developing a more sophisticated observation instrument called 'Inputlog' (Leijten & Van Waes, 2005a), which would make the analyses of several categories faster and more accurate¹³. Inputlog is a logging tool that enables researchers to record and consequently analyze writing processes at different levels (i.e. text analyses, mode analysis, and pause analysis). For a full description see Leijten and Van Waes (2006a, also chapter 8). Inputlog will make it easier to describe the data and will make it possible to analyze the data more precisely on a larger scale.

This brings us to another element in our study. As described above, the participants in this study were free to write whatever text they were planning, varying from a short routine letter to a complex summons. We preferred letting the people work in their own environment and let them develop their own learning and writing strategy. The advantage of this approach is the ecological validity of the data observed. However, this resulted in a wide range of text types and genres, differing for instance in length and structure. Therefore, it would be interesting to compare the writing modes speech recognition versus keyboard & mouse or the classical dictating devices in a more controlled way. The study described here will form a solid ground to set up such a controlled experimental study in which we will observe writers while writing texts on the basis of a well-defined task in different writing modes.

Finally, we would like to mention that the purpose of this research project is not only to focus on the speech recognition as a technical device. The analyses in this study show that the focus on a new writing mode also enables us to gain a better insight into the specific aspects of cognitive processes that are responsible for the development of writing processes in general. This line of approach will be even more central in our further research. We hope to report on this in the near future.

Notes

This study is part of an NRI research project on the influence of speech recognition on the writing process (Research grant of the University of Antwerp 2000-2002).

Acknowledgements

We would like to thank all the people that participated in this study, and especially Frederik and Steven, for their willingness to learn to work with speech recognition and let us observe them while doing so. We are also grateful to Lernout & Hauspie (now Nuance) who gave us a free license to work with VoiceXpress Legal™ during the course of this study.

¹³ Existing logging-tools as Trace-it (Severinson Eklundh & Kollberg, 1996) or ScriptLog (Strömquist & Malmsten, 1997) are either developed for a specific environment, or not adequately adapted to the current Windows environment. A combination with the speech mode also is not possible in the current tools.

We would also like to thank Ivy Ackerman for preparing the data of the quantitative study on writing modes, Sarah Ransdell and the anonymous reviewers for their constructive comments, and Kim Sleurs for proofreading this article.

Appendix

Transcript of Frederik's fragment (original transcript in Dutch)

0.00.16.12	<p>Om deze redenen <nieuwe lijn> Behage het de rechtbank <nieuwe lijn><nieuwe lijn> De vordering van verzoekerstervankelijk en gegrond te Om deze redenen JJ Behage het de rechtbank JJ De vordering van verzoekerstervankelijk en gegrond te</p>
0.00.16.24	<p>horen verklaren <nieuwe lijn><nieuwe lijn> Vast te stellen dat deze aangelegenheid niet voor ernstige betwisting vatbaar is en derhalve horen verklaren JJ Assen JS stellen dat deze aangelegenheid niet voor vereiste 60 betwisting vatbaar is en derhalve</p>
0.00.16.33	<p>op de inleidingszitting dient te worden behandeld <nieuwe lijn><nieuwe lijn> Gedaagden solidaar <komma> minstens in solidum <komma> op de inleidingszitting dient te worden behandeld JJ Gedaagden solidaar, minstens insolidum 61,</p>
0.00.16.46	<p>de ene bij gebreke aan de andere <komma> te zien en te horen veroordelen om te betalen aan verzoekerstervolgende sommen de ene bij gebreke aan de andere <komma> te zien en te horen veroordelen om te betalen ad 62 verzoekerstervolgende sommen</p>
0.00.16.46	<p><dubbelpunt><nieuwe lijn><nieuwe lijn><gedachtenstreepte> voor wat betreft herstelling met inbegrip van de : JJ gedachtenstreepte 63 - voor wat betreft herstelling met inbegrip van de 64 / 0.17.00.17.16 \</p>
0.00.17.18	<p>vorken <dubbelpunt> 1.117.872 Belgische Frank vorken: 1.117. 008 en feestje gebruiksvreemde 64 872 BEF 64 ctrl+shuft+ 65 / - - / 66 / 0.25.38 \ 67 / 0.25.42 \ 68</p>

4

Repair strategies in writing with speech recognition

The effect of experience with classical dictating

Abstract: This chapter describes the repair strategies of writers who have started using speech recognition systems for writing business texts. The writers differed in their previous writing experience. They either had classical dictating experience or they were used to writing their texts with a word processor. The study confirms the potential hybrid character of speech recognition as a writing mode. One of the main differences between classical dictating and dictating with speech recognition is that the writer gets the dictated text displayed on the screen almost immediately, making it directly accessible in a word processor. As previous research has shown, the interaction with ‘the text produced so far’ is a crucial aspect in the organization of the writing process. The case study presented here also lends support to this idea. The extent to which the developing text influences the writing process may depend upon the extent to which writers make use of it while dictating. The writers’ interaction with the text produced so far, is also influenced by errors that occur in the representation of the dictated text, for instance due to (technical) misrecognition by the software. In the case study described in this chapter, we observed that the interaction with the imperfect text on the screen can either lead to a highly recursive writing process in which every error is repaired almost immediately, or it can also lead to a less recursive writing process in which repairs are made at the end of a section or a first draft. Speech recognition seems to create a writing environment that is open for different writing styles.

This chapter has been published as Leijten, M., & Van Waes, L. (2006). Repair strategies in writing with speech recognition: The effect of experience with classical dictating. In L. Van Waes, M. Leijten & C. Neuwirth (Vol. Eds.) Writing and Digital Media (Vol. 17, pp. 31-46). Oxford: Elsevier.

Keywords: Speech recognition, writing processes, dictating, adaptation processes, research method, writing modes, writing experience, writing profiles, errors, repairs, revisions.

1 Introduction

‘See what you say’ was one of the slogans that was used to promote speech recognition software for dictating purposes in the early years of this technology. Indeed, the fact that the technology enabled writers to quickly produce a visual representation of the spoken text, combined benefits of dictating machines and word processors. The software released in the mid 1980’s did not quite fulfill this promise. However, twenty years later, speech recognition systems have improved dramatically (especially versions adapted for specific professions like law and medicine have a very acceptable speech to text conversion level, delivering up to 99% accuracy, after the program has been carefully trained). As is the case with other digital media, speech recognition software is still improving. Consequently, it should be kept in mind that studying the use of this software and its implications for the writing process research will be further influenced by this evolution. However, even in this stage of the technical developments the new writing mode enables us to gain insight in aspects of the writing process that are difficult to isolate in other writing modes.

The use of speech recognition in writing is extensively described by MacArthur (2006) and Quinlan (2006). They also give an overall review of the growing body of research on writing and speech technology, with special attention to studies on writers with learning disabilities.

In the present study we would like to present a case study in which two writers were observed during their first weeks in using speech technology for their day-to-day writing tasks. Both writers were lawyers: one was experienced in classical dictating (using a dictaphone); the other had no dictating experience at all. The focus of this study is on the way the writers interact with the dictated text that appears on their computer screen when dictating to the computer. More specifically, we observed the interaction with the ‘imperfect’ text produced so far. Due to (technical) misrecognition of the spoken text – especially in the adaptation phase – the text on screen is not always a perfect representation of what has been dictated. This imperfect text produced so far might cause an extra cognitive load and might also distract the writers during the planning of new text.

1.1 Speech recognition and writing

Looking back in time, dictation and writing seem to have a ‘cyclic relation’ with each other. That is certainly a conclusion one can draw when reading Honeycutt’s (2004) analysis of dictation as a composing method in Western history. In his description of the long history of dictation, he clearly shows “how dictation’s shifting role as a form

of literacy has been influenced by the dual mediation of technological tools and existing cultural practices" (Honeycutt, 2003, p. 294). By traveling through the history he shows that slight changes in the material conditions of reading and writing radically changed the relationship between author and text. The introduction of speech recognition as a writing tool for dictation practices might change this relationship once again. In contrast to more recent dictation practices in which there was a distinct gap between composition and written production, this new technological implementation brings together the oral and the written representation of the text. This characteristic of the new writing context created by voice recognition is the central starting point of this chapter. For a more detailed description of speech recognition and text-to-speech we would like to refer to Quinlan (2006) and MacArthur (2006).

1.2 The text produced so far and writing media

Since speech recognition brings together the oral and the written representation of text, we would like to describe speech recognition as a hybrid writing mode. It combines characteristics of both classical dictating and keyboard based word processing. Especially the new characteristics of the 'text produced so far' play a crucial role in writing with speech recognition. We think speech recognition has three particular points of interest related to the interaction with the text produced so far: visibility of the text produced so far, moment of visual feedback, and the correctness of the visual representation. Classical dictating does not create any immediate visualization of the text produced so far. Related to this, previous studies show that classical dictating is characterized by a high degree of linearity in the text production (Schilperoord, 1996). Writers dictate sentences or phrases one after the other and only few revisions are made. Revising almost exclusively takes place mentally before the text is dictated to the recorder. Keyboard based word processing on the other hand is characterized by immediate visual feedback of the text produced so far. Therefore, the computer writing process is typically characterized by a high degree of non-linearity (Severinson Eklundh, 1994; Van Waes & Schellens, 2003). Most computer writers consider a paragraph, or even a sentence, as a unit that is planned, formulated, reviewed and revised in short recursive episodes (Van den Bergh & Rijlaarsdam, 1996). The constant feedback on the screen offers them the opportunity to revise a lot, without losing the overview of the final text (Haas, 1989; Honeycutt, 2003). In word processing, tools like cutting and pasting also facilitate revising.

In contrast to the traditional dictating mode, writers using speech technology get immediate written feedback on the computer screen. This previously mentioned technical characteristic creates the possibility to review the text in all stages of the writing process either by speech or by keyboard, opening the gates to non-linearity. However, this written feedback, other than in keyboard based word processing and typed output of classical dictating, can contain misrecognitions that are of a different kind than ordinary typing errors.

As the writing model in Figure 1 shows, the text produced so far is an important component of the task environment in monitoring the writing process and it is closely related to the composing medium¹.

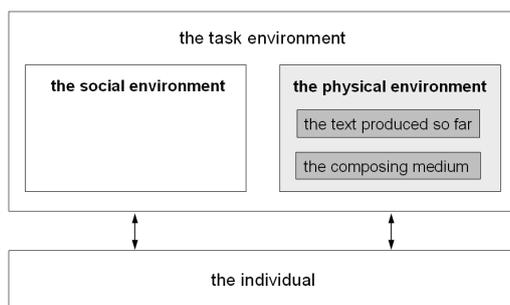


Figure 1. Synthesis of the writing model of Hayes (1996, p. 4).

The evaluation of the text produced so far is a very complex task, both in keyboard based word processing and in classical dictating. Studies of Wood, Willoughby, Specht and Porter (2002) and Haas (1996) have shown the importance of the physical availability and the 'sense' of the text produced so far. In their opinion computer technologies might not sufficiently support the memory and organizational demands that writers need. These cognitive demands to keep the 'sense' of the text, however, differ for each writer. Consequently, we assume that writers with different previous experience in the use of writing modes may require different ways to facilitate their writing process.

In our research project we observed twenty professional writers who worked with speech technology as a writing device for the first time. This group was divided into two groups: a group with previous classical dictating experience and one without previous classical dictating experience. The participants were all observed when writing in their own professional context. In other articles (Leijten, 2007; Leijten & Van Waes, 2003b; 2005b, also chapter 2 and 3), we describe two case studies that focus on the adaptation, learning, and writing processes of these initial speech recognition users. We tried to answer the question whether the writers' previous writing experiences (keyboard based word processing vs. classical dictating) would influence their adaptation to writing with speech recognition. The very detailed analyses of the case studies showed that the writer with previous classical dictating experience adhered more explicitly to the habits developed in the traditional dictating writing mode, whereas the writer without previous dictating experience stuck more to a fragmented writing process typical for a writer using a word processor. Speech recognition seems

¹ In our opinion the TPSF component is more complex than the writing model of Hayes (1996) claims. In chapter 7 we describe the interaction between the physical and mental representation with the text produced so far in greater detail (cf. Figure 3).

to create a writing environment that is open for different writing styles. Both writers explicitly use the speech mode as an input device, but hold to their previous writing profile throughout the adaptation process and hardly seem to deviate from the style that they reported in a questionnaire on writing profiles (Leijten & Van Waes, 2005b). In other words, the speech recognition mode itself does not seem to trigger writers to adapt a specific writing style, as opposed to what happens when writers have to adapt to write their texts in either the dictating or the word processor mode.

An additional quantitative study (Leijten & Van Waes, 2005b, see also chapter 3) that focused on the use of writing modes (speech versus keyboard & mouse) showed that writers with previous dictating experience make less use of the speech mode than writers without previous dictating experience. Especially in the second half of the writing process the use of speech input decreases for the experienced group. We could not find an explanation for these results solely based on these studies. Why do experienced dictators not use speech as much, although they already have experience in this way of writing? Is the text produced so far on the screen - and especially the misrecognitions - too distracting for them compared to their previous writing experience? Instead of a (correct) mental representation of the text produced so far, the writers are now faced with an (imperfect) visualized text on the screen. Therefore, a possible explanation for the lower percentage of speech input for experienced dictators could be that, based on their experience, they prefer to correct the errors in their text in the keyboard & mouse mode. For classical dictators the speech mode might be primarily associated with the production of new text and the keyboard & mouse mode with the revision of the first draft. This would seem a possible explanation since we see that also the number of switches or shifts from 'speech input' to 'non-speech input' drops considerably in the second half of the writing process for the experienced dictators. To support this reasoning, we describe a case study in this chapter in which we explore the writing strategies related to repairs, and describe the relation between the use of writing modes and the repair behavior.

In the preceding section we have situated the present study in the broader context of previous and related research. In section two, we present the research questions. In the next section we shortly describe the research project this study is based on and present the categorization model we developed to analyze the 'repair behavior'² of the writers. In the repair analysis we study the way in which the participants deal with technical problems and revisions while writing. In section four, we give a detailed analysis of the case study in which the writing process of a participant with previous (classical) dictating experience is compared with a participant who did not have this experience. The participants are both lawyers with comparable working and computer experience. The characteristics of their writing process are described from the

² We have adapted the notion of 'repair' from research in the field of conversation analysis. Among other things, it is used to refer to corrections of misunderstanding and mishearing in natural speech (Schegloff, Jefferson, & Sacks, 1977).

different perspectives presented in the categorization model. Conclusions are drawn in section five, followed by a discussion of the broader perspective of the research project and related to further research.

2 Research questions

As a sequel to the previous studies on the influence of speech recognition on writing processes we would like to focus this study on the repair strategy of initial speech recognition users. We would like to explain why experienced dictators make less use of the speech mode than non-dictators. Because the participants in the study were either familiar with word processing and classical dictating or only with word processing, we were able to take up this dissimilarity in their initial writing experience.

The main research question of this study is: What is the effect of errors in 'the text produced so far' on the reviewing strategy of novice speech recognition users? Subquestions are: How are repairs distributed over the writing process? What is the effect of repairs on the monitoring of writing modes? When do writers (prefer to) repair an error during the writing process? To answer these questions, we have developed a categorization model that has enabled us to describe the specific characteristics of the errors that could occur in speech recognition based writing.

3 Description of the case study

In this section we describe the categorization model we developed to reveal the repair strategies of the writers involved in the study. First, we give a description of the participants and the writing tasks that they were involved in. Then, we explain the design and procedure of the research study (for a full description see Leijten & Van Waes, 2005b; chapter 3).

3.1 Participants and writing tasks

The two participants in this case study, Frederik and Steven, were selected from the larger group of participants that took part in the research project described above. Because the focus in this study is on the difference in writing experience, we selected, on the one hand, a lawyer who did not have any previous dictating experience, and on the other hand a lawyer with the same amount of working experience and the same learning style³ (Kolb, 1984), but with significant experience in classical dictating. The different tasks the participants conducted were job-related and were part of their normal writing activities, for example letters, e-mails or reports. During the observation sessions we collected both product and process data. The product data gave us information about the length of the texts (e.g. number of words, mean

³ According to Kolb's taxonomy, both Frederik and Steven are accommodators. They scored high on active experimentation and concrete experience. Kolb's model states that people with an accommodative orientation tend to solve problems in a trial-and-error manner.

words per sentence), and the duration of the observation. The process data provided us with information about the gradual development of the writing process.

3.2 Design and procedure

Because the participants were novice users of speech recognition software, they first watched an introductory video – provided by the software company – about the use of the speech technology program.

The participants were asked to use the speech recognition system during their day-to-day work for about at least three hours a week. They could decide for themselves how to use the software and were not restricted to using speech input only. In total we observed the participants five times, after 1, 3, 6, 9 and 12 hours, while writing in their own environment. In the hours in between the participants worked for themselves with the speech recognition software.

We collected the writing process data with an on-line screen camera (Camtasia™) and a sound recorder (Quickrecord™). The on-line screen camera did not interfere with the writing process of the writers. Apart from a small icon in the tray bar, the recorder was not visible when it recorded all the writer's actions. Because of the combination of the different input modes (keyboard, mouse and speech) we could not use existing key logging programs⁴.

3.3 Categorization model: repairs

To describe the repair process data, we developed a categorization model (Appendix) that takes the complexity of the hybrid writing mode into account and makes the enormous amount of process data accessible for further research.

The model is based on earlier taxonomies for the analysis of writing processes (Faigley & Witte, 1981; Hayes & Hayes, 1980; Karat, Halverson, Horn, & Karat, 1999; Lindgren & Sullivan, 2006; Severinson Eklundh, 1994; Van Waes, 1991), and is complemented with new categories that were observed while analyzing several recorded sessions in detail. Activities related to the problem-solving process of the participants using speech recognition, are the most important additions to the existing taxonomies. This category is referred to as 'repair' behavior. We use the term 'repair' to refer to the recursive actions writers perform when dealing with technical problems (caused by using keyboard & mouse and speech input) and revisions that occur during the writing processes (for a more detailed description of the model, see Leijten & Van Waes, 2005b; chapter 3).

⁴ At this moment we are developing a logging tool that can log speech input (more information on www.inputlog.net, see also Chapter 8).

4 Case study

The participants, Frederik and Steven, were observed while writing different types of business texts during their day-to-day work. Therefore, their observation sessions differed somewhat in length. For this study, we eliminated the data of the first observation session (after one hour), because of the exploratory character. The mean time of the observations in the case study is 21'59" (Frederik: $M = 19'01''$, Steven: $M = 24'57''$). During the observed writing sessions Frederik produced an average of 19 words per minute and Steven 13 words per minute⁵. Because of the variable length of the observation sessions, the texts also differed in length. Their texts were between 267 and 449 words per session (Frederik: $M = 355$ words; Steven: $M = 330$ words).

4.1 General

In this case study we coded 575 repairs⁶ in the four observation sessions in which the participants either solved a technical problem in their text or made a revision (Frederik: $M = 76.25$ per session and Steven: $M = 67.50$ per session). A technical problem can be labeled as a misrecognition when dictated text is misrecognized and words appears differently than was intended. When a command is misrecognized and the computer performs another action the technical problem is called a command misrecognition. For example, instead of executing the command 'end of the sentence' correctly, 'end of the sentence' appears on the screen.

In table 1 the number of repairs per minute is presented. The table shows that Frederik executes 4.02 repairs per minute on average, while Steven carries out about 2.70 repairs per minute.

Table 1. Mean number of repairs, problems and revisions per minute

	Frederik (+D)		Steven (-D)	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Repairs per minute	4.02	0.28	2.70	0.60
Technical problems per minute	2.89	0.72	2.04	0.70
Revisions per minute	1.13	0.60	0.66	0.14

Of all repairs, technical problems represent 70% and revisions about 30%. In absolute numbers, only a few revisions were made in every writing session (min. 10 – max. 34). Perhaps this is due to the large number of technical problems the writers encountered. As a result, these errors could have distracted the writers.

⁵ Feng, Karat & Sears (2005) report the following data in their study: novices composed a text with speech recognition at a rate of 8 words per minute, and completed a similar task using a keyboard and mouse at a speed of about 19 words per minute. By way of comparison, in natural language production people speak on average about 125-150 words per minute.

⁶ Some numbers might differ from previous studies because the categorization model in this article is more fine-grained than in Leijten & Van Waes (2003; 2005).

Table 2 shows that Frederik has to deal with different types of technical problems than Steven. Whereas Steven has a lot of command misrecognitions to repair (44.4%), Frederik has a lot of ‘other’ technical problems. Frederik had to deal with many unaccountable errors caused by the software. This might be (partly) explained by the fact that Frederik – in contrast with Steven – does not actually ‘train’ the software by repeating words that are misrecognized. Frederik prefers to correct errors as they appear and not to prevent them from happening again, whereas Steven anticipates on certain errors and adapts his writing behavior to the speech technology.

Table 2. Type of technical problems in percentages

Technical problems	Frederik (+D)	Steven (-D)
	% (n = 221)	% (n = 205)
Misrecognitions	62.4	51.3
Command misrecognitions	12.7	44.4
Dictated text as command	5.4	0
Other	19.5	4.4

More than 60% of the repairs (technical problems and revisions) are situated at the sentence level, which is quite a high percentage of higher-level repairs. Previous analyses of the reviewing behavior showed that in general writers who write on a computer are more concerned with lower-level aspects of the texts (Van Waes & Schellens, 2003). Frederik treats both types of repair similarly. He solves about 75% of the technical problems and the revisions within the paragraph, and 25% outside the paragraph. Steven seems to adapt his strategy to the type of repair. He solves 93.1% of his technical problems within the paragraph. For revisions this percentage is 63.6. Also the goal of the revisions differs for both participants. Frederik mostly revises to make formal changes and to correct typing errors (64.3%), whereas Steven's revisions are for 69.7% content based.

4.2 Distribution of the repairs over the writing process

If we take a look at the distribution of the repairs over the writing process, we notice that Frederik carries out more repairs in the second half of the writing process than in the first half, 67% versus 33%. Steven, on the other hand, performs 56% of his repairs in the first half and 44% in the second half of the writing process.

If we divide the writing processes of Frederik and Steven in five intervals (see Figure 2), a more clearly defined pattern emerges. Frederik's writing process is characterized by some repairs in the beginning, but he performs most of his repairs at the end of his writing sessions. Steven, on the other hand, begins his writing session with a large amount of repairs. We can explain this difference by referring to Steven's strategy to dictate also personal names, addresses, postcodes and numbers (cf. supra). This causes a lot of speech recognition problems, which he prefers to repair with speech immediately, in order to be able to use them later on again. In the middle of his writing process, some repairs occur, but reaching the end the amount of repairs

increases again, mostly because of his intensive revising of his text in the final stage of the writing session.

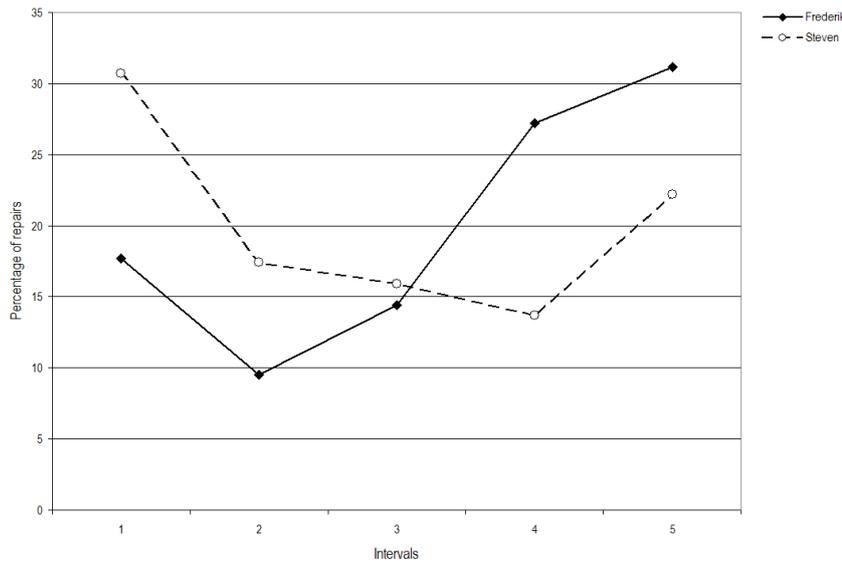


Figure 2. Percentage of repairs per interval (5).

This pattern is affirmed by the number of attempts to repair an error. Frederik solved 88% of his repairs in the first attempt compared to Steven's 66%. In 26% of the repairs Steven is willing to try two to three times. Frederik will try two to three times in only 10% of his repairs. The repair strategy of Frederik is more static than Steven's. The example in Figure 3 illustrates Frederik's repair behavior. In this fragment he is confronted with a misrecognition. To repair it, he switches from speech recognition to keyboard and mouse - using shortcuts.

0:10:10	Now she has agreed in a contract with buyer with her buyer	that before the 20 th of September
	Now she has agreed in a contract wit buyer ⁸⁵ with her buyer	that before a 20 th septemvir ⁸⁶
	⁸⁵ ctrl + ← ctrl + ⌘ ctrl + →	

Figure 3. Short transcription of repair behavior Frederik (+D).

As mentioned above, we analyzed the repairs on two different levels: the level of the error and the level of the error correction. These levels differ only in 2.30 % of the repairs of Frederik. Steven on the other hand more frequently tried to repair errors on a different level than the actual problem occurred. In 11.48% of the repairs the levels differ. To illustrate this strategy we have transcribed a short fragment in which Steven solves an error that occurred at word level (Figure 4). However, instead of only deleting the word 'a' he prefers to dictate the whole text segment again.

0.02.47	You state that my client has to get these before the end of the month	before the end of the month
	You state that my client has to get these a before the end of the month ¹⁰	before the end of the month
	•	<undo ¹⁰ >

Figure 4. Short transcription of repair behavior Steven (-D).

The distribution of technical problems and revisions over the two halves are almost the same for both writers. In the first half of the writing process repairs consist of 80% technical problems, and in the second half this amount drops to 70%. Similar to the reviewing behavior in writing with a word processor both writers revise twice as much in the second half of the writing process as in the first half.

4.3 Effect of repairs on modus monitoring

Speech recognition enables writers to produce and to revise their text with speech input. Consequently, writers that switch between the sub processes of formulating and revising can either choose to stick to the same writing mode or to switch between writing modes. In our previous analyses we noticed quite a lot of mode switches. From a cognitive perspective, and based on the observation that in the adaptation phase writers often experienced problems in navigating their text with speech, we hypothesized that these mode switches might frequently co-occur with the writer's decision to execute a revision or other repair.

In contrast to what we expected, our analyses of the case study analysis show that both Frederik and Steven do not switch writing modes in about 75 to 80% of the repairs. However, their overall writing profile is quite different. If we analyze the most preferred writing mode before a repair, we see a contrastive picture. Frederik uses keyboard & mouse before a repair in 73.4% of the cases as opposed to Steven who uses speech in 75% of the instances. In other words, Frederik holds on to using keyboard & mouse, while Steven prefers to continue to write with speech. The occurrence of errors in the text produced so far hardly seems to influence this behavior.

If we refine the analysis and take a closer look at the first and second half of their writing processes, we see an even more different pattern for Frederik and Steven (Table 3). In general Frederik has to deal with more repairs in the second half than in the first half, but he also switches more from speech to keyboard or mouse in the first half. Steven on the other hand, shows a more constant pattern.

If we add up to this the distribution of the writing mode and the mode switches within both halves, we see quite an inconstant pattern for Frederik. In the first half of the writing process, the writing modes speech versus keyboard & mouse are used almost equally before a repair, but in the second half Frederik prefers to write and make repairs with keyboard & mouse. That is the reason why he almost does not have to switch in the second half; his main writing mode is already keyboard & mouse. However, in the first half he switches about 35% from speech to another writing mode.

Table 3. The percentage of writing modes and switches per half of the writing sessions

	Frederik (+D)		Steven (-D)	
	1 st half % (n = 101)	2 nd half % (n = 204)	1 st half % (n = 151)	2 nd half % (n = 119)
Writing mode used before repair				
Speech	49.5	15.2	84.8	63.0
Keyboard & mouse	50.5	84.8	15.2	37.0
Type of writing mode switch				
No switch	60.4	81.4	78.1	82.4
From speech to keyboard or mouse	34.7	12.3	14.6	16.8
Other	4.9	6.4	7.3	0.8

Steven's mode behavior is quite constant in the first and second half of his writing process. Although the percentage of speech before a repair drops from 85% to 63%, Steven prefers speech in the first and in the second half of the writing process. He leaves the speech writing mode to make a repair only in 15% of the total amount of switches in the writing process. In other words, he prefers to produce and repair his text with speech recognition.

4.4 Immediate or delayed repair of errors

When writing with speech recognition, writers have two options to repair an error. Their first option is to repair it immediately, and they can try to write a 'first time final draft'. The second option writers have, is to dictate a first draft without paying much attention to technical problems or revisions, delaying the repair of errors to the final writing phase. Frederik's and Stevens writing processes can be characterized by a combination of both options (see Table 4).

Table 4. Moment and direction of repair

Moment of repair	Frederik (+D)	Steven (-D)
	%	%
Immediate	43.0	83.3
Not immediate	57.0	16.7
Direction of repair		
Backwards	52.5	86.3
Forwards	47.5	13.7

If we take a look at the moment and direction of the repairs (immediate vs. not immediate and backward vs. forward, cf. supra) we notice that about 83% of Steven's repairs are solved immediately, and that Frederik solves only 43% of his repairs immediately. Furthermore, we see that about 86% of Steven's repairs are backward movements in the text as opposed to 53% for Frederik. In other words, Frederik basically opts to repair and revise his text after a paragraph has been completed, or even waits until the first draft of the text is completed. This organization of the writing

process corresponds to research findings that report a more linear development of the writing process of dictators in contrast to a more recursive development of the non-dictators' writing processes.

Table 5. Percentage of technical problems and revision per draft (1st and 2nd)

Technical problem	Frederik (+D)	Steven (-D)
	%	%
1 st draft	65.6	89.2
2 nd draft	34.4	10.8
Revisions		
1 st draft	35.7	80.3
2 nd draft	64.3	19.7

In general Frederik solves 43% of his repairs for the second draft, whereas Steven solves his problems in 87% of the time in the first draft. Both participants deal differently with technical problems and revisions. Table 5 shows that Frederik solves most of his technical problems in the first draft (65.6%), but postpones revisions to the second draft (64.3%). Steven on the other hand, prefers both to solve technical problems and to revise in the first draft.

A further analysis of the repair behavior shows that Frederik only solves 35% of the misrecognitions immediately. This is in contrast with Steven, who solves 79% of the misrecognitions immediately at the point of utterance. The percentage of command misrecognitions solved immediately is even higher for Steven: 98%. He does not seem to tolerate this kind of large errors. Frederik solves 57% of the command misrecognitions straight away, and seems not to notice the others in the text produced so far at this stage in the writing process. In Table 6 we refine the type of errors into the categories: size, technicality and location of error.

Table 6. Percentage of errors immediately repaired

Size of error	Frederik (+D)	Steven (-D)
	%	%
Small error (≤ 2 characters)	47.72	67.24
Large error (> 2 characters)	47.03	95.51
Technicality of error		
Typical SR error	40.48	89.63
Keyboard & mouse error	96.15	60.00
Location of error in sentence		
End	64.10	94.15
Beginning and middle	17.78	39.43

Both participants differ in the amount and type of errors they solve immediately. However, both writers do solve nonexistent words as a result from typing errors immediately. They both do not tolerate this kind of larger error in their text. Frederik

almost solves all the keyboard & mouse errors in his text, but is more tolerant to the other categories. Steven is not that tolerant, and prefers to immediately solve almost all larger errors, typical speech recognition errors and errors that are located at the point of utterance.

5 Conclusions and discussion

In this chapter we have described the repair strategies of two writers with different dictating experience. The study confirms the potential hybrid character of speech recognition as a writing mode, since it combines characteristics of both classical dictating and keyboard based word processing. One of the main differences between classical dictating and dictating with speech recognition is that the writer gets feedback on the screen almost immediately. The case analysis presented here also lends support to the idea that the interaction with 'the text produced so far' is an important aspect of the task environment in the organization of the writing process (cf. also Hayes' model, 1996). It certainly influences the writing process, but also the extent to which writers use this feedback while dictating differs.

Both participants explicitly use the speech mode as an input device, but also seem to hold to their previous writing profile. Frederik maintained the writing habits he developed by using traditional dictating devices, while Steven holds to his word processing writing style and relies more heavily on the visual feedback of his dictation that appears as typed text on the screen. In other words, the speech recognition mode itself does not seem to trigger writers to adapt a specific writing style, as opposed to what happens when writers have to adapt to write their texts in either the dictating or the word processor mode (Schilperoord, 1996; Van Waes & Schellens, 2003).

One of the main differences in the writing process of both writers seems to be related to the mode monitoring in combination with the switching behavior. Although the amount of switches is comparable for both writers, the strategies to control the writing mode seem to be different. On the one hand, we can conclude that there is no direct effect of repairs on the monitoring of writing modes, because repairs do not directly lead to mode switches, namely Frederik and Steven do not switch writing modes in about 75-80% of the repairs. On the other hand, we can conclude that both participants differ in the way they repair their texts. In sum, Steven mainly repairs errors immediately (and backwards) as opposed to Frederik who often postpones the repair of errors, resulting in more grouped (and mostly forwards) repairs. They have developed other strategies. Frederik regularly *ignores* errors that appear in the text produced so far and *postpones* repairing errors to a later stage. Steven sometimes *anticipates* on errors and switches writing modes to avoid errors. For example, instead of trying to dictate personal names they switch - before they have to dictate them - to keyboard & mouse and type the name with keyboard & mouse.

So, both participants differ in the way they prefer to repair errors. They also differ in the amount and type of errors they solve immediately. Frederik prefers to solve almost all the keyboard & mouse errors in his text, possibly because he does not

have to switch writing modes to solve these repairs. He seems more tolerant to the speech recognition errors and often postpones the correction of those errors. Overall, Steven is not that tolerant and prefers to immediately solve almost all larger errors, typical speech recognition errors and errors that are located at the point of utterance. However, both writers do solve almost all nonexistent words immediately. They seem not to tolerate these kinds of errors in their text.

Is it strange that a writer who prefers a first time final draft, like Steven, does leave some errors in the text unnoticed? Are those errors not solved intentionally or are they overlooked? A possible explanation could be that smaller errors and errors in the beginning of the sentence are easier to miss. Earlier research has already shown that rereading of the text produced so far with the intention to further generate and formulate text, is characterized by a high degree of conceptuality. Writers in those instances do not really evaluate the correctness of their text, but only observe the 'gestalt' of what has been written as a trigger to further text production. So, on the one hand the interaction with the text on the screen can lead to a highly recursive writing process in which every error is repaired almost immediately, but on the other hand it can also lead to a less recursive writing process in which repairs are made at the end of a paragraph or a text and left unnoticed at the point of utterance.

In an experimental follow-up study we would like to further explore the repair behavior of the participants and more specifically the interaction with a (imperfect) 'text produced so far' (see also chapter 5 & 6). We would like to identify different characteristics of the (imperfect) 'text produced so far', and select material from our corpus as input data. In a controlled experiment we would like to present this material to participants in a writing task, in order to be able to better describe the interaction with the 'text produced so far', and evaluate the cognitive load caused by imperfect representations.

Notes

This study is part of an NOI research project on the influence of speech recognition on the writing process (Research grant of the University of Antwerp 2000-2002 & 2002-2004).

Acknowledgements

We would like to thank all the people that participated in this study, and especially Frederik and Steven, for their willingness to learn to work with speech recognition and let us observe them while doing so. We are also grateful to Lernout & Hauspie (now Nuance) who gave us a free license to work with VoiceXpress Legal™ during the course of this study.

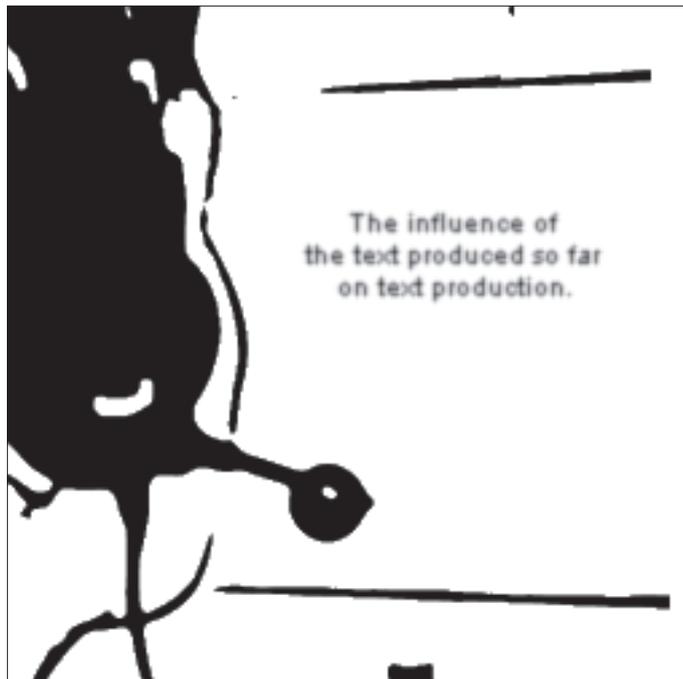
Appendix

Categorization model for the repair analysis

Repairs		
type of repair	1. technical problems	2. revisions
writing mode before repair	1. keyboard 2. mouse	3. speech
writing mode during repair	1. keyboard 2. mouse 3. speech 4. keyboard & mouse	5. combination speech & keyboard or mouse 6. training module
absolute time	absolute time of start correction in writing process	
halves	position of the correction in writing process by dividing session into two equal halves	
level of correction	1. character 2. word 3. sentence segment	4. sentence 5. paragraph 6. punctuation
remoteness of correction	1. within sentence 2. within paragraph	3. outside paragraph
writing phase	1. first draft	2. second draft
direction of correction	1. backward (before point of inscription) 2. forward (after point of inscription)	
Technical problems		
type of technical problem (taxonomy partly based on Karat et. al, 1999)	1. single misrecognition 2. multiple misrecognition 3. command misrecognition (symbol/punctuation)	4. command misrecognition (function keys) 5. dictation as command 6. no recognition 7. other
intention & outcome	1. word 2. symbol 3. function key	4. navigation 5. no input/output
cause of technical problem	1. user 2. speech recognizer	3. environment 4. no explanation
number of attempts	absolute number of attempts to solve problem	
success rate	1. successful	2. not successful
location of the error in the sentence (mathematical)	1. beginning 2. middle	3. end
occurrence of error	1. in speech recognition 2. in keyboard & mouse mode	3. in both writing modes
existence of output error	1. occurrence in SR	2. occurrence in SR K&M
error size	1. small (≤ 2 characters different)	2. large (> 2 characters different)
Revisions		
type of revision	1. addition 2. deletion 3. substitution	4. reordering 5. permutation
goal of revision	1. form 2. meaning	3. typing errors

Section II

Error correction strategies in isolated contexts



5

The effect of errors in the text produced so far

Strategy decisions based on error span, input mode, and lexicality

Abstract: Error analysis involves detecting, diagnosing and correcting discrepancies between the text produced so far (TPSF) and the writers mental representation of what the text should be. While many factors determine the choice of strategy, cognitive effort is a major contributor to this choice. This research shows how cognitive effort during error analysis affects strategy choice and success as measured by a series of online text production measures. Text production is shown to be influenced most by error span, i.e. whether or not the error spans more or less than two characters. Next, it is influenced by input mode, that is whether or not the error has been generated by speech recognition or keyboard, and finally by lexicality, i.e. whether or not the error comprises an existing word. Correction of larger error spans are corrected more successful smaller errors. Writers impose a wise speed accuracy tradeoff during large error spans since correction is better, but preparation times and production times take longer, and interference reaction times are slower. During large error spans, there is a tendency to opt for error correction first, especially when errors occurred in the condition in which the TPSF is not preceded by speech. In general the addition of speech frees the cognitive demands of writing: shorter preparation and reaction times. Writers also opt more often to continue text production when the TPSF is presented auditory first.

Keywords: Cognitive effort, dictation, dual task technique, error analysis, speech recognition, text produced so far (TPSF), text production, technology of writing, working memory.

This chapter is based on Leijten, M., Ransdell, S., & Van Waes, L. (submitted). Isolating the effects of writing mode from error span, input mode, and lexicality when correcting text production errors.

1 Introduction

To write well is to write with an eye for change. Modern writing technology allows the writer to produce and change the text easily via keyboard based word processing. As a consequence, a high degree of non-linearity is characteristic for these writing processes. In the late 90's speech recognition emerged as a writing medium. This medium is a hybrid writing mode that combines characteristics of classical dictating and word processing. The main strength of speech recognition lies in the combination of high speed text composition and the appearance of the text produced so far (TPSF) on the screen. However, writing with speech recognition does not yet result in a 100% faultless text on screen. For instance, when a writer dictates 'various' it can be recognized as 'vary us'. This kind of (semantic) errors require extra monitoring and make it more difficult to benefit from the speed of composition (Honeycutt, 2003). Consequently, writers that use speech recognition for text production must revise intensively. Even more than other writers they need to 'write with an eye for change'.

Revision during writing involves error analysis, comprising error detection, diagnosing and correction. This process has received much attention in cognitive science (cf. Rabbitt, 1978; Rabbitt, Cummings, & Vyas, 1978; Sternberg, 1969) and, more recently in computer based writing research (Hacker, 1997; Hacker, Plumb, Butterfield, Quathamier et al., 1994; Larigauderie, Gaonac'h, & Lacroix, 1998; Piolat, Roussey, Olive, & Amada, 2004). Even more recently, Leijten and Van Waes (2005b, also chapter 2, 3 and 4) reported on various error correction strategies of professional writers that were novice speech recognition users. The speech recognition users seemed to switch frequently from detection to correction, rather than continuing to write, resulting in a quite non-linear writing process. However, this observation did not hold for all the writers. A case study showed that one writer preferred to correct errors in the TPSF *immediately* and that the other writer showed a preference to *delay* error correction, with the exception of typical keyboard errors (cf. chapter 4).

Writers always need to make strategy decisions to deal with the errors in the TPSF. In the speech recognition mode, however, another process emerges. The writer constantly has to answer the question 'Is the text produced so far correctly presented?' In other words, is the text that the writer has dictated to the speech recognizer correctly presented in the text that appears on the screen (see Figure 1)?

The text dictated via speech recognition can either be presented correctly or not. If it is correct, the writer can continue with text production or plan on revising the text at a later point. Or the text can be presented incorrectly because a technical misrecognition occurred. In that instance, the writer may or may not detect the error in the TPSF. If the writer does, he can choose again between text production and reviewing. If a technical problem is detected by the writer, he can then ignore the problem, solve it at a later stage, or correct the error immediately. In the latter case, he can either perform the technical revision immediately or he can train the speech recognizer for

future reference. The speech recognizer does not necessarily 'know' every word that writers use and therefore one can make the software smarter by adding words to the dictionary and by training the pronunciation of words. Ultimately, speech recognition can lead to a strategy of delaying (minimal) error correction.

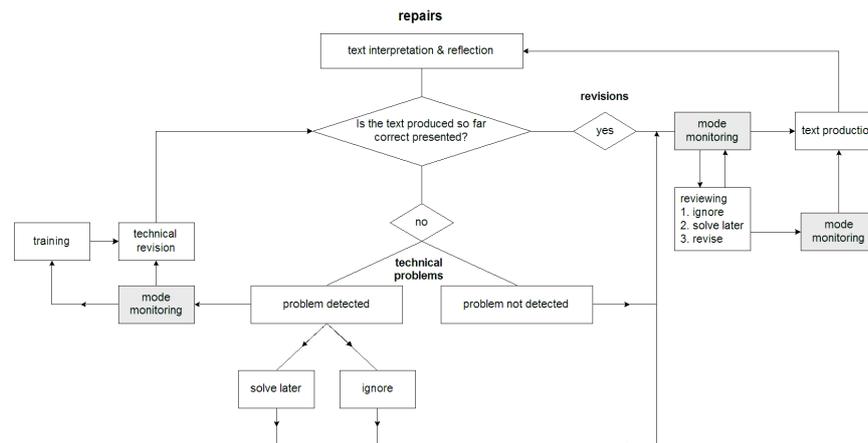


Figure 1. Speech recognition and the text produced so far.

A related quantitative study (Leijten & Van Waes, 2006b) showed that writers not only differed in the way they repair errors but also in the number and type of errors they solve immediately. Some participants solved almost all the keyboard & mouse errors in the text immediately – possibly because there was no need to switch writing modes to solve these repairs – while they were much more tolerant of speech recognition errors. Other writers, however, are less tolerant of this type of error and often immediately solve almost all larger errors, typical speech recognition errors, and errors that were located at the point of utterance. However, all writers did solve errors involving nonexistent words immediately. They seemed intolerant of these errors in their texts.

Is it strange that a writer, who prefers a first time final draft, at the same time delays to correct a few errors? Are those errors not solved on purpose or are they overlooked? A possible answer could be that smaller errors and errors in the beginning of the sentence are easier to miss. Earlier research has already shown that rereading of the TPSF with the intention to further generate and formulate text is characterized by a high degree of success (Blau, 1983). Writers in those circumstances do not really evaluate the correctness of their text, but only observe the 'Gestalt' of what has been written as a trigger to further text production. So, on the one hand the interaction with the text on the screen can lead to a highly recursive writing process in which every error is repaired almost immediately, but on the other hand it can also lead to a less recursive writing process in which errors are corrected at the end of a paragraph or a text and left unnoticed at the point of utterance.

The objective of the present study is to explain differences in revising behavior: why are certain errors immediately corrected, while others are delayed? We assume that working memory plays an important role in this decision process. Therefore, we describe the differences in cognitive load caused by various error types. In other words, some error types request too much attention of a writer, and need to be corrected immediately, before text production can be continued. Other errors can remain in the memory of the writer and do not need to be solved immediately. It is plausible that error correction strategies of writers are affected by working memory.

In the following sections we describe the requirements for successful error detection, the effect of the TPSF on the writing process and finally we elaborate on the influence of working memory in writing processes.

1.1 Successful error detection

Error detection is vastly improved if one can anticipate the requirements for fixing errors (Rabbitt, Cummings, & Vyas, 1978). During writing, errors can be any discrepancy between the TPSF and one's mental representation of how the text should be. Errors come in a wide variety of types and some are easier to process than others.

Larigauderie, Gaonac'h and Lacroix (1998) found that central executive processes in working memory (cf. section 4.3) are involved in detecting semantic and syntactical errors, but less so for typographical errors. Furthermore, they found that the disruption of the phonological loop mainly affected processing above the word level. They also found that greater processing spans, ranging from one word, several words within a clause, to words across clause boundaries respectively, required more memory resources than smaller spans. These two variables, error type and processing span were additive in their effects on successful error correction. In the writing task Larigauderie et al. (1998) used, a page long text was presented including errors of many types not isolated by an experimental design. In the present experimental study, we present error types that naturally occur in a typical writing task to determine strategy decisions writers make at the point of utterance when hearing and/or seeing text.

Hacker et al. (1994) found that writers first need to know how to correct a wide range of errors (meaning-based, grammar-based, or spelling-based errors) to detect them accurately. However, if an error is a simple typo, it is easier to detect than a meaning-based error because the latter requires text comprehension. Not only were spelling errors better detected, their detection also predicted correction. Not surprisingly, writing time, error type determination, along with the writer's linguistic knowledge, and knowledge of the text topic, facilitate error detection and correction.

At present, the jury is still out on how writing technologies such as speech recognition technology impact successful error detection. Detailed online records of writing by keyboard and speech recognition software will reveal the most common types of errors created by both, and the ways in which writers change their strategies to accommodate them.

1.2 The effect of the text produced so far on the writing process

The TPSF may play a different role in writing with speech recognition than it does computer-based word processing. Since speech recognition combines characteristics of both classical dictating and computer writing, it is useful to describe speech recognition as a hybrid writing mode (see Figure 2). That is why we position speech recognition between classical dictating and computer based word-processing. Similar to classical dictating speech recognition, texts are also audible via text-to-speech. Like in computer writing the TPSF in the speech recognition mode is visible on the screen (1). As in computer writing, the emerging text appears almost immediately on the screen, not letter by letter, but in text segments or phrases (2). After training, the text on screen is a more or less correct representation of what has been dictated (3).

We consider the presentation of the classical dictating device to be always correct because the audio on the tape is identical to the input; of course, the transition to paper is dependent on the typist¹. In computer writing, the on screen representation as classified as (semi) correct because of the typing errors that may occur. However, the misrecognitions of the speech recognizer - and consequently the incorrect representations on the screen - are of a different kind. There is a possible overt conflict between the TPSF and speech in speech recognition and a possible covert conflict in computer writing (4).

Classical dictating	Speech recognition	Computer writing
1. invisible text (audible after rewind) 2. no simultaneous feedback 3. correct presentation 4. no conflict in presentation	1. visible and audible text 2. (semi) simultaneous feedback 3. (semi) correct presentation 4. overt conflict between TPSF and speech	1. visible text on screen 2. simultaneous visual feedback 3. (semi) correct presentation 4. covert conflict between TPSF and text as intended

Figure 2. Hybrid characteristics of speech recognition.

These characteristics lead to differences in the writing process between classical dictating and computer writing. Previous studies show that classical dictating is characterized by a high degree of linearity in the text production (Schilperoord, 1996). Writers dictate sentences or phrases one after the other and only few revisions are made. The only revising usually taking place is a mental revision before the text is dictated to the recorder. The computer writing process is typically characterized by a high degree of non-linearity (Severinson Eklundh, 1994; Van Waes & Schellens, 2003). Most computer writers consider the paragraph, or even a sentence, as a unit that is planned, formulated, reviewed and revised in short recursive episodes (Van den Bergh & Rijlaarsdam, 1996). The constant feedback on the screen offers them the possibility to revise extensively, without losing the overview of the final text (Haas, 1989a, 1989b; Honeycutt, 2003).

¹ For a more detailed description of speech recognition and text-to-speech we would like to refer to Honeycutt (2003), MacArthur (2006) and Quinlan (2006).

So, in contrast to the traditional dictating mode, writers using speech technology receive immediate written feedback on the computer screen that may overtly conflict with the dictated TPSF. This previously mentioned technical characteristic creates the possibility to review the text in all stages of the writing process either by speech or by the complementary use of keyboard (without speech), inviting non-linearity (4).

1.3 Working memory and writing

Revising in general and error correction in particular can be seen as cognitive demanding activities. Almost every article on writing includes a paraphrase on 'writing is cognitively demanding'. These high demanding activities have been described in several models on working memory (Baddeley, 1986; Baddeley & Hitch, 1974; Ericsson & Kintsch, 1995; Kellogg, 1996; McCutchen, 1996; Shah & Miyake, 1996). Writers need to juggle the constraints of the several subprocesses. As Torrance and Galbraith (2006, p. 12) state:

Finally, we have suggested that although some aspects of the writing process can be strategically controlled, others, such as the need to suppress irrelevant information or the need to re-read to refresh transient memory, arise as a consequence of a cycle of processing as it occurs on-line. No matter how skilled we are at managing the writing process, there is an irreducible core of potential conflicts. Writing will always be a struggle to reconcile competing demands. Writers have – motivationally – to accept this if they are to get the task done. (p. 12)

Kellogg integrated the six basic writing subprocesses with the working memory model of Baddeley (Kellogg, 1996, 2004; Levy & Marek, 1999). This model on working memory was developed after a range of experiments on working memory processing. Via the dual task paradigm (cf. method section 2.4) Baddeley found that the working memory consist of several subsystems. This main finding led to the development of a tripartite model (Baddeley, 1986). In short, the model consists of a central executive and two slave subsystems, the visuo-spatial sketchpad and the phonological loop. The visuo-spatial sketchpad stores visual and spatial information (i.c. visual TPSF) and the phonological loop stores verbal and auditory information (i.c. auditory TPSF). The central executive manages both parts.

The writing subprocesses that are most relevant to this study are reading and editing. Kellogg states that reading is related to the central executive and the phonological loop. Editing is related to the central executive. Figure 3 shows all the relations. Of course, editing can also happen prior to text execution. In this experiment we focus only on editing of the TPSF (cf. difference between internal and external revisions, Lindgren & Sullivan, 2006). Kellogg states that editing signals errors in the output of planning, translating, programming, and executing. Then, feedback about the error to the appropriate process is needed. A recursive pattern of the above-mentioned planning to executing process is put into action. This recursive process can occur immediately after production of the error, but may also be delayed. The strategy adopted by the writer for allocating working memory to monitoring versus formulation and execution affects the decision process of correcting immediately or delaying error

correction (Kellogg, 2004).

Writing and its subprocesses place a high demand on the storage and processing capacities of the working memory. The logic of speech recognition is to reduce cognitive demands, especially during the production of text, while increasing auditory resources available to aid rehearsal in a phonological loop (Kellogg, 1996). Quinlan (2004; 2006) has shown that less fluent writers show significantly increased text length and decreased surface errors during narrative creation by voice (speech recognition) as opposed to traditional text production by hand. Less fluent writers benefit from the lower physical effort in writing with speech recognition. The automaticity of text production is particularly important for skilled writing since general capacity may then be allocated to other subprocesses such as planning and revising (Bourdin & Fayol, 1994). It is not clear, however, if the execution characteristic of the writing mode is the most important characteristic writers benefit from². For example, speech recognition generates only real words as errors because these items are part of the available lexicon while word processed errors can be typographical errors resulting in non-words.

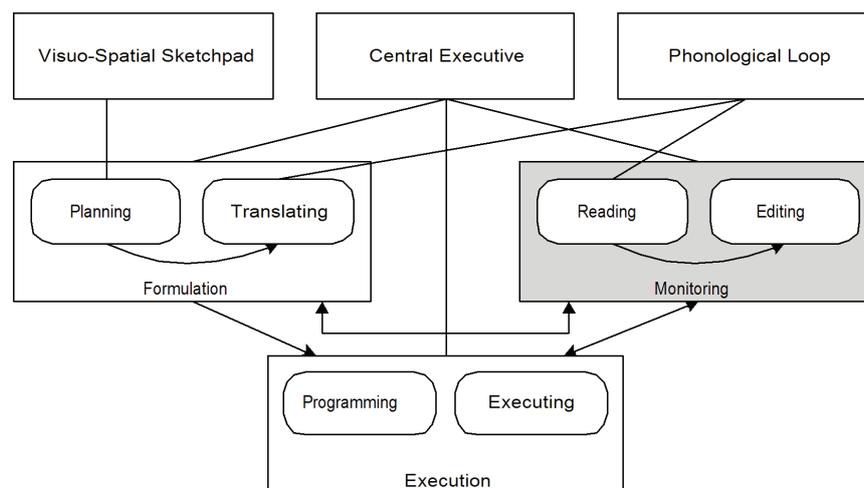


Figure 3. A representation of Kellogg's extension of Baddeley's model of working memory to writing. Adapted from Kellogg (1996).

In the vast amount of literature on working memory, several definitions have been mixed: working memory, cognitive load, cognitive resources, cognitive capacity and cognitive effort. As Figure 3 shows, the working memory exists of various components. Working memory refers to the ability of temporarily maintaining mental representations for the performance of a cognitive task. The cognitive load of a task refers to

² Most studies show that speech recognition could be less demanding to generate text (MacArthur, 2006; Quinlan, 2004; 2006) are done with special populations who already experience great demands from keyboard & mouse.

the load on working memory during problem solving activities (like writing). Piolat et al. (2004) present a clear distinction between the latter three definitions. Cognitive resources can be seen as the mental energy that is available at a certain moment during cognitive processes. The cognitive capacity is a more general measure that is quite constant per person. Each individual has a certain amount of resources that are available to him or her. This can be measured by various tests of working memory span (Daneman & Carpenter, 1980; Ransdell & Levy, 1999; Towse, Hitch, & Hutton, 1998). In this chapter we are mainly interested in the cognitive effort a writing task imposes on a writer. This effort corresponds with the amount of resources that are required by a writing task. In this study we will gain insight in the cognitive effort it takes for writers to solve various error types during text production, while the TPSF is offered only visually or also via the auditory channel.

Writing with speech may alter balancing working memory resources relative to writing without speech. Since increasing numbers of novice writers are learning to use speech recognition in an effort to improve their writing – or who, in case of repetitive strain injury are forced to do so – it is imperative to understand the changes that are being caused by this new mode of writing. Furthermore, despite the differences related to the writing mode, we hope that these findings will also contribute to a better understanding of the interaction with the TPSF during writing in general, that is, when writing without speech recognition.

1.4 Hypotheses

We assume that working memory makes a substantial contribution to the primary task of error detection and text completion and that it competes for resources with the secondary task of responding to an auditory probe. By comparing error correction strategies we can then determine the relative contributions of cognitive effort from several sources, error span, input mode, and lexicality. We assume that the cognitive effort will differ for error types. By comparing error types we can determine the effect of error types on the working memory.

1.4.1 Effect of error presentation mode (speech vs. non-speech) on the cognitive effort of error correction

The first hypothesis compares the effect of the two main experimental conditions: the TPSF was presented either visually or was read aloud also via voice. We expect that errors that occur after the initial clause has been shown in the non-speech condition require less cognitive effort than errors that occur after the context is offered in the speech condition. This assumption is based on findings in research on classical dictation (Gould, 1978; Gould & Alfaro, 1984; Schilperoord, 1996). Writers who dictate their texts to a machine need to make a mental representation of the text. We assume that writers who only have to compare the TPSF to the expected content are less distracted by the error than writers who have to compare the TPSF both to the

expected content and the speech prompt that contained an error-free ‘voice representation’.

Hypothesis 1: Errors that occur in the TPSF that were only showed visually cost less cognitive effort than errors that occur in the TPSF that also has been read aloud to the writers.

Our hypothesis is based on two lines of reasoning. As mentioned above, we expect that writers, who hear the dictated TPSF first, will be more focused on text production. In other words, offering the TPSF via speech might cause a focus on continuing text production. This positive characteristic of speech might have a counterpart. The focus on text production might cause an extra cognitive burden to switch to the revising subprocess and to evaluate the form and content of the TPSF. In other words, errors in the TPSF might distract writers because of the focus on text production. In theory writers do not need to read the TPSF, but the appearance of the TPFS on the screen – combined with positioning the cursor in the text – is such a strong trigger for text monitoring that we expect writers to ‘glance’ at the TPSF. We assume this action is rather superficial and mainly aimed at getting the ‘Gestalt’ of the text so as to continue production, rather than carefully rereading – and evaluating – the TPSF for further text production.

Secondly, we assume that in the auditory condition the writers may consider the reading of the TPSF as a kind of triple task: they get a written representation of the TPSF in the context screen, they hear the dictated form of the TPSF and then the (deficient) TPSF appears on screen. We expect that it is easier to detect and correct the error when writers only have to compare the visualization of the TPSF to the expected content and form. They will be less distracted by the error in the TPSF than writers who have to compare the TPSF on the basis of the (extra) auditory prompt that contains an errorless ‘spoken representation’ of the TPSF. Consequently, we expect it to be easier to detect an error in the TPSF in the condition without than with an auditory prompt.

1.4.2 Effect of error span on the cognitive effort of error correction

In the second hypothesis, we expect that large error spans (covering a character spread of more than two characters) cost more cognitive effort than small errors³. Large errors differ to a greater extent from the mental representation than smaller errors do. Leijten & Van Waes (2005) show that writers prefer to solve most large errors immediately after they appear/occur in the TPSF. We suggest two reasons. A first explanation can be that large errors are easy to detect and are therefore easy to solve immediately. Another explanation can be that large errors lead to a prominent deviation from the intended text that cannot be ignored. They impose a larger cognitive load on the writer and apparently create an urgent need to fine-tune the text again by correcting the error.

³ For instance, the difference between the correct spelling of the word ‘speech recognition’ and the incorrect spelling of ‘speech recognition’ (small error) on the one hand, and of ‘speech regoingition’ (large error) on the other hand (cf. section Materials).

Hypothesis 2: Large errors cost more cognitive effort than small errors.

Hypothesis 2 directly compares error spans. Error span refers to the number of characters separating components of an error. When the difference between the correct and the incorrect word is large (i.e. more than two characters), it may be easier to recognize the error, but at the same time, it may require more working memory resources due to the time delay required for maintaining the difference in representation. The time delay causes the need to re-read the sentence in order to engage in error correction or editing. This re-reading is known as highly complex and demanding on working memory (Just & Carpenter, 1992).

1.4.3 Effect of input mode on the cognitive effort of error correction

In the third hypothesis, we expect that small errors occurring in writing with keyboard and mouse take less cognitive effort to solve than small errors occurring in writing with speech recognition. Input mode refers to whether the error was naturally-occurring within text created on the computer by keyboard and mouse or speech recognition. Examples of typical speech recognition errors are possessive pronouns that become personal pronouns (mine vs. my) and double words if only one single word was intended (the the) and the insertion of full stops that unintentionally creates a new sentence (cf. section 2.3 Material).

Hypothesis 3: Errors that are originated in keyboard based writing cost less cognitive effort than errors that are originated in speech recognition.

On average, writers have much more practice with the types of errors made with keyboards compared to those with a novel system like speech recognition. The present study will directly compare the specific effect of each input mode among.

We expect the physical environment of writing with keyboard & mouse to be more closely connected to the writer than the speech recognition environment. In this period of time, this closeness can be the decisive factor in detecting errors in the TPSF⁴. Hypothesis 3 will compare the effect of small errors caused by keyboard or speech recognition entry.

1.4.4 Effect of lexicality on the cognitive effort of error correction

In the fourth hypothesis, we refine the third hypothesis. We expect that small errors that occur only in writing with keyboard & mouse take less cognitive effort than small errors that can occur in writing with speech recognition or keyboard & mouse, because the writers in the previous mentioned case study (Leijten & Van Waes, 2005b) showed a unanimous preference to solve nonexistent words immediately.

Hypothesis 4: Errors that are originated in keyboard based writing (non-existing words) cost less cognitive effort than errors that are originated in speech recognition or keyboard based writing (existing words).

⁴ Please note that the errors mentioned here only occur in speech recognition based writing processes, but that the participants themselves do not use speech recognition during this study. The errors are just originated in the speech recognition mode.

In the normal course of events, non-existent words only occur in writing with keyboard & mouse. The speech recognizer will generate, by definition, only existing words.

- Non-existing word: typing error in the word 'streert' instead of 'street'.
- Existing word: speech recognition error in the word 'eye' instead of 'I' (cf. section 2.3 Material).

Hypothesis 4 compares errors that involve lexicality (semantic level). Lexicality refers to whether the error is in a real word or a non-word. The former can be meaning-based or surface-based while the latter can only be surface-based and should therefore be easier to detect and correct. Real words should take greater resources to process than non-existing words, because the context involved in detecting the error necessarily exceeds the boundaries of the word itself. The findings in Hacker et al. (1994) and Larigauderie et al. (1998) suggest that spelling errors are easier to solve than semantic based errors, implying that non-existing word errors should be easier to solve than real word errors. In other words, we hypothesize that keyboard errors are easier to solve than errors that could occur as well in speech recognition as in keyboard based word processing.

In sum, the present study provides an analysis of the variations in cognitive effort related to error correction. The design includes the most frequently occurring error types found in a case study of professional writers (Leijten & Van Waes, 2003b, 2005b). The error types were presented to college students who were asked to detect errors and complete causal statements. An analysis of the task components of error correction will provide information about the mechanisms by which working memory resources constrain revision during writing.

2 Method

To answer the research questions above, we set up a controlled reading-writing experiment. So far, error correction strategies using speech recognition have been described in natural writing tasks. In this study we opt to isolate various error types that are most common during writing with speech recognition and with keyboard & mouse. Furthermore, the complexity of the writing task is controlled for. In this study we compare many writers' strategy decisions as a function of error type.

Participants were invited to participate in a one hour experiment during which they had to take two short initial tests and complete two sets of reading-writing tasks in two different modes, one purely visual task and the other a read aloud task before the visual representation of the TPSF appeared.

The task consisted of a set of sentences that were presented to the participants one by one to provide a new context. After every sentence the participants had to click the 'ok' button, indicating that they had finished reading the sentence.

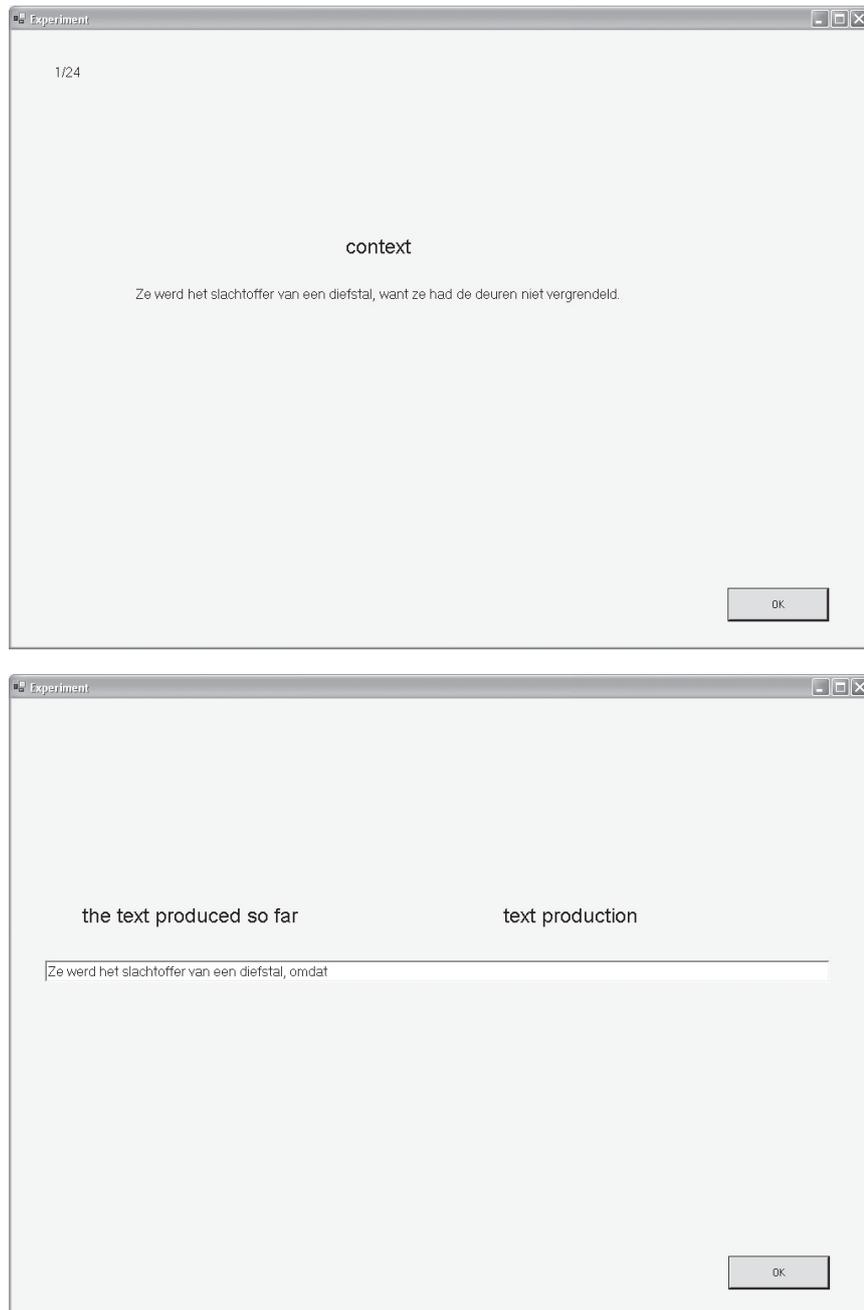


Figure 4. Example of (annotated) context screen and text produced so far.

A subclause of the previous sentence was then presented as TPSF in another subordinate causal structure, and the participants were prompted to complete the sentence. Figure 4 shows an example of the context sentence (top) and the TPSF positioned in a box to complete the sentence (bottom).

Four kinds of errors were implemented in the TPSF. In this section, we shortly describe the profile of the participants, the design of the experiment, the software used, the materials that were used in the reading-writing tests, and the procedure the participants followed.

2.1 Participants

Sixty students participated in this experiment⁵. The students all had Dutch as their first language and were between 18 and 22 years old. They were randomly assigned to the experimental conditions. The participants who volunteered to take part in the experiment received a free movie ticket.

2.2 Design

The experiment employed a 2 (experimental condition: mode of presentation speech versus non-speech) by 2 (two sets of sentences) within-subjects design (see Table 1). We constructed two sets of sentences in which an equal number of errors was distributed in a comparable way. Error type was equally distributed. The order in which these sets of sentences were presented to the participants was counterbalanced in the design of the experiment. The sequence of how the sentences were offered was also varied (only visual/non-speech or read-aloud and visual/speech).

Table 1. Experiment design

Number of group	Order of experimental condition (speech, non-speech and set of sentences)	
Group 1	non-speech set of sentences 1	speech set of sentences 2
Group 2	non-speech set of sentences 2	speech set of sentences 1
Group 3	speech set of sentences 2	non-speech set of sentences 1
Group 4	speech set of sentences 1	non-speech set of sentences 2

2.3 Materials

The main part of the experiment consisted of a reading-writing task. The participants had to read and complete 60 short sentences. They first read a short sentence which provided a context for the next sentence that had to be completed in the next step of the procedure.

All experimental sentences that contained deficiencies were marked by a causal coherence relationship. All the materials were presented in Dutch.

⁵ In total, 67 students participated in the experiment. On the basis of technical aspects (missing values due to a technical problem in the logging file), and an outlier analysis of both the quality of the corrections (primary variable) and the reaction time behavior (secondary variable), 7 students were excluded from the study.

Example:	
Context:	Because it has rained, the street is wet.
Correct TPSF:	The street is wet, because ...
Incorrect TPSF:	The streert is wet, because ...

In 24 out of the 60 sentences that were used in the experiment, we varied four types of errors to construct the deficient TPSF based on error span, input mode and error lexicality. The errors were taken from a larger corpus of data collected in a previous study on the influence of writing business texts with speech recognition (cf. section 1.4). The types of errors were replicated in the sentences built for the current experiment. The errors we selected were either caused by writing with speech recognition or by writing with keyboard and mouse.

An example of a typical error in the speech recognition mode⁶ caused by a mis-recognition of the dictated text is:

Example:	
(1) Spoken input	'I am writing a short text.'
(2) Incorrect output	' E ye am writing a short text.'

This kind of error will not occur in writing with keyboard & mouse. Other mistakes however could be classified as 'mode independent':

Example:	
(3) Spoken input	'The street is wet, because it has rained.'
(4) Incorrect output	'The street is wet, because it has drained .'

Because the 'd' and 'r' are adjacent keys on most keyboards, a writer could make this type of error easily. The typing error in this example leads to another existing word. Therefore, this error could also occur in the speech recognition mode. So although, the process that leads to the error may be different (ergonomic versus phonological), the written representation can be identical. On the other hand, some type of errors will not occur in speech recognition, and are exclusively found in texts produced with keyboard and mouse. These kinds of errors result in non-existing words.

Example:	
(5) 'The streert is wet, because it has rained.'	

Related to these characteristics of errors occurring in speech recognition and in writing with keyboard and mouse, we also decided to differentiate the size of difference (number of characters that are different between the intended word (clause) and the actual representation). Table 2 shows the classification of the errors taken into account the mode-specific characteristics of speech recognition and keyboard based word processing. An example of the four error types as they are included in the experiment can be found in Appendix 1.

⁶ Speech recognition always produces correct words. The words can be incorrect in a certain context, but the words themselves exist and they are always spelled correctly.

Table 2. Classification of errors

Category	Error span	Type of error	
		Input mode	Lexicality
SR Large	large: > 2 characters	only in speech recognition	existing words
SR Small	small: ≤ 2 characters	only in speech recognition	existing words
SR Keyboard Small	small: ≤ 2 characters	in speech recognition and keyboard & mouse	existing words
Keyboard Small	small: ≤ 2 characters	only in keyboard & mouse	non-existing words

The location of the errors in the sentences also varied. In each category, half of the errors were placed at the beginning and half of the errors at the end of the first text segment. One sentence was also constructed in each category in which there was an error at both locations. These sentences were randomly distributed in each test set. Finally, so as not to create a default attitude that was preconditioned to finding errors in the TPSF clauses, more than half of the sentences that were presented were correct and contained no mistakes. Some of these so-called filler sentences were constructed with a temporal relation instead of a causal structure. There was a concentration of correct sentences at the beginning of each set.

Four of the correct sentences were 'mirror' sentences of the incorrect sentences, allowing us to compare the reading-writing behavior in an even more controlled way on a small set of sentences. Table 3 gives an overview of the characteristics of each set of 30 sentences.

Table 3. Overview of the characteristics of the sentences used in each reading-writing task

Characteristics	Number of sentences
Test sentences (4 correct, 2 incorrect)	6
Incorrect sentences (4 type of errors * 3 places of occurrences)	12
Correct sentences	8
Correct sentences based on incorrect sentences (one sentence for every type of error)	4
Total	30

All sentences were more or less equal in length: the contexts contained 70 to 90 characters; subordinate clauses 31 to 43 characters; number of characters context set 1, $M = 78.42$ and set 2, $M = 79.22$.

Before the materials were used in the experiment, the manipulation of the sentences was pretested on validity among 32 students (17 male, 15 female). The students were asked to read 29 sentences consisting of a context and a short text segment that included information from the context (see example below).

Each sentence contained one or more errors. The errors were all marked. For each error the participants had to assess the seriousness of the error on a 7-point Likert scale. They were instructed to read the sentences as if they were part of a first draft of a group work. They had to indicate the seriousness of the error.

3. First part reading-writing test
4. Second part reading-writing test
5. Questionnaire

Before a new part started, the participants systematically received a written and an oral introduction to the new task (the instructions can be found in Appendix 2). A short trial task preceded every main task. To manage the experiment and the different flows, a special program was developed (Microsoft Visual Basic.Net). The program controlled the design, and stored the results of the tests, as well as several time stamps, the text produced in the writing task and the text location. We also integrated a Counting Span Test (CSTC) by Levy & Ransdell (2001)⁷ to assess the participants' cognitive capacity. To log the linear development of the writing process during the completion task, Inputlog (Van Waes & Leijten, 2006; also chapter 8) was used to capture the keyboard & mouse input and calculate the pausing time afterwards.

The first test was the *Counting Span Test*. This method assesses how well people can store and process information at the same time (working memory span). The Counting Span Test involves counting simple arrays of objects while simultaneously trying to remember an ever-increasing number of counts that are required as the memory sets become larger (Adams, Hitch, & Hutton, 1997; Levy & Ransdell, 2001; Ransdell & Hecht, 2003; Ransdell & Levy, 1999; Towse, Hitch & Hutton, 1998)⁸.

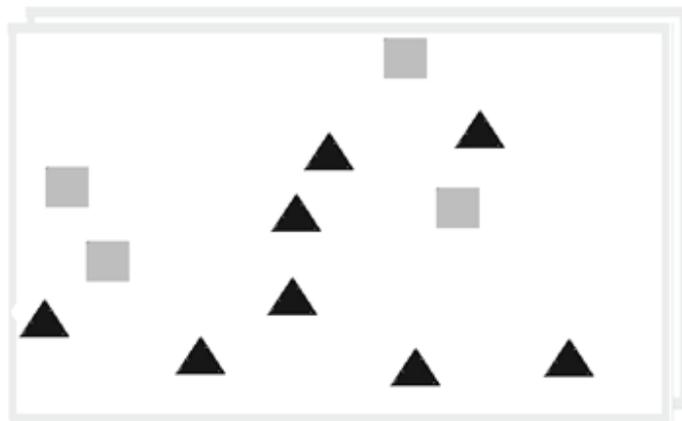


Figure 5. Example of card for Counting Span Test: squares to count (grey) and triangles to ignore (black).

⁷ This program can be obtained from www.psychologysoftware.com.

⁸ Towse et al. (1998) are cautious about this method, but Ransdell & Hecht (2003) showed cross-age effects. To increase difficulty we have chosen to perform the large card final method (the large card last condition required holding the count over a longer retention interval).

The participants were instructed to count all the blue squares on cards with blue squares and red triangles (see Figure 5). As soon as the participants had finished counting the blue squares, they had to type in the correct number. Then another card appeared and again the correct number of squares had to be typed in as fast as possible. After two, three or four cards, small blank textboxes were presented on the screen and the participants had to sequentially recall the number of squares on the previous cards.

The Counting Span Test started with two count cards as a test. After this trial session, the participants had to go through two main sessions. In both sessions the complexity of the task was gradually increased. The general measurement of working memory span was quite equally distributed over the participants. Almost all the participants reached the highest level of the Counting Span Test. As we expected the cognitive capacity of participants did not seem to differ very much. In Table 4 the general data of the Counting Span Test are shown.

Table 4. General data Counting Span Test

	<i>N</i>	Percentage of correct items		Mean time in seconds	
		%	<i>SD</i>	<i>M (s)</i>	<i>SD</i>
Level 2	52	95.51	19.83	4.32	1.33
Level 3	50	95.33	16.51	5.47	1.41
Level 4	50	91.84	22.08	7.31	2.06

The second test was aimed at measuring the mean baseline *interference reaction time* of the participants. As stated above, one of the most powerful ways of discovering working memory contributions to writing has been to employ dual-task techniques (Baddeley & Hitch, 1974; Kellogg, 1996, 2001; Levy & Ransdell, 2001; Olive, 2004; Ransdell, Levy, & Kellogg, 2002). Fisk, Derrick and Schneider (1986-87) review the criteria by which such techniques are most valid. First, the primary and secondary tasks must require a trade-off between common resources needed for processing and storing information. A vast literature on working span relies on a dual task in which words, numbers, sentences, or other stimuli are processed for meaning while other information must be stored for later processing (Chenoweth & Hayes, 2003; Daneman & Carpenter, 1980; Hoskyn & Swanson, 2003; Kellogg, 1999; Ransdell & Hecht, 2003; Ransdell & Levy, 1999). Second, the primary task performance must decrease as a result of the addition of the secondary task. In writing the degradation of writing performance is seen as evidence that the primary task writing suffers from the secondary task: cognitive resources overlap. An opposite effect can be interpreted in the same way. Longer reaction times to a secondary task can be interpreted as a high cognitive effort that needs to be invested in the writing task at the moment of the secondary task (Olive & Kellogg, 2002). Third, the resources allocated to each task must be stable. Implications for error analysis during writing mean that: the primary task, error detection and completion, must compete with the secondary task, in this

case, responding to a variable interval tone; the primary task must be overloaded by the secondary; and there must be practice to reach a consistent level of performance. When conducting a secondary task it is also necessary to measure the mean baseline reaction time as a reference measure. Thirty auditory probes were randomly distributed in an interval with a mean of 8 seconds and a range of 2 to 12 seconds. Participants were asked to press a button as rapidly as possible whenever they heard an auditory probe. After every probe the participants were asked to reposition their hands on the keyboard. The median baseline reaction time of each participant was calculated from the 25 last reaction times. The first five probes were treated as trial probes.

The third and the fourth tests were *reading-writing tests* with or without the addition of a spoken script prior to the visual presentation of the clause (TPSF). The participants were also informed that during the writing tests they would occasionally hear auditory probes (beep tones). They were asked to react as rapidly as possible to these probes by pressing the special button. During the reading-writing tests, the probes were distributed semi-randomly over the sentences that had to be completed. In the sentences with an error the probes were always presented in such a way that they occurred during the reading process; in the other sentences, especially in the test phase and in the non-causal sentences (temporal sentences), they were randomly distributed either in the reading or the writing phase. In some sentences the probe was not offered so as not to condition the participants.

The participants were informed that they should complete the sentences and that they should always try to write correct sentences. They were also told that they should focus both on accuracy and on speed. They had to finish the sentence as fast as possible and they had to – if necessary – correct the errors in the part of the sentence that was presented as a TPSF prompt. It was also explicitly mentioned that they should decide themselves if they preferred either to correct the sentence first, or to complete the sentence first and then correct the TPSF, if necessary. Next to this they were also instructed to respond as rapidly as possible to the auditory probes.

In the final *questionnaire* the participants were asked about their previous experience in computer-based word processing and speech recognition. Additionally they were asked about the task complexity and to indicate which sentences they did not fully complete and why.

2.5 Dependent variables

In order to analyze cognitive effort and error correction strategy, six dependent variables were derived from the logging data of the TPSF-program and Inputlog.

(a) Preparation time

The preparation time was defined as the time that passed between the moment the context screen was closed and the first mouse click to position the cursor in the TPSF screen, either to complete the sentence, or to correct an error in the TPSF. Since the

items in the speech condition appeared only after the spoken script was presented, the duration of this script was subtracted from the preparation time.

(b) Delayed error correction

For every sentence we logged whether the cursor was initially either positioned within the TPSF clause that was presented as a writing prompt, or after the clause. We used these data as an indication of the participants' preference to correct the error in the TPSF first or to complete the missing part of the sentence first (and delay the correction of the error, if at all it was noticed). Inputlog enabled us to code these data. The initial position of the cursor in the text completion part was programmed as a set x-value. This corresponds to a fixed x-value in the logging data, that is, 380: lower values refer to a cursor positioning in the TPSF and vice versa.

Example of logging output:
(1) Left Button: 00:00.15,109 (ClockTime) 274 (x-value) 345 (y-value) = in TPSF-segment.
(2) Left Button: 00:00.18,329 (ClockTime) 380 (x-value) 339 (y-value) = in text completion segment.

In our experiment the participants preferred to complete the sentence first in 89.2% ($SD = 15.6$) of the items. The individual preference ranged between 50% and 100%.

(c) Production time (writing)

The production time was defined as the period between the moment when the screen with the context sentence was closed and the moment when the screen with the TPSF to be completed was closed. In this period the TPSF was read, the sentence was completed, and possible errors in the TPSF were corrected. For the items in the speech condition, the duration of the spoken script was subtracted from the production time.

(d) Production quantity

In the analysis of the production quantity, the number of characters that were generated in completing the TPSF was calculated. The production quantity in the revision of the first clause of the sentence was not included in the analysis. We also calculated the difference between the correct completion of the sentence (theoretical number of characters to be produced based on the context) and the actual number of production quantity by the participant in the experiment. On average 30 characters had to be produced (about 5 to 6 words per clause). The number of characters to be produced in every item was kept almost constant for every item ($SD = 0.7$). Table 5 shows that the actual number of production quantity by the participants highly corresponded to the theoretical optimum: the average difference is only 1.3 characters.

The high equality between the theoretical optimum and the experimental realization showed that the writers succeeded in the text completion task: they completed the content of the text as requested (error correction is measured in accuracy, cf. below).

Table 5. Mean number of characters in completing the TPSF-clause

	Mean number of characters	
	<i>M</i>	<i>SD</i>
Theoretical optimum	30.7	0.7
Experimental realization	29.4	1.8
Difference	1.3	1.0

(e) Accuracy (quality of the correction of errors)

Accuracy here represents the percentage of sentences with a (manipulated) error that were rewritten correctly. Half of the sentences in the experiment contained either one or two errors which were always presented in the first clause of the sentence that had to be completed (TPSF). The participants corrected an average of 86.6 percent of the errors ($SD = 8.4$). In the analyses we only integrated the sentences that were corrected.

(f) Reaction time

The reaction time was defined as the time that passed between the moment when the auditory probe (beep) was given and the moment the button was pressed. The median reaction time of the baseline test for each participant (cf. initial reaction time test) was subtracted from the reaction time of each item to correct for individual differences. When the participant failed to press the button before completing the sentence, we coded the reaction time as missing; when the reaction was shorter than the participant's median of the baseline test, the reaction time was recoded to a zero value (representing a very short reaction time). Because we wanted to be sure that the reaction time in incorrect sentences was strictly related to the error and that the error was noticed as such, the reaction time for the incorrect sentences was coded as missing in those cases where the TPSF was not revised.

3 Analyses

To test the hypothesis we used the General Linear Model for the overall analyses, in which we compared the four error types and the experimental condition. To test the individual hypothesis per experimental condition (speech versus non-speech) we used the paired samples *t*-test⁹, because of the within-subjects design.

The analyses are based on the causal sentences that contained one error and that were corrected by the participants. We assume that errors that are not corrected are not noticed by the writers. The behavior of the participants is then the same as for the correct sentences. That is why we have only taken into account the corrected sentences.

⁹ The reaction times are per hypothesis based on the mean values. However, the overall measurement is performed with the median reaction times, because the largeness of the overall data we calculated the median here to reduce the impact of outliers.

4 Results

A set of six variables is used to test our hypotheses. Before we present the results of the statistical tests related to the hypotheses, we briefly report the general findings comparing the participants' interaction with the correct and the incorrect TPSF (overall and mirror analysis).

4.1 General findings

These results create a framework for the more specific error analyses presented in the results section. For the correct items, we excluded the sentences with a temporal relation as they were included in the experimental set as distracters. We only included the results for the correct sentences with a causal relation between the TPSF clause and the missing clause, as is the case in all the incorrect sentences. For the incorrect items, only those sentences that contained a single error in the TPSF were included.

Table 6 shows that when a correct TPSF clause is presented, the participants need less time before they decide to start completing the sentence ($t(59) = -5.25, p < .001$). This small difference is not confirmed in the mirror analysis in which we limited the analysis to the eight items that were both presented correctly and incorrectly (cf. 2.3 Materials).

Table 6. Mean values for the correct versus the incorrect sentences

	Correct		Incorrect		<i>p</i>
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	
Overall analysis					
Preparation time (s)	1.46	0.51	1.70	0.60	**1
Delayed error correction (%)	98.43	6.15	80.31	2.16	**1
Production time (s)	14.34	3.46	17.17	4.35	**1
Reaction time (ms)	250	125	272	142	*2

* $p < .05$; ** $p < .01$

¹ The significance is based on the overall data: GLM for the two conditions on the one hand and the four error types.

² The significance is based on the overall data: t-test for the two conditions (median).

The overall analysis of the delayed error correction at the start of the completion of the TPSF indicates that for about 20% of the incorrect items the participants preferred to correct the error prior to the completion of the sentence¹⁰.

The production time needed to complete (and correct) the TPSF is shorter for the correct items ($M = 14.34$ seconds, $SD = 3.46$) as opposed to the incorrect items ($M = 17.17$ seconds, $SD = 4.35$, $t(59) = -10.956, p < .001$). The analysis of the mirror sen-

¹⁰ We should take into account that the analysis of the initial cursor position is only an indication of the correction behavior. Of course, a participant can initially decide to complete (part of) the sentence, positions the cursor in that part, but then changes his mind and corrects the error first anyway. The deviation of the 100% score for the correct sentences in the general analysis is to be interpreted parallel to the noise factor in the incorrect items.

tences confirms this finding. The extra production time is partly used to correct the error, but in general also the time needed to read an incorrect TPSF takes significantly longer (cf. preparation time).

The overall analysis also reveals a significant difference in reaction time ($t(59) = -2.284, p < .05$). As expected, the cognitive effort is affected by the opposition correct versus incorrect TPSF.

4.2 The effect of mode of error presentation (speech vs. non-speech)

The first hypothesis about the experimental condition (speech versus non-speech) was not confirmed by the preparation times, the strategy inferred from delayed error correction, nor by the reaction time. Errors that occurred in the TPSF after the context had been prompted visually were solved differently than errors that occur after the context is offered both visually and auditory. Delayed error correction showed that the writers delayed error correction in the TPSF more often when the TPSF was presented also in speech. The results of the preparation time analysis (Table 7) build on this result.

Table 7. Mean values in both the speech and non-speech conditions

	Speech		Non-speech		<i>p</i>
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	
Preparation time (s)	1.10	0.47	2.38	1.22	**1
Delayed error correction (%)	93.81	17.63	66.07	35.42	**1
Production time (s)	17.86	4.72	18.89	5.38	⁻¹
Accuracy (%)	87.85	11.02	88.11	8.64	⁻¹
Reaction time (ms)	254	158	284	166	*2

* $p < .05$; ** $p < .01$

¹ The significance is based on the overall data: GLM for the two conditions on the one hand and the four error types.

² The significance is based on the overall data: T-test for the two conditions (median).

The speech condition of writing led to reliably faster preparation times and delayed error correction indicating that the speech condition leads to a more explicit preference to complete writing first and correct the error only after writing. The faster preparation time indicates that writers more superficially read the TPSF in the speech condition, which is in line with the shorter reaction time during reading in this condition. No other measures showed a significant difference between both conditions.

The preparation time for the items with an error in the TPSF in the condition without speech is longer than with speech (speech: $M = 1.10, SD = 0.47$ vs. non-speech: $M = 2.44, SD = 1.22, F(1, 59) = 79.576, p < .001$). The results for the analysis of the delayed error correction reinforce this finding and reveal another interaction with the TPSF: in the speech condition, the participants tended to complete the sentence first, while in the non-speech condition, they tended to correct the error first (speech: $M = 93.81, SD = 17.63$ vs. non-speech: $M = 66.07, SD = 35.24, F(1, 59) = 26.716, p < .001$). However, the results of the accuracy test show that the participants were not

more successful in the speech condition than in the non-speech condition (speech: $M = 87.85$, $SD = 11.02$ vs. non-speech: $M = 88.11$, $SD = 8.64$). Apparently, participants only preferred to correct the errors after having completed the sentence. On the basis of the data one cannot make a quality inference about the difference between a strategy of 'ignoring' or 'overlooking' the error in the first phase of completing the TPSF. There was a significant effect found in the cognitive effort: the reaction time in both modes was significantly different (speech: $M = 254$, $SD = 158$ vs. non-speech: $M = 284$, $SD = 166$); $t(59) = 2.47$, $p < .05$). In general, the speech condition freed resources to respond faster to the secondary task.

Apparently, writers are not aware of this positive effect of speech recognition. Half of the participants indicated in the questionnaire that the writing task without the addition of the auditory information was the easiest (46.55%), and the other half indicated that this addition made the task easier (53.45%). Only 8% of the participants had previous experience in speech recognition, only one of them had more than 20 hours of practice. They used it for word processing and for chatting.

4.3 The effect of error span on cognitive effort

Table 8 shows that the preparation time for the items with a large error in the TPSF is longer than that for the small errors (SR Large: $M = 2.07$, $SD = 0.47$ vs. SR Small: $M = 1.48$, $SD = 0.54$, $F(1, 59) = 36.39$, $p < .001$). The interaction between the type of error and the speech condition is also significant ($F(1, 59) = 13.37$, $p < .01$). Larger errors take more time to process, especially when the text is not preceded by speech. The main effect of delayed error correction was non-significant ($F(1, 59) = 2.57$, $p = .115$). However, the interaction effect was significant ($F(1, 59) = 5.53$, $p < .05$), indicating that writers in the non-speech condition preferred to correct the large error first and after that complete the sentence. The T-test confirms this finding ($t(59) = -2.38$, $p < .05$).

Table 8. Mean values in both the speech and non-speech condition for error span

	SR Large Error		SR Small Error		<i>p</i>
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	
Preparation time (s)	2.07	0.94	1.48	0.54	**
Speech	1.20	0.70	1.00	0.49	*
Non-speech	2.95	1.67	1.97	0.86	**
Delayed error correction (%)	78.75	25.14	82.92	24.13	-
Speech	93.33	21.52	92.50	24.05	-
Non-speech	64.17	41.26	73.33	37.36	*
Production time (s)	20.48	6.40	15.61	4.39	**
Speech	20.83	7.97	15.32	5.05	**
Non-speech	22.12	8.22	15.89	5.25	**
Accuracy (%)	92.50	16.14	84.17	19.51	*
Speech	92.50	22.22	85.00	26.52	-
Non-speech	92.50	20.22	83.34	23.77	*
Reaction time (ms)	372	255	303	161	*
Speech	332	344	270	172	-
Non-speech	423	309	345	233	**

* $p < .05$; ** $p < .01$

In both conditions, the writers needed more time to produce the text segment with the large error ($F(1, 59) = 115.95, p < .001$). The type of mode does not affect this finding. A closer look at the accuracy of solving the errors in the text correctly shows that large errors, those that take more time to process, are easier to correct. The accuracy for the large error is higher than for small errors ($t(59) = 2.38, p < .05$). Smaller errors are more easily overlooked, especially in the mode of writing without speech ($t(59) = 2.10, p < .05$). Whereas smaller errors are easier to overlook, larger errors distract more than smaller errors. The reaction time on the auditory probe is larger, and therefore takes more time, in sentences with a large error ($F(1, 59) = 4.35, p < .05$).

In our hypothesis we stated that it may be easier to recognize larger errors, but at the same time, it may require more working memory resources due to the time delay required for maintaining the difference in representation. To find evidence for this statement we analyzed the pausing behavior before the error correction. The pausing time before a revision in the TPSF is an important indicator of the cognitive effort it takes to correct the error. For the coding of this variable we used the linear text representation generated by Inputlog (also chapter 8). This XML file enabled us to identify each pause before the actual error correction. The pause threshold value for the analysis was 500 milliseconds. We coded initial error correction in TPSF (=immediate error correction) and error correction that followed text production (=delayed error correction). If writers start with error correction than the pausing time is equal to preparation time (positioning of cursor near error in TPSF). In cases where revision follows text production the pause is measured from the end of the text production until the movement of the cursor to the error in the TPSF. Figure 6 shows an example of both error correction types. The writers are both completing the next sentence (literal translation from Dutch - not phonological - for the error):

Example:

Context: Because the topic interests me, I will go to the conference next week.
(Dutch: Omdat het thema mij interesseert, ga ik volgende week naar die conferentie.)
TPSF incorrect: I am going to the **competence** next week, because
(Dutch: Ik ga volgende week naar die **concurrentie**, want...)

time	linear representation	explanation
14:20,00- 14:59,00	Left Button[947,710] Movement[340,342] {4000} Left Button[340,342] Movement[454,350] {672} BS 8 {688}fer{766}entie{547} Movement[414,352] {844}het · thema · interesseert · mij. Movement[904,697] {1141} Left Button[904,697]	click OK ([x,y] within zone 'ok'-button) positioning in TPSF-zone pause of 4 sec before revision (=immediate) deletion by backspace of 'concurrentie' typing of 'ferentie' mouse movement to production-zone finishing sentence click OK

time	linear representation	explanation
14:20,00- 14:59,00	Left Button[893,711] {3266} Movement[388,345] {547} Left Button[389,345] {687}het · thema · inte{672}resseer BS BS rt · mij. {1203} Movement[280,346] Left Button[295,345] Movement[308,350] fe Movement[912,702] {3328} Left Button[912,702]	click OK pause of something over 3 seconds positioning in production-zone sentence completion (correction typing error) pause of 1.2 sec before revision (=delayed) mouse movement to TPSF-zone selection of 'cur' replace by 'fe' (resulting in conferentie) click OK

Figure 6. Example of immediate versus delayed error correction.

Results in Table 9 show that writers paused significantly longer before a larger error than before a small error ($F(1,17) = .359, p < .01$). This result is the same for the speech and non-speech condition.

Table 9. Mean pause time for immediate and delayed correction for error span

	SR Large Error		SR Small Error		<i>p</i>
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	
Pause time (ms)	2.27	1.32	1.64	1.16	**
Immediate	4.72	2.34	2.93	0.88	**
Delayed	0.92	0.42	1.07	0.70	**

** $p < .01$

There is a strong interaction effect between the pausing behavior of error span and immediate and delayed error correction ($F(1,17) = .350, p < .01$). The pausing times before errors that are corrected immediately is probably distributed over several processes: reading the TPSF, detecting the error and perhaps building a mental representation of the text completion (not necessary in this order). This may account for the longer initial pause durations. When error correction is delayed, the focus is most likely on error detection, diagnosing and correction. The example in Figure 7 shows that the error detection has already happened during the initial preparation time, but that error correction is delayed. In those cases writers prefer to continue with text production first, probably to free their cognitive resources from maintaining the context information in their short term memory.

The example shows that the error detection probably took place during the initial pause of the preparation time. This explains why a pause of half a second is sufficient to position the cursor near the error in the TPSF. Other data show an even shorter pause before this positioning. The diagnosis of the error and the determination of the correction strategy (e.g. deleting the whole word versus opting for letter substitution) take more time and elapse in a more fragmented manner.

linear representation	explanation
Left Button[905,706] {2891} Movement[437,340]	click OK initial pause of almost 3 sec and positioning in productionzone
{563} het · t{750}hema · interesseert · mij. {531} Movement[286,348]	sentence completion (pause in 'thema') short pause before movement to TPSF
{2156} Left Button[293,348]	pause of 2 sec and positioning of cursor
{1141} Movement[303,346]	pause of 1 sec and small mouse movement
{703} BS BS BS fe{672} Movement[929,707]	short pause and change of 'concurrentie' in 'conferentie'
{1000} Left Button[928,706]	pause of 1 sec and click OK

Figure 7. Example of delayed error correction.

The pausing data might provide more information about the moment the participants detect and diagnose the error. The pausing data show various possible profiles. Short initial pauses lead to a fast decision process by the writers. Writers decide to correct the error immediately or continue text production based on the gist of the text. During longer initial pauses various strategies might be conducted. Writers could have long initial pauses that lead to immediate error correction. During the initial pause the writer focuses on sentence completion. He re-reads the TPSF in order to complete the text (grasping the gist of the text for further text production) and is confronted and consequently distracted by the error in the TPSF, which leads to immediate error correction. Conversely, the long initial pause could lead to delayed error correction. During the initial pause the writer again re-reads the TPSF in order to complete the text, and is confronted with the error in the TPSF. However, he decides to prioritize sentence completion and to delay error correction. The error detection is stored in short term memory to help locate the error after sentence completion.

The latter strategy can cause information loss. The example in Figure 8 provides an example of this situation. We are aware of the fact that we do not have direct evidence of the eye movements of the participants. This additional data could confirm our assumptions. The example is therefore more illustrative (cf. discussion). The example shows a rather long initial pause and a delay of error correction. The writer seems to hesitate on the word 'interest' and a typing error distracts the writer during text production, which again causes a longer (re-)orientation on the TPSF.

The examples in Figure 7 and Figure 8 show that writers have several options when allocating their cognitive resources in re-reading the TPSF and correcting the error. They also show that extra time delay does cause the need to re-read the sentence in order to engage in error correction. The data show that re-reading indeed is demanding on cognitive resources.

Section 4.4 and 4.5 provide more information on the error types, input mode and lexicality.

linear representation	explanation
Button[907,693] Movement[393,346] {3468} Left Button[393,345] {687} het · thema{796} · {907}inter{594}esser	click OK and direct movement to production zone initial pause of 3.5 sec and repositioning sentence completion (pause between and in words) short pause, double correction of typing error and completion sentence
{640} BS et BS rt · mij {3281} .{859} Movement[297,348] {2515}	pause before and after full stop and positioning in TPSF, pause of 2.5 sec
Left Button[296,348] Left Button[297,348] {1094} conferentie {1531}	selection of TPSF-error (full word by double click) pause of 1 sec and error correction pause of 1.5 sec
Movement[916,702] Left Button [916,700]	click OK

Figure 8. Example of delayed error correction (with additional error correction).

4.4 The effect of input mode on cognitive effort

The results in Table 10 show that the participants were equally distracted by the typical small speech recognition errors (cf. (d)rained) and the keyboard errors. The error categories do not lead to a difference in preparation time ($F(1, 59) = 2.854, p = .096$). However, the results show that the preparation behavior, especially when confronted with small speech recognition errors, is very diverse among the participants (Keyboard Small: $M = 1.20, SD = 1.21$). The production time and the reaction time are comparable for both error types.

Table 10. Mean values for small errors in both the speech and non-speech condition for the relation between the input mode and the error

	SR Small Error		Keyboard Small Error		<i>p</i>
	M	SD	M	SD	
Preparation time (s)	1.48	0.45	1.62	0.76	-
Speech	0.99	0.49	1.20	1.21	-
Non-speech	1.97	0.86	2.03	0.90	-
Delayed error correction (%)	82.92	24.13	77.08	24.91	**
Speech	92.50	24.05	92.50	24.05	-
Non-speech	73.33	37.36	61.66	43.54	*
Production time (s)	15.61	4.39	15.70	4.46	-
Speech	15.32	5.05	15.33	4.58	-
Non-speech	15.89	5.25	16.06	5.79	-
Accuracy (%)	84.17	19.51	96.25	9.00	**
Speech	85.00	26.52	95.83	13.94	**
Non-speech	83.33	23.77	96.67	12.58	**
Reaction time (ms)	313	171	332	201	-
Speech	286	196	328	239	-
Non-speech	342	232	344	242	-

* $p < .05$; ** $p < .01$

Although the participants do not seem to be more distracted by one error category or the other, they do show a stronger tendency to immediately correct the typical

keyboard error before completing the TPSF. The typical speech recognition errors do not seem to initiate this kind of behavior ($F(1, 59) = 8.48, p < .01$). Especially if the typical keyboard error is not preceded by speech, the participants prefer to solve the error first (interaction effect: $F(1, 59) = 8.48, p < .05$).

Our expectation that small keyboard and mouse writing errors can be better solved than small speech recognition writing errors is confirmed by the accuracy of both error types. The typical keyboard errors seem to be easier to solve, because the accuracy of typical keyboard errors is higher than the accuracy of the speech recognition errors (SR Small: $M = 84.17, SD = 19.51$; Keyboard Small: $M = 96.25, SD = 9.0$ ($t(59) = -4.66, p < .001$). In both conditions, that is, speech and non-speech, typical keyboard errors have a higher accuracy (speech: $t(59) = -3.423, p < .001$, non-speech: $t(59) = -4.000, p < .001$).

4.5 The effect of lexicality on cognitive effort

One of the characteristics of using speech technology as a dictating device is that errors that are generated through misrecognition, always result in existing, correctly spelled words or word groups ('eye' instead of 'I'). This is in contrast with most errors that occur in texts produced by keyboard & mouse. A typing error is often caused by pressing the key of an adjacent letter on the keyboard or just missing a key. Generally these typing errors result in non-existing words ('streert' instead of 'street'). It was expected that small typing errors resulting in non-existent words (Keyboard Small) are easier to identify when reading the TPSF and can be solved more efficiently, and with less cognitive effort than small errors that result in existing words (SR | Keyboard Small).

Most of the results shown in Table 11 do not support this hypothesis. Neither the preparation time nor the production or reaction time seems to be influenced by lexicality. Based on these analyses, the interaction with the TPSF is not influenced by the fact that the error in the TPSF is an existing word or not. However, in the latter situation, the participants prefer to complete the sentence first before correcting the error in the TPSF, especially if the TPSF clause is not read aloud. This is similar to findings in our previous study with novice speech recognition users who also did not tolerate non-words and corrected them immediately. (SR | Keyboard Small: $M = 70.83\%, SD = 39.37$ vs. Keyboard Small: $M = 61.66\%, SD = 43.54, F(1, 59) = 5.506, p = .022$ (no interaction effect: $p > .05$). Of the sixty participants, 17 always correct the error in the TPSF first for the items of the non-lexical error category (Keyboard Small) whereas 12 are characterized by a mixed pattern, explaining the high standard deviation for this variable.

The only variable that supports the hypothesis about lexicality is the accuracy (Table 11). The analysis of the quality of the error correction yielded a main effect of error category (Existing Words SR | Keyboard Small: $M = 89.17\%, SD = 21.33$ vs. Non-existing Words Keyboard Small: $M = 96.25, SD = 9.00; t(59) = -3.752, p < .001$).

Table 11. Mean values in both the speech and non-speech condition for lexicality of the error

	SR Keyboard Small Error Existing Words		Keyboard Small Error Non-existing Words		<i>p</i>
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	
Preparation time (sec)	1.62	0.72	1.62	0.76	-
Speech	1.52	0.69	1.20	1.21	-
Non-speech	2.08	1.10	2.03	0.90	-
Delayed error correction (%)	82.50	22.69	77.08	24.91	*
Speech	94.17	20.77	92.50	24.05	-
Non-speech	70.83	39.37	61.66	43.54	*
Production time (sec)	15.88	3.68	15.70	4.46	-
Speech	15.52	4.18	15.33	4.58	-
Non-speech	16.24	5.07	16.06	5.79	-
Accuracy (%)	89.17	13.31	96.25	9.00	**
Speech	88.33	21.33	95.83	13.94	*
Non-speech	90.00	20.17	96.67	12.58	*
Reaction time (ms)	361	238	334	207	-
Speech	419	383	328	235	-
Non-speech	326	197	341	239	-

* $p < .05$; ** $p < .01$

Errors resulting in existing words seem harder to identify than errors that result in non-existing words, both in the speech and non-speech condition. This result might also partly explain the difference in correction behavior (cf. delayed error correction).

5 Conclusions and discussion

The experimental condition, in which the TPSF was either offered only on screen, or also read-aloud before the visual prompt, influences the writers' strategies during error analysis. In the speech condition writers delay error correction more often and start writing sooner than in the non-speech condition. When speech is present, writers can overtly compare the TPSF on screen with the speech. However, when speech is absent, only an internal, covert conflict is possible. The added speech sometimes confirms that the TPSF is the text intended and sometimes it does not. Without speech, this kind of explicit confirmation is not possible. The present results show that writers adjust to this uncertainty in the TPSF by correcting errors immediately and by needing more time to either continue text production or correct the error (hypothesis 1).

Error span has a quite consistent effect on strategy choice (hypothesis 2). The effect is especially powerful when interacting with the speech condition. A text that is not preceded by speech and that contains large errors leads to the highest accuracy of error correction. On the one hand large errors lead to longer preparation and production times, slower interference reaction times, indicating that they consume more working memory resources, and on the other hand they produce a higher rate of error analysis success than small errors. When confronted with a large error, writers seem to choose a wise speed accuracy tradeoff. And this successful strategy is associated with a pattern of correcting errors when they are detected at the point of utterance. One implication of this result in general is that the TPSF influences error analy-

sis by encouraging frequent repairs of errors as they occur, even when they are large or more spread out in the text. The pausing behavior of writers provides extra evidence for the complexity of large errors. The pausing times were longer before large errors, indicating that it took writers more effort to solve them. However, when the error correction of the large errors was delayed the writers could easily return to the error, probably because they were so obvious to find. The correction was more difficult in this situation, perhaps because retrieving the correct information about the context is a highly demanding activity.

One benefit of the present research is that mode of writing can be separated from errors that are caused by a particular mode of writing, namely input mode. That is, writing with speech recognition generates a type of error that is distinct from those found when writing with keyboard & mouse, but the error types can be experimentally separated from mode of writing in the laboratory. In this experiment we have operationalized this on two levels: the input mode and the lexicality of words. The results for both operationalizations are to a large extent in line with each other (hypothesis 3 and 4). Both errors inherent to writing with speech recognition and with keyboard and mouse are equally distracting. But when speech recognition errors are involved the strategy is more likely to be associated with lower rates of error correction success and delay of correction. Keyboard based errors are better solved, especially when the TPSF is also read aloud.

We have seen that writers engage in different strategies to juggle the constraints that writing imposes on them. One of the biggest advantages of speech recognition over keyboard based word processing seems to be the fluency that writers can produce text with (respectively about 140 wpm versus 40 wpm). The main advantage of speech recognition over classical dictating is stated to be the immediate visual feedback on the screen. The TPSF is seen as an important input for further text generation. Concurrently the TPSF shows a delicate balance between cognitive processes. The re-reading of the last formulated sentences might have a positive effect on reducing working memory demands (Torrance & Gabraith, 2006). Alamargot, Dansac, Ros & Chuy (2005) showed recently that low-span writers read-back more frequently than high-span writers. However, Olive & Piolat (2002) found that removing visual feedback does not influence the quality of the text or the fluency with which it was written. The writers that were prevented from re-reading the short argumentative texts they were writing responded even faster to a secondary task. This provides evidence the hypothesis that re-reading the TPSF is very demanding. In our experiment the writers were in fact forced to read the TPSF, because they were asked to detect possible errors. So, the TPSF in general and the deficient text in particular appears to be a key element in the distribution of the working memory to free resources to do 'a good job' in any writing. Kellogg states that editing (in this study; repair) is a component of the central executive and reading of the central executive and the phonological loop¹¹.

¹¹ This topic in the discussion is partly based on personal communications with R. Kellogg of the University of St. Louis (June 2005 and February 2007). We would like to thank him for his supportive and critical remarks.

This does not seem to be completely in line with the results of the current experiment since the visible TPSF on the screen seems to influence the strategies of writers. We would like to suggest integrating the visuo-spatial sketchpad in the description of the monitoring part of the writing model (cf. Figure 1). The visuo-spatial sketchpad is used to maintain and process visual or spatial information (Levy & Marek, 1999), for example visualizing ideas, organizational schemes and lay-out. Research showed that the visual working memory is also involved in composing concrete definitions (Sadowski, Kealy, Goetz, & Paivio, 1997). The spatial component helps in organizing ideas about the text hierarchically (Kellogg, 1988).

Since the relation between the monitoring process and the working memory is crucial in this study, we first relate our findings between monitoring and the phonological loop and consequently add the influence of the visuo-spatial sketchpad. As stated before, monitoring interacts with the central executive and the phonological loop. The results of our study confirm the role of the phonological loop. For instance, the preparation time – indicating the time the writer needed to read the context phrase that needed to be completed – is both influenced by the accuracy of the TPSF and by the speech recognition (e.g. the addition of speech shortens the preparation time to complete the sentence – preferred strategy). What is more, the editing strategy and, in most instances, also the reaction time seems to be influenced by these factors indicating a relation between the monitoring process and the phonological loop. However, the results also show a distinct relationship with the visuo-spatial sketchpad. One characteristic of speech recognition errors is that the different outputs are phonologically identical, but they might have a completely different visual representation. The phonological loop might be helpful in retrieving the correct word, but we see that large errors take more preparation time, also in the condition that is preceded with speech. According to us, the visuo-spatial sketchpad can be of importance in two aspects: the orthographic (re)presentation of the visual word stimulus (screen) and visual-tactile (keyboard). A large speech recognition error can be divergent from the intended output. Two Dutch examples are (in English respectively: allude (to) and undo):

correct words:	alludeerde	ongedaan maken
incorrect words:	allen tegen dertien	ogen dan maken

The mental picture of the correct word is rather deviant from the incorrect representation. In our opinion writers 'see' the incorrectness of the word, not only because they re-read the text, but also as an apparent visual misrepresentation. We assume this is a different kind of monitoring than purely reading for flaws in the text and generating new ideas. On the other hand, Kellogg states that 'revising'¹² is a renewed activation of planning, translating, programming and execution. We believe the same holds for editing. However, the visuo-spatial component is not described yet from this point of view in the stages of error detection, diagnosing and correcting.

¹² Kellogg (1988) defines revising as a between draft activity: referring to the (final) phase of text composition.

In this experiment the errors are already placed in the text, and the text appeared as if it were dictated. The visual-tactile line of approach is related to typing with keyboard (although one could compare it to a mispronunciation while dictating to a speech recognizer). According to the model a small typing error is handled solely by the central executive. Here too, the visual(-tactile) information may be significant. Blind typists and seeing typists might 'feel' the errors that they make. Typing and dictating requires a constant monitoring of the output on the screen. Again, writers verify the direct relation between the key pressed, the word dictated, with the output on the screen. This is not only demanding for the phonological loop, or the central executive, but also for the visuo-spatial sketchpad. The influence of the word-picture on error detection and the visual-tactile can be the object of further research.

6 Further research

In this study we introduced a new technique to experimentally isolate the impact of error type. The results show that our hypotheses are not always confirmed, and that the interaction of writers with the TPSF is a very complex process, especially when the intended context is not correctly represented. Of course, when interpreting these results we have to take into account certain limitations that are inherent to the experimental setting and the way in which we implemented the errors in our data set. In the last part of this article we would like to discuss some of these aspects and relate them to options for further research.

In testing the four hypotheses we consecutively compared the effect of different error types. However, the dataset of these separate error types is rather limited per category, mainly due to our choice to create a bias towards correct sentences and to explore the effect of different error types. Therefore, some of these results should be interpreted with caution and a replication with larger data sets per category might be a good way to validate the current results. At this stage, however, it was our major purpose to explore the effect of all the different error types. Previous research (Hacker et al., 1994; Larigauderie, Gaonac'h, & Lacroix, 1998; Leijten & Van Waes, 2005) did not provide enough information, nor on error types, nor on the effect of the writing mode. In a follow-up study we would like to use a more limited set of error types, allowing for a larger number of occurrences in the data set.

Since the data in this experiment are repeated measures – in line with the common practice in writing research – we chose to conduct GLM repeated measures¹³ (ANOVA). However, from a statistical point of view, two objections can be made to our approach (De Maeyer & Rymenans, 2004; Quené & Van den Bergh, 2004). The first objection is that we did not take into account the nested data: the observations within each participant are correlated because they are made within the same participant. In other words, each sentence is nested in a participant. Since

¹³ The same measurement is made several times on each subject or case. If between-subjects factors are specified, they divide the population into groups.

we aggregated the data to the participant level we did not take into account the hierarchical structure of the data (60 participants * 48 sentences should be 2880 observations: 1440 correct ($n = 24$), 1440 incorrect ($n = 24$)). In addition, we needed to code some data as missing, because a subject missed data on one observation. Since ANOVA does not allow missing data, more data needed to be discarded. Multilevel analysis has more statistical power. Therefore we will re-analyze the dataset conducting multilevel analyses.

Another aspect of this study is related to the experimental setting. In this study we simulated a writing context in which writers were confronted with the TPSF in different conditions. The writers needed to store (planning) information in their short term memory to correctly fulfill the writing (completion) task. Obviously, this task differs from a normal writing situation. For instance, the (re)-reading and writing process in the experiment is more explicitly focused on the sentence level and related to local planning behavior of writers; the context is not only created mentally, but presented visually as textual material. Since we isolated error types effects of error types were described per type, while in complex writing tasks writers need to deal with combinations of error types. However, in spite of these differences, we observed similar patterns in the experimental setting as compared to our previous study in which we observed writers in their normal working environment (Leijten & Van Waes, 2005b, also chapter 2, 3 and 4). For instance, in both studies writers use different writing strategies to correct errors in the TPSF. Nevertheless, more research is needed – both ethnographically and experimentally – to validate the variety of cognitive processes during the interaction with the TPSF.

A third aspect of the study that needs further thought refers to the interpretation of certain subprocesses the writers are involved in during the interaction with the TPSF. This study confirms our hypothesis that errors in the TPSF influence the writing (and repair) strategies of writers. The data show that writers sometimes opt to correct the error first and sometimes continue with text production first. This difference is partially explained by the kind of error the writers are confronted with, and partially by personal preferences. In other words, a certain error type in the TPSF is likely to be handled in a specific way, but allows for certain variety. Based on the current study, however, it is difficult to relate this variety to certain personal preferences of the writers. Moreover, in those instances where the error in the TPSF is not corrected at the point of utterance, the collected data do not allow us to conclude whether the writers detected the error or not in the first interaction with the TPSF. This study gives insight in the global correction process of different types of errors, but to describe the error detection in more detail, it is necessary to replicate this study and add a more granulated observation method of the (re)reading behavior, for example eye-tracking. This technology would enable us to record every movement (saccade) and fixation of the writers' eye during the interaction with the TPSF. Data resulting from this kind of observation could provide a more detailed basis for analyzing the subprocess of error detection. For instance, when we observe a fixation of the eye on the error during the first interaction with the visual presentation of the TPSF, we will be able to differenti-

ate more explicitly between 'postponing strategies' and those instances in which the error is simply overlooked (cf. Figure 1).

Another issue is closely related to this aspect of the observation methods involved in the study, and refers to the kinds of (re)reading involved in the interaction with the TPSF. During this interaction different types of reading might be involved. Rereading the TPSF is often aimed at building a context to produce new text. The focus of the reading process is on the pragmatic, semantic and/or syntactic aspects of the text to make the information in the short term memory more explicit. The interaction with the TPSF can either take place on a local level (word/phrase), or on a more global level (multiple sentences). However, during the interaction the focus of the reading might change. For instance, due to an error in the text the focus of the reading process might shift to a more evaluative type of reading paying more attention to the spelling or other correctness aspects (even at a local level). The writer's reading attention span may also be oriented towards a verification of the writer's intention. In the experimental setting we have tried to create a situation that allows for all types of reading described above. For instance, by presenting the participants more correct than incorrect phrases, we tried to create a default reading attitude which was not evaluative but one which was oriented towards the production of new text (i.c. completing the sentence with a causal pattern).

Nevertheless, in further research, we would like to be able to get more information about the different cognitive processes and orientation of the reading process. We hope that the registration of the eye-movements through eye-tracking can provide data that enable us to characterize the reading process on the local level in more detail. We think that a more refined analysis of the interaction with the TPSF is important for a better understanding of the dynamics underlying text production.

The final aspect refers to the correction profiles of the writers. As described in Figure 1, the participants in our case study (Leijten & Van Waes, 2005) differed in the way they preferred to repair errors during writing. This observation also holds for the participants in the experiment at hand. Some participants prefer to correct most errors before completing the text, while others do exactly the opposite, or change their strategy. The behavior of the last two groups in particular opens perspectives for further analysis. What is the rationale behind the strategy of delaying the correction of an error? Which cognitive aspects related to the working memory influence this behavior? Is the strategy dominated by an absolute preference, or is it related to the type of interaction with the TPSF and/or the type of error encountered during that interaction? A more refined analysis in which the data are studied from the perspective of the participants' profile and complemented with more detailed information of the cognitive (sub) process, might reveal complementary factors. This type of analysis could take into account different subprofiles of writing strategies that are used by writers who delay error repair.

Generally, five steps are characteristic for the interaction with the TPSF aimed at the formulation of new text: (a) reading of the TPSF – (b) detection of the error – (c) diagnosis – (d) editing – (e) completion of the text. Based on the walkthrough of these

steps we can distinguish two diverse profiles: 'handle' versus 'postpone'.

Writers who do not delay the error correction follow this process linearly (ab-c d e), we would like to call this the 'handle profile'; other writers prefer a non-linear process, that could be bundled in the terminology as a 'postpone profile'. On the basis of our observations three different subprofiles can be distinguished in this last postpone group.

1. Gestalt profile [a e (a) bcd]: When following the 'Gestalt profile' writers do not really 'read' the TPSF, they only 'perceive the Gestalt of the text'. They are mainly focused on grasping the main gist of the text in order to produce new text. Only after completing (part of) the text they detect and diagnose possible errors and decide to correct them.
2. Detection profile [ab e cd]: The 'Detection profile' is characterized by an initial reading process in which evaluation of the correctness of the TPSF takes place while reading. However, writers who follow this profile prefer not to diagnose and edit the error first. Probably because of constraints of the working memory (sometimes related to the complexity of the error), they decide to complete the text first and delay the diagnosis and correction.
3. Diagnosis profile [abc e d]: The 'Diagnosis profile' only differs from the 'Detection profile' because writers in this profile decide to delay the editing of the error after having diagnosed it.

Therefore, we would like to replicate this kind of study taking into account the handle and postpone profiles, if possible, in combination with a more refined analysis of the types of reading involved (cf. supra). We think that the variation of condition - with and without speech - might again add a more layered interpretation of the analyses. Further research should address this issue more explicitly.

Acknowledgements

We would like to thank Isabelle De Ridder for her help in co-designing and coordinating the experimental sessions, and collecting the main data. Special thanks go to Sarah Ransdell for our fruitful discussions. Bart Van de Velde did a great job in programming the experiment. We also would like to thank Michael Levy for adapting the Counting Span Test for Visual Basic.NET. Finally, we thank Tom Van Hout for proofreading this chapter.

The project was funded as a BOF/NOI (New Research Initiatives) project by the University of Antwerp (2002-2004). The experiment software and logging tool are available from the internet at the following URLs respectively: www.ua.ac.be/marielle.leijten and www.inputlog.net.

Appendix 1

Example of four kinds of errors

Large difference, occurring only in speech recognition writing mode	
context	Omdat niemand anders het durft, wil ik zeggen dat het vak interessant is. Because no one else dares, I would like to say that the course is interesting.
1st segment	Dat wil zeggen dat het vak eendjes Zand is, want That will say that the course duckling Sand is, because (<i>literal</i>) That means that the course is in the rest thing, because (<i>plausible</i>)
Small difference, occurring in speech recognition	
context	Omdat hij mijn resultaten wilde bekijken, haalde hij het verslag uit mijn kast. Because he wanted to take a look at my results, he took the report out of my closet.
1st segment	Hij haalde het verslag uit mij kast, want He took the report out of my closet, because
Small difference, occurring in speech recognition and keyboard & mouse writing mode (existing words)	
context	Omdat je in het buitenland zat, ben je niet uitgenodigd op dat feestje. Because you were abroad you were not invited to that party.
1st segment	Je vent niet uitgenodigd op dat feestje, want You bloke not invited to that party, because (<i>literal</i>) You art not invited to that party, because (<i>plausible</i>)
Small difference, only occurring in keyboard & mouse writing mode (non-existing words)	
context	Omdat Jan tegen een boom was gereden, had de fiets schade aan het voorwiel. Because John had crashed into a tree, the bike's front wheel was damaged.
1st segment	De fiets had schade aan het voorwiel*, because De bike was damaged at the front wheek, because
	* The keys 'l' and 'm' are adjacent keys on a keyboard with azerty settings

Appendix 2

Overview of instructions

General procedure

Course of the experiment

The session will last about one hour and consists of 5 parts.

1. Memory test
2. Reaction test
3. Writing test | part I
4. Writing test | part II
5. Questionnaire

You will receive a short instruction before each test. Then you get the opportunity to practice. There is a short break after each test.

Good luck.

Counting span test

Memory test

The memory test shows a card with blue squares and red triangles.

1. Count the blue squares.
2. Enter the number of blue squares as soon as possible.
Hereafter appears a next page with squares and triangles.
3. Repeat the procedure.
4. Remember the number of blue squares of the previous pages.
5. After 2, 3 or 4 times you enter the numbers again in the boxes that appear.

The edges of the cards show you how many numbers you need to remember. In the example below it is three.

The memory test consists of three sessions: one practice session and two basic sessions.

To start with the practice session click 'ok'.

Introduction Baseline Reaction Test

Reaction test

In this test we will measure your ability to react.

1. Wait until you hear a beep.
2. React as quickly as possible to the beep by pressing the button next to the computer.
3. Put your hands back to your keyboard afterwards.

Don't keep your hands on the button.

Procedure 1: non-speech condition

Writing test | part I

1. A sentence will appear on the screen.
Read this sentence carefully and press the 'ok' button when you are finished.
2. A new sentence appears on the screen.
3. Complete this sentence as quickly as possible with the information from the first sentence. Some sentences that you need to complete might contain an error. Correct the error as soon as possible. You are free to choose whether you prefer to complete the sentence first or whether you correct the error first.

Remark:

The sentence needs to be perfect before you continue with the next sentence.

Attention please!

Now and then you will hear a beep. Push the button next to the computer as soon as possible. The first sentences that will appear are test sentences. This way you can practice.

Procedure 2: speech condition

Writing test | part II

1. A sentence will appear on the screen.
Read this sentence carefully and press the 'ok' button when you are finished.
2. You will hear a part of the next sentence via the headset.
3. A new sentence appears on the screen.
Complete this sentence as quickly as possible with the information from the first sentence. Some sentences that you need to complete might contain an error.

Correct the error as soon as possible. You are free to choose whether you prefer to complete the sentence first or whether you correct the error first.

Remark:

The sentence needs to be perfect before you continue with the next sentence.

Attention please!

Now and then you will hear a beep. Push the button next to the computer as soon as possible. The first sentences that will appear are test sentences. This way you can practice.

6

Isolating the effects of writing mode from error span, input mode, and lexicality when correcting text production errors

A multilevel analysis

Abstract: Error analysis involves detecting and correcting discrepancies between the text produced so far (TPSF) and the writer's mental representation of what the text should be. In the study presented in the previous chapter, we hypothesized that error correction with speech – that is dictating with speech recognition software – differs from keyboard as a delivery system for two reasons. It produces auditory commands and, consequently, different error types.

In this chapter we reanalyze the data from the TPSF experiment (chapter 5) from a hierarchical perspective. The study reported on here measured the effects of (1) mode of presentation (auditory or visual-tactile), (2) error span, whether the error spans more or less than two characters, (3) mode of writing, whether text has been generated by speech recognition or keyboard, and (4) lexicality, whether the text error comprises an existing word. In reanalyzing the data we have opted for a multilevel approach in order to increase the statistical power in comparison with the unilevel analysis. A multilevel analysis provides a sound basis for verifying whether certain participants' characteristics might have disturbed our interpretation of the effect of condition and error type on the writer's interaction with the TPSF. A multilevel analysis was conducted to take into account the hierarchical nature of

This chapter is an article in preparation Leijten, M., De Maeyer, S., Ransdell, S. & Van Waes, L. (in preparation). Isolating the effects of writing mode from input type, error span, and lexicality when correcting text production errors: a multilevel analysis.

these data. For each variable (interference reaction time, preparation time, production time, immediacy of error correction, and accuracy of error correction), multilevel regression models will be presented. As such, we take into account possible disturbing person characteristics while testing the effect of the different conditions and error types at the sentence level.

When comparing the resulting errors as manipulated variables, we find that it is the auditory property of the speech, the input type itself that frees resources for the primary task of writing, both when completing correct representations of TPSF clauses and non-correct clauses. Large errors cost more cognitive effort, and they are solved with a higher accuracy than small errors. The latter also holds for small errors that result in non-existing words.

Keywords: Cognitive effort, dictation, dual task technique, error analysis, multilevel analysis, speech recognition, text produced so far (TPSF), text production, technology of writing, working memory.

1 Introduction

In the previous chapter we analyzed several characteristics of the way in which writers deal with problems in the 'text produced so far' (TPSF) when producing new text. We were mainly interested in the influence of different types of errors in the TPSF on the cognitive load of the writer. Also the effect of an auditory presentation of the TPSF prior to a visual presentation of the TPSF on screen was part of the study. The aim of this manipulation was to create an experimental situation in which the reading process that precedes the production of new text could be observed in a more controlled way. The experimental setting should give us better insight in the interaction of writers with the TPSF during the writing process, performed either with traditional word processing software or with speech recognition software.

In the previous chapter we described the data of the TPSF experiment using traditional unilevel statistical analyses and while aggregating the observations per writer and per error type (Leijten, Ransdell, & Van Waes, submitted). The choice for unilevel analysis was a rather pragmatic choice and was in line with the statistical tradition in most writing research. In this field multilevel analysis is not yet very common. The most frequent analyses that are performed in error detection research are ANOVA Repeated Measures (e.g. Hacker, Plumb, Butterfield, Quathamier et al., 1994; Larigauderie, Gaonac'h, & Lacroix, 1998). The application of multilevel analyses is mainly disseminated by Van den Bergh (Quené & Van den Bergh, 2004; Van den Bergh & Rijlaarsdam, 1996)¹. Furthermore, we opted to use unilevel as a first exploration of the dataset, and to describe the tendency of the data. However, we are aware that multilevel is a more powerful and precise research method.

¹ For a guide to multilevel analysis we would like to refer to a tutorial by Quené and Van den Bergh (2004).

The unilevel approach leads to a possible loss of statistical power due to data aggregation on the participant's level resulting in one mean score per condition and per error type. These aggregated data do not always adequately treat differences between writers and between sentences when analyzing their behavior during the interaction with different error types in the TPSF (presented in two conditions). By aggregating we created data on how our respondents preferred to react on a TPSF of a certain kind, but we leveled out the possible individual differences between the sentences. Not taking into account this nuance can lead to an aggregation bias in the interpretation of the analyses. To avoid this aggregation bias (Bernstein, 1990) and fully take into account the within-writer and the within-sentence variance, multilevel analyses can be performed. Van den Bergh & Rijlaarsdam (1996) introduced multilevel analysis in writing research. They argue that multilevel models are often more powerful and that each observation can be nested within individuals (within error types and conditions). The result of this method is that each observation can be treated equally taking into account the differences between writers and sentence characteristics (Van den Bergh & Rijlaarsdam, 1996, p. 220). Also the fact that the number of observations per person sometimes slightly differs², does not affect the power of the analyses. So, the main advantage of multilevel methods is that they account for the hierarchy within collected observations and the dependencies within a hierarchical structure (see also Goldstein, 1995).

In this chapter we will reanalyze the data from the TPSF experiment described in the previous chapter from a hierarchical perspective. This multilevel approach enables us to question the loss of statistical power of the unilevel analysis and provides a sound basis to verify whether certain participants' characteristics might have disturbed our interpretation of the effect of condition and error type on the writer's interaction with the TPSF.

For each variable (interference reaction time, preparation time, production time, immediacy of error correction, and accuracy of error correction), multilevel regression models are presented with TPSF sentences (level 1) nested within respondents (level 2). Each model consists of three parts: an estimated mean for that specific variable, a characterization of each writer (as a deviation of the mean), and a characterization of each TPSF sentence as it was presented to the writer (as a deviation from the mean of that writer). This approach enabled us to analyze the effects of different conditions and error types on each dependent variable at the sentence level, taking into account possible disturbing person characteristics.

Before we introduce the different models in more detail, we first present the hypotheses that will be the focus of the multilevel analysis and explain the method of this study. For a more elaborate overview of the background of the experiment and a

² Although the number and type of sentences is strictly controlled in this experiment – as opposed to more ecologically valid observations of writing processes – a few sentences could not be added to the data set for technical reasons (e.g. when the starting time preceded the beep for the second task the reaction time was not taken into account because it did not intervene with the reading of the TPSF). The results for the variables related to these sentences were coded as missing values, which resulted in a data set with slightly deviating total numbers of scores.

review of the related literature, we refer to the introductory sections of the previous chapter (see chapter 5).

2 Hypothesis

Based on the results of the previous analysis and taking into account the possibilities of the multilevel perspective, we reconsider the hypotheses put forward in the previous chapter. Of course, the main perspective of this study is to re-examine the hypotheses in the light of the extra power of the multilevel approach. Therefore, as in the previous study, we mainly focus on the effect of the experimental speech condition and the three error types: (1) error span: large versus small size errors, (2) input type: speech recognition based errors versus keyboard based errors, and (3) lexicality: existent versus non-existent words. However, we would also like to make one step backwards in the first hypothesis and consider whether there is any effect of the experimental speech condition on the (cognitive) interaction with TPSF's that are presented correctly, that is without any manipulated errors.

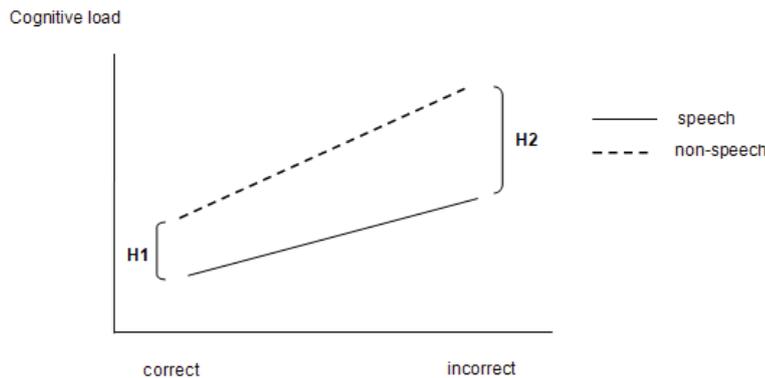


Figure 1. Hypothetical relation between auditory condition, correctness of the text.

Hypothesis 1: effect of mode of presentation (auditory versus visual-tactile) on the interaction with correct text produced so far (TPSF)

We hypothesized that the addition of speech in the auditory condition could have a positive effect on the memory load of writers during the initial interaction with the correct TPSF (see Figure 1).

This hypothesis is based on the presupposition that processing text via the auditory channel (speech recognition) requires fewer cognitive resources than via the visual-tactile channel (word processor), especially when no correction is involved. Therefore, in this hypothesis we expect, for instance, that Interference Reaction Time,

a measure of the time needed to turn attention to the secondary task of responding to an auditory probe while generating text, to be faster during speech presentation compared to the single visual-tactile presentation. Other measures of cognitive load (e.g. preparation time) should point in the same directions.

The main reason why so much effort – and money – is invested in building powerful speech recognition systems is that this technology enables physically disabled persons to write and edit text using their voice, but also that the actual production of text is much faster than with keyboard and mouse (Honeycutt, 2003). One could expect that this optimization of text production would lead to a decrease of cognitive effort related to this subprocess. Moreover, as Crossman (2004) argues in his futuristic essay, VIVO [Voice-In/Voice-Out]: The Coming Age of Talking Computers, on the rising oral culture, speech can be considered as the most natural mode of language production. He sees writing as a ‘transitory technology’ and presents cultural, philosophical and historical evidence for this point of view (see also introductory chapter).

In the best possible use of speech recognition software for writing, all sentences should be produced correctly. The present experimental design takes advantage of the possibility of first isolating correct from incorrect sentences and therefore maximizes the possible positive impact of auditory input on memory load. The isolation of the correct sentences and the assumed positive effect of the auditory condition on the cognitive effort in the production task is the basis for the first hypothesis.

Hypothesis 2: effect of mode of presentation (auditory versus visual-tactile) on the interaction with incorrect text produced so far (TPSF)

Unfortunately, the ideal world as described above does not yet exist. At this moment the state of the art of speech recognition is such that the errorless production of text is not possible (accuracy levels of expert users of speech recognition are up to 99% in an ideal situation, resulting in about one or two errors every five lines). Consequently, writers must develop compensation strategies for dealing with the errors in the TPSF (Leijten & Van Waes, 2005, also chapters 2, 3 and 4). Therefore, we formulate expectations for writing processes that are not errorless and in which the already produced text does contain deficiencies.

The second hypothesis compares the effect of the TPSF that was either presented auditory first (speech condition) or only visual-tactile (non-speech condition). Based on the results in chapter 5, we expect that errors that occur after the initial clause has been shown only visually, without speech, are more cognitive demanding than errors that occur after the context is offered both visually and auditory, with speech (see Figure 1).

Hypothesis 3: effect of error type on the interaction with the text produced so far (TPSF)

In the final hypothesis we formulate our expectations related to the different kind of errors that appear in the deficient TPSF clauses. We distinguish three error types based

on (a) error span (large vs. small errors), (b) input type (small speech recognition errors vs. small keyboard errors) and (c) effect of lexicality (existing vs. non-existing words). Our expectations are mainly based on the (tendencies of the) results presented in the previous chapter (see Table 1).

Table 1. Overview of hypothesis three in relation to cognitive load

Hypotheses			
H3a	SR Large errors	>	SR Small errors
H3b	SR Small Keyboard errors	>	Keyboard errors
H3c	Existing words	>	Non-existing words

Hypothesis 3a: effect of error span (small vs. large errors)

This hypothesis directly compares error spans. Error span refers to the number of characters separating components of an error. When the difference between the correct and the incorrect word is large (i.c. covering a character spread³ of more than two characters), it may be easier to recognize the error, but at the same time, it may require more memory resources due to the time delay required for maintaining the difference in representation.

Based on the results in the previous chapter, we expect that large error spans - more than two characters – lead to a higher cognitive load than small errors.

Hypothesis 3b: effect of input type (small speech recognition errors versus small keyboard errors)

In this hypothesis we compare the effect of the input type, i.c. small errors made by keyboard or speech recognition entry. Based on the tendencies of the results in the previous chapter, we expect that small errors that occur in writing with keyboard and mouse can be solved with less cognitive effort than small errors that can occur in writing with speech recognition.

The participants in our experiment have much more extensive practice with the types of errors made with keyboards compared to those with a novel system, speech recognition. Therefore, we expect the physical environment of writing with keyboard & mouse to be more closely connected to the writer than with speech recognition. In this period of time, this proximity can be a decisive factor in efficiently detecting and correcting errors in the TPSF.

Hypothesis 3c: effect of lexicality (existing words versus non-existing words)

This hypothesis compares errors that involve lexicality (semantic level). Lexicality refers to whether the error is within a real word or a non-word. Errors within a real word can be meaning-based or surface-based while errors within non-words can only be surface-based. Because speech recognizers use a lexicon, they will generate,

³ For instance, the difference between the correct spelling of the word ‘speech recognition’ and the incorrect spelling of ‘speech recognition’ (small error) on the one hand, and of ‘speech recoingition’ (large error) on the other hand (cf. materials section).

by definition, only existent, real words. In the normal course of events, non-existent words only occur in writing with keyboard & mouse and are caused by typing mistakes.

The finding in Hacker et al. (1994) and Larigauderie et al. (1998) that spelling errors are easier to detect than semantic errors, suggests that non-word errors should be easier to detect than real word errors. Therefore, and also on the basis of the tendencies of the results in the previous chapter, we predict that errors resulting in non-existent words - that occur only in writing with keyboard and mouse - can be solved more efficiently than errors resulting in other existent words - that can occur in writing with speech recognition or, by chance, in keyboard & mouse.

In sum, the present study provides an analysis of online text production that isolates the effects of writing mode from, accuracy, error span, input type, and lexicality. The design includes the most frequently occurring error types found in a case study of professional writers (Leijten & Van Waes, 2003b; 2005b, see also chapters 2 and 3). The error types were presented to college students who were asked to detect and correct errors in the TPSF and complete causal statements (no order was specified).

An analysis of a wide range of (online) measures as (a) interference reaction time, a measure of working memory resource allocation, (b) strategy, as measured by preparation time and tendency to correct errors immediately or after further text production, and (c) success as measured by accuracy to correct errors, provide information about the interaction with the TPSF in general and, more specifically, about mechanisms by which new writing technologies might constrain revision during writing.

3 Method

To answer the research questions above, we set up a controlled reading-writing experiment. Participants were invited to participate in a one hour experiment during which they had to take two short initial tests and completed two sets of reading-writing tasks in two different modes, that is, one only visual and one read aloud before the visual representation appeared.

The task consisted of a set of sentences that were presented to the participants to provide a new context. After every sentence the participants had to click the 'ok' button, to indicate that they had finished reading the sentence. A subclause of the previous sentence was then presented as TPSF in another subordinate causal structure, and the participants were prompted to complete the sentence. Since this research project is already described in great detail in chapter 5, we will only provide a summary of the method in this chapter. For a full description we refer to chapter 5. In short, the experimental session consisted of five parts: counting span test, baseline reaction time test, first part of reading-writing test, second part of reading-writing test and a questionnaire.

3.1 Participants

Sixty students participated in this experiment. The students all had Dutch as their first language and were between 18 and 22 years old.

3.2 Materials

The main part of the experiment consisted of a reading-writing task. The participants had to read and complete 60 short sentences. They first read a short sentence which provided a context for the next sentence that had to be completed in the next step of the procedure.

Example:
 Context: Because it has rained, the street is wet.
 Correct TPSF: The street is wet, because ...
 Incorrect TPSF: The streert is wet, because ...

All experimental sentences that contained deficiencies were built according to a causal coherence relationship. All the materials were presented in Dutch. In 24 out of the 60 sentences that were used in the experiment, we varied four types of errors to construct the deficient TPSF based on error span, mode of writing and lexicality of the errors (Table 2).

Table 2. Classification of errors

Category	Error span	Type of error	
		Input mode	Lexicality
SR Large	large: > 2 characters	only in speech recognition	existing words
SR Small	small: ≤ 2 characters	only in speech recognition	existing words
SR Keyboard Small	small: ≤ 2 characters	in speech recognition and keyboard & mouse	existing words
Keyboard Small	small: ≤ 2 characters	only in keyboard & mouse	non-existing words

3.3 Design

The experiment employed a 2 (experimental condition speech versus non-speech) by 2 (two sets of sentences) within-subjects design (see Table 3). We constructed two sets of sentences in which an equal number of errors were distributed in a comparable way.

Also the kind of errors was equally varied. The order in which these sets of sentences were presented to the participants was counterbalanced in the design of the experiment. Also the sequence of how the sentences were offered was varied (non-speech and speech).

Table 3. Design of the experiment

Number of group	Order of experimental condition (speech, non-speech and set of sentences)	
Group 1	non-speech set of sentences 1	speech set of sentences 2
Group 2	non-speech set of sentences 2	speech set of sentences 1
Group 3	speech set of sentences 2	non-speech set of sentences 1
Group 4	speech set of sentences 1	non-speech set of sentences 2

3.4 Dependent variables

Five dependent variables were derived from the logging data of the experiment (TPSF-program and Inputlog):

1. Reaction time: the reaction time was defined as the time that passed between the moment when the auditory probe (beep) was given and the moment the button was pressed.
2. Preparation time: the preparation time was defined as the time that passed between the moment the context screen was closed and the first mouse click to position the cursor in the TPSF screen, either to complete the sentence, or to correct an error in the TPSF.
3. Production time: the production time was defined as the period between the moment when the screen with the context sentence was closed and the moment when the screen with the TPSF to be completed was closed.
4. Delayed error correction: for every sentence we logged whether the cursor was initially either positioned within the TPSF clause that was presented as a writing prompt (immediate error correction), or after the clause (delayed error correction).
5. Accuracy: the accuracy represents the percentage of sentences with a (manipulated) error that was rewritten correctly.

4 Data analysis

In this section we describe the multilevel analyses that we performed. As stated in the introduction, multilevel analysis is an adequate statistical method to describe hierarchically structured data (De Maeyer & Rymenans, 2004; Goldstein, 1995; Hox, 2002; Snijders & Bosker, 1999; Van den Bergh & Rijlaarsdam, 1996). In this experiment each participant completed a set of sentences. In other words, sentences are 'nested' into participants. Multilevel analysis enables us to control for differences between participants. Participants might have varying correcting strategies, and in this study we expect them to be influenced by the differences in error types. Multilevel analysis helps us to define whether the participants are the most decisive factor in correcting various errors, or whether the error types have a (more) decisive influence on error correction.

We conducted the multilevel analysis in three steps (see Figure 2). In the first step

we estimated the so-called ‘zero model’ to gain insight on the one hand in the variance between participants and on the other hand in the variance between sentences. These variances provide information about the distribution of the variance of participants as opposed to the variance of sentences. By calculating the intra class correlations (ICC) we can evaluate the relative amount of variance that can be attributed to the TPSF sentences themselves rather than the participants.

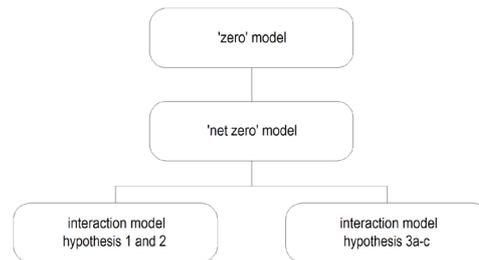


Figure 2. Flow of the multilevel analysis.

In the next step we estimated the ‘net zero model’ in which we have integrated the variables that characterize the participants. In this step we have analyzed whether specific characteristics of the participants are of influence that need to be taken into account in the further analyses. Based on these models we get insight in the unique variance between sentences, after correction for higher level variables. It is this unique variance that we want to explain in accordance with our hypotheses.

In the final step we used interaction models (see section 4.2). The first type of interaction model is related to the experimental condition. Sentences that are offered only visual, versus sentences that are also presented via speech are compared respectively for correct sentences and for incorrect sentences. The second type of interaction model compares the various error types that the writers are confronted with.

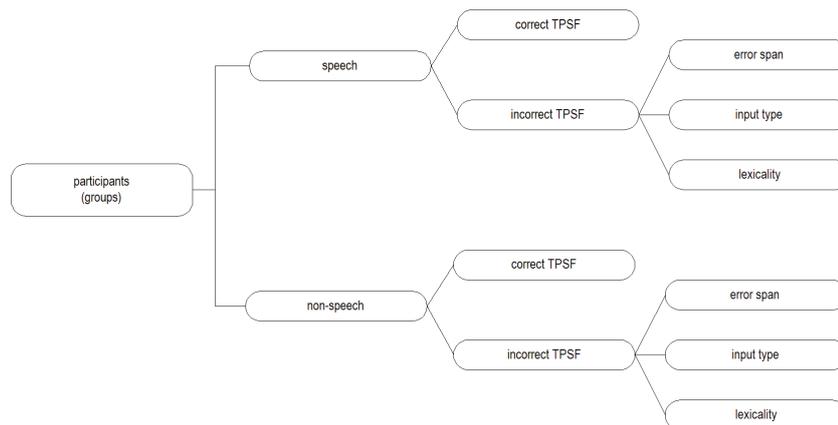


Figure 3. Hierarchical model describing the participant’s (level 2) and the sentence (level 1) characteristics used in the multilevel analysis.

In Figure 3 we describe the hierarchy between the two levels that were used in the multilevel analysis: the participant level (including participants' characteristics like, for instance, sex) and the sentence level (including characteristics of the sentence, i.c. the two experimental conditions, the correctness and the three error types).

4.1 Characteristics of participants and sentences

In the analyses we estimate the effects of independent variables at participant and sentence level. The following explanatory variables were isolated for the participants:

1. Sex: male or female.
2. Counting span: sum of factor accuracy/speed of counting span. The results of the counting span are adapted to integrate in the model. Therefore, we have calculated the accuracy of the task completion in the different stages as a percentage and combined that score with the speed in which the participants provided the correct number of counted blue squares. Since the counting span consisted of four degrees of difficulty, we have calculated a weighted factor for the highest three degrees⁴. The formula is written as: $(\text{accuracy degree 2}/\text{time degree 2}) * 1 + (\text{accuracy degree 3}/\text{time degree 3}) * 3 + (\text{accuracy degree 4}/\text{time degree 4}) * 6$. This factor describes a variation of the counting span measure between the participants.
3. Groups: descriptive classification of preference to delay error correction. We analyzed the preference of writers to correct errors in the TPSF immediately versus to delay error correction. Therefore, we have calculated the mean percentage of the cursor position in the incorrect sentences. We have opted to describe four groups who show a distinct preference when they perform error correction. The median percentage of the cursor position is 88. We used this value as the breaking point to distinguish the so called 'immediate' from the 'delayed' group. Practically this immediate group has values ranging from 46 to 88%. The median of the immediate group is 65 and this again is used as the breaking value to split the immediate group into 'immediate' (46%-64%) and 'immediate medium' (65%-88%). The variance within those two groups is still relatively high. The delayed groups, on the other hand, are quite coherent. The median and breaking point for this group is 98%. The participants prefer to complete almost every sentence before they switch to error correction. The group 'delay medium' ranges from 92% to 96%. The last group; the so called 'delay group', is almost unanimous in their preference: first they complete the text and then they correct the error.
4. Median baseline (this measure is only included for the variable reaction time). The median reaction time of the baseline test for each participant (cf. initial reaction time test) was added to the model as a residual to take the individual differences in reaction into account.

⁴ The first level was excluded from the analyses because this level did not differentiate between participants.

<p>Participants</p> <ul style="list-style-type: none"> - Sex (male or female) - Counting span - Groups immediate versus delayed - Median baseline reaction time <p>Sentences</p> <ul style="list-style-type: none"> - Auditory condition (speech or non-speech) - Correctness (correct or incorrect sentences) <p>Data focus</p> <ul style="list-style-type: none"> - Experimental condition (mode of presentation): correct (H1) and incorrect sentences (H2) - Error type based on span (H3a) - Error type based on input type (H3b) - Error type based on lexicality (H3c) <p>Dependent variables describing cognitive effort</p> <ul style="list-style-type: none"> - Reaction Time - Preparation Time - Production Time - Delayed correction - Accuracy of correction (quality)

Figure 4. Variables used in the multilevel analyses.

The sentences might be influenced by the auditory condition (speech or non-speech) and by correctness.

Figure 4 shows an overview of the variables that we have used in the analyses. The data are further elaborated in the hypothesis section (section 2) and the variables that measure cognitive effort are described in the method section (section 3.4).

4.2 Models

Each model consists of three types of parameters: an estimated mean for that specific variable (fixed β_0), a characterization of each writer as a deviation of the mean (random: u_{0j}), and a characterization of each TPSF sentence as it was presented to the writer as a deviation from the mean of that writer (random e_{0ij}). In general the zero model, estimating the effect of the influence of the experimental condition or the effect of error types can be presented as follows:

$$Y_{ij} = \beta_0 X_0 + [u_{0j} + e_{0ij}] \quad [1]$$

In this model Y_{ij} represents the estimated value of a response variable (i.c. referring to a measure of cognitive effort: reaction time, preparation time, or production time) for every sentence (i) nested within a participant (j).

The model consists of a fixed part and a random part (between square brackets). The fixed part in the zero model contains only one explanatory variable X_0 with a constant value of 1. As a consequence, only one regression coefficient is estimated (β_0), an estimate of the overall mean value for the dependent variable. The random part consists of two parts. The first scores describe the individual deviation on the participants' level related to the overall mean ($[u_{0j}]$) and the second series of resid-

ual scores describes the deviations at the sentence level as a deviation from the mean of that writer ($[e_{oj}]$). The variances of both residual scores can be interpreted respectively as the estimated variance on the level of the participant (level 2) and the sentence (level 1).

Because in this study we used two types of variables to describe cognitive effort – continuous and binominal response variables (Rashbash, Steele, Browne, & Prosser, 2004) – we could not only use univariate multilevel regression models as described above [Formula 1]. The binominal response variables (i.c. cursor position and accuracy) had to be fitted into another type of models, the so called logit multilevel regression model (Goldstein, 1995). It can be written as follows:

$$\text{logit}(\pi_{ij}) = \beta_0 + u_{oj} \quad [2]$$

In this model π_{ij} is the probability that person j gets a score of 1 on the dependent variable for sentence i . We used the logit link function because this enables us to translate estimates in odds ratios and calculate estimated probabilities based on these odds ratios.

In the next step we also added the variables that characterize the participants to the zero models (i.c. sex, counting span, median base line reaction time and preference for delayed correction). This resulted in so called ‘net zero models’. This procedure enabled us to analyze which characteristics significantly influenced the value of the estimated means of the dependent variable. These characteristics are added in the interaction models to test our hypotheses. By doing this, we take into account possible disturbing person characteristics while testing our hypotheses.

To test the first two hypotheses we compared the mean values on our dependent variables for the mode of presentation and correctness of the TPSF (i.c. auditory condition [speech or non-speech] and correctness [correct or incorrect sentences]). Besides differences in means between the four sentences types, it is possible that the variances between sentences differ for these four sentences types. In other words, it is possible that we encounter bigger differences between incorrect sentences accompanied with speech than between correct sentences accompanied with speech. To translate these assumptions in a model we combined the mode of presentation and correctness of the TPSF into four dummy variables ($D_1 - D_4$) identifying the characteristics per sentence. The interaction model (for the continuous variables) can be written as follows:

$$Y_{ij} = \beta_1 * D_{1ij} + \beta_2 * D_{2ij} + \beta_3 * D_{3ij} + \beta_4 * D_{4ij} + (\beta_{5j} * X_{1j} + \dots + \beta_{6j} * X_{6j}) + [u_{1j} + u_{2j} + u_{3j} + u_{4j} + e_{1ij} + e_{2ij} + e_{3ij} + e_{4ij}] \quad [3]$$

$D_1 =$ non-speech - not correct
 $D_2 =$ non-speech - correct
 $D_3 =$ speech - not correct
 $D_4 =$ speech - correct

In this model β_1 through β_4 are the estimates of the mean value on the dependent variable. For both the mode of presentation and correctness of the TPSF we estimate a residual at the person level (u_{1j} through u_{4j}) and at the sentence level (e_{1ij} through e_{4ij}). Finally these models also contain variables at the respondent level (X_{1j} through X_{4j}), for which the effect sizes are estimated (β_{5j} through β_{8j}).

To test the third hypothesis a comparable interaction model was built. In this model the different error types were added to the model to estimate the effect of error span, input type and lexicality (see also Table 2). Because every error type was presented to the participants in both the speech and the non-speech condition, the model can be represented as follows:

$$Y_{ij} = \beta_1 * D_{1ij_non-speech} + \beta_2 * D_{2ij_non-speech} + \beta_3 * D_{3ij_non-speech} + \beta_4 * D_{4ij_non-speech} + \beta_5 * D_{5ij_speech} + \beta_6 * D_{6ij_speech} + \beta_7 * D_{7ij_speech} + \beta_8 * D_{8ij_speech} + (\beta_{9j} * X_{1j} + \dots + \beta_{12j} * X_{4j}) + [u_{1j} + u_{2j} + u_{3j} + u_{4j} + e_{1ij} + e_{2ij} + e_{3ij} + e_{4ij}] \quad [4]$$

D_1 and D_5 = SR Large error resp. for non-speech and speech

D_2 and D_6 = SR Small error resp. for non-speech and speech

D_3 and D_7 = SR Small | Keyboard error resp. for non-speech and speech

D_4 and D_8 = Keyboard Small (non-existing) error resp. for non-speech and speech

We have chosen not to model separate variances for our 8 different conditions, but for four conditions. The reason therefore lies in a loss of precision of the estimates given the number of parameters to estimate in that case.

In the following section we present the results of the multilevel analyses used to test the three hypotheses put forward in this study.

5 Results

In a first step of the multilevel analysis procedure we have explored the intra class correlation for each of the dependent variables we used to operationalize the memory load. The participants and the imbedded sentences differed for reaction time, preparation time and production time. The zero model estimates an intra class correlation on reaction time of 23% on the participant level. Since this value is significant it is advised to conduct multilevel analyses on the data. In the next step we added the various characteristics of the participants and the sentences to the zero model as to construct the net zero model.

The participants' characteristics sex, counting span and group do not have a significant effect on reaction time, but the median baseline that we have calculated for each participant on the basis of the initial reaction test showed to be significant (Estimated difference = 0.768, $SE = 0.240$). That is why we have taken into account the baseline of the reaction time in all analyses on reaction time. If we take the median baseline into account then the ICC for participants slightly decreases from 23% to about 20%. Table 4 shows the estimated means for the intercepts and the par-

participants' characteristics that influenced the variables. The characteristics of the participants that differed significantly are taken into account in the further analyses.

Table 4. Parameter estimates of intercept, participants' characteristics and the intra class correlations for reaction time, preparation time and production time

	Reaction time		Preparation time		Production time	
	Zero model	Net zero model	Zero model	Net zero model	Zero model	Net zero model
	<i>Est.</i> (<i>SE</i>)					
<i>Fixed Part</i>						
Intercept	836.876 (22.115)	450.253 (122.743)	1650.333 (79.011)	3346.116 (217.463)	17865.79 (539.675)	19317.83 (747.155)
Sex	--	--	--	--	--	-2722.076 (1023.024)
Counting span	--	--	--	--	--	--
Immediate/delayed error correction	--	--	--	-21.154 (2.625)	--	--
Median baseline RT	--	0.786 (0.240)	--	--	--	--
<i>Random part</i>						
Participant level	27218 (5358)	22948 (4578)	329699 (68422)	135017 (32846)	16663120 (3195115)	14816740 (2844342)
Sentence level	89847 (2551)	89847 (2551)	2141602 (57184)	2141603 (57187)	38754860 (1034494)	38754940 (10355526)
ICC	0.23	0.20	0.13	0.06	0.30	0.28

ICC = intra class correlation, *Est.* = parameter estimate, *SE* = standard error

The estimated mean reaction time after taking into account participant characteristics is 450ms. The variance is accounted for by 20% by the participants and by 80% by the sentences. Preparation time is influenced positively at participant level by the preference to correct errors immediately or to delay error correction (estimated mean = 0.786, *SE* = 0.240). Production time is influenced by the participants' sex. The female writers have significantly shorter production times than the male writers (estimated mean = -2722.076, *SE* = 1023.024). The intra class correlations are respectively 20%, 6% and 28% accounted for by the difference at the participant level. In other words, differences in the dependent variables are accounted for in about $\frac{3}{4}$ by differences in sentences. Only $\frac{1}{4}$ is accounted for by the participants. Given the fact that the estimates for the variances at the participant level follow a Student-t distribution we can conclude that they all are significantly different from zero. In other words, a significant part of the variances between sentences can be attributed to the participant level, making multilevel analysis necessary.

Given that delayed error correction and accuracy are based on multilevel logit regressions, we conducted a Wald test (Rashbash et al., 2004) for the variance at participants' level to see if multilevel analyses are necessary. According to the Wald test on the variances at the participant level, delayed error correction needs to be analyzed with a multilevel model ($\chi^2=22,02$, $p < .001$). The Chi-square for accuracy is

3.50, $p < .06$. Since this value is at the edge of the .05 significance level, we decided not to exclude this variable from the multilevel analyses. Table 5 shows the estimated means for the intercepts and the participants' characteristics that influenced the variables.

Table 5. Parameter estimates of intercept, participants' characteristics and intra class correlations for delayed error correction and accuracy

	Delayed error correction		Accuracy	
	Zero model	Net zero model	Zero model	Net zero model
	<i>Est.</i> (<i>SE</i>)	<i>Est.</i> (<i>SE</i>)	<i>Est.</i> (<i>SE</i>)	<i>Est.</i> (<i>SE</i>)
<i>Fixed part</i>				
Intercept	2.037 (0.154)	1.190 (0.465)	1.657 (0.089)	1.087 (0.328)
Sex	--	--	--	--
Counting span	--	--	--	--
Delayed correction	--	--	--	0.007 (0.004)
Median baseline RT	--	--	--	--
<i>Random part</i>				
Participant level variance	1.222 (0.260)	1.179 (0.266)	0.161 (0.086)	0.139 (0.328)

Est. = parameter estimate, *SE* = standard error

Delayed error correction has an estimated mean of 1.190. None of the participants' characteristics is of influence on the preference to position the cursor in the TPSF or in the text completion part. The quality (accuracy) of the texts is influenced by the participants characteristic to prefer error correction immediately or to delay correction (delayed correction = 0.007, $SE = 0.004$). Although the significance level of .05 is not reached, this characteristic is also taken into account into the further modeling on the hypothesis level.

5.1 Effect of mode of error presentation on the interaction with correct TPSF

To describe the interaction between mode of presentation and the TPSF we have built 'interaction models'. The first formal interaction model [3] describes the effect of offering the correct or incorrect TPSF in an auditory and a visual condition (speech vs. non-speech). It comprises hypothesis 1 and 2. For the effect of the auditory condition on the interaction with corrected sentences we compared the estimates of parameters of β_{1ij} and β_{2ij} .

In the first hypothesis we expected that the addition of speech in the auditory condition could have a positive effect on the memory load of writers. Table 6 shows the results for the speech versus non-speech condition on reaction time, preparation time and production time for correct sentences.

Table 6. Parameter estimates for the correct TPSF in the speech and non-speech condition⁵

	Speech		Non-speech		χ^2	sign.
	Est.	SE	Est.	SE		
Reaction time	465.46	73.20	464.28	72.26	0.00	-
Preparation time	1493.14	79.82	2453.32	114.48	72.29	0.00
Production time	16506.06	538.66	17568.11	691.86	1.88	-

The reaction time on the secondary task did not differ in the condition that the TPSF was also offered via speech. At first sight this is contrary to our predictions. We assumed that processing text via speech would require less cognitive resources and therefore result in faster reaction times. In both conditions the participants only needed about 465 milliseconds to respond (after correction of the median baseline RT). In general, we still assume this hypothesis needs reconsideration. The writing task in this study is rather easy. Writers need to complete one sentence at a time. Therefore, in a follow up study we have varied the task difficulty which resulted in a significant effect on cognitive load. Offering complex context showed to be significant more difficult than to produce a text based on a less complex context, which was represented in significantly less accuracy and larger production times (Quinlan, Loncke, Leijten, & Van Waes, in preparation). We assume that in this current study the free working memory to respond to a secondary task is comparable for the various writers, because they all have enough free resources left. Another explanation could be that the nature of the experimental task might have influenced the participants' behavior in dealing with correct sentences. Knowing that the TPSF might contain errors, the participants might have been looking more carefully to the correct sentences than they would in a normal situation.

If however, we consider preparation time as another measure of cognitive effort, that is, the time it takes to start with the completion task, then we do see the difference that we expected in the first hypothesis. In the condition where writers first heard the TPSF read out aloud before it was shown on the screen. They needed significantly less preparation time (speech = 1.49 seconds versus non-speech = 2.45 seconds preparation time ($\chi^2(df=1) = 72,29, p < .001$). The difference in preparation time also affects the production time to complete the sentence. However it does not lead to a significant difference in the total production time. Text production is more or less of equal duration in the speech and the non-speech condition.

5.2 Effect of mode of error presentation on the interaction with incorrect TPSF

The second hypothesis focuses on incorrect text. In general, we expect it to be easier to compare the mental representation of the TPSF with only the visual feedback on the screen, than to compare it with visual and auditory feedback. In short, to compare

⁵ Because in this table we only report data that are related to 'correct' sentences and consequently no errors in the TPSF needed to be corrected, we do not report the values for cursor position and accuracy.

two things is easier than comparing three things. Above that, the auditory information of the TPSF probably leads to a focus on text production. In Table 7 the interaction with incorrect text is shown for the speech and the non-speech condition.

Table 7. Parameter estimates for the incorrect TPSF in the speech and non-speech condition

	Speech		Non-speech		χ^2	sign.
	Est.	SE	Est.	SE		
Reaction time	448.25	72.67	470.84	72.70	0.48	0.49
Preparation time	1502.29	79.95	2784.07	145.80	78.49	0.00
Production time	20863.77	596.21	22001.50	710.02	1.88	0.19
Delayed error correction	2.72	0.39	0.67	0.20	22.15	0.00
Accuracy	1.48	0.18	1.50	0.62	0.02	0.89

If writers are confronted with errors in the TPSF, they still respond in the same way to the secondary task. In general, offering the text via speech has no influence on the reaction time when reading a TPSF clause in which an error occurs.

However, the variance between sentences does differ significantly (variance speech = 104197 ($SE = 5753$), non-speech = 63553, ($SE = 3523$) ($\chi^2(1)=36,30$, $p < .001$); the variance between participants does not. Sentence that are offered also via the auditory mode show a significant higher variance. As in the correct sentences, writers need less time to reflect on what their first writing action will be. The preparation time is significantly lower if the TPSF is also offered via speech (respectively 1.50 seconds compared to 2.78 seconds). When we compare the behavior of writers who prefer to correct the errors immediately or not, we see a significant difference in the use of preparation time. The group of writers that prefers to correct the error in the text immediately in the non-speech condition needs three times more preparation time than the group of writers that prefer to delay error correction in the speech condition (preparation time non-speech immediate = 2.58 seconds versus preparation time speech delay = 0.87 seconds) (see Appendix Table 12). These are the two most diverse patterns related to preparation time.

The production time is comparable in both conditions (about 20 seconds). The cursor position in incorrect sentences is significantly influenced by the spoken TPSF. The chance that writers prefer to complete the sentence first is in the speech condition 31% higher than that they prefer to first correct the error in the TPSF (speech = 94% vs. non-speech = 66%)⁶. The odds of giving priority to text completion in speech is $15.24/1.95 = 7.81$ times the odds compared to the situation in which the TPSF is not presented with speech⁷.

The influence of the speech condition to either delay the error correction or not, seems to be different depending on the participants' general preference to postpone error

⁶ The Chances were calculated on the basis of the reported beta scores as follows: $\text{Chance}[X] = 1/(1+(\exp(-\text{betascore}(X))))*100$

⁷ The Oddsratios were calculated on the basis of the reported beta scores as follows: $\text{Oddsratio}[X] = \exp(\text{betascore}(X))$

correction or not. Figure 5 shows the preference for delayed error correction of the four writer groups (immediate, immediate medium, delayed medium, delayed). The graphical representation clearly shows the general tendency to delay the error correction when the TPSF is dictated first (speech-condition). Speech clearly reinforces the writers' preference to delay the error correction and to prioritize the completion of the sentence. However, Figure 5 also shows that the behavior within the non-speech condition is much more diverse than in the speech condition. In the former condition the group that – relatively spoken – is least eager to delay error correction in the speech condition (about 77%), delays three times less errors (about 24%) in the non-speech condition (see also appendix Table 14). On the other hand there are participants that hardly solve any errors immediately, certainly not in the speech condition and hardly in the non-speech condition (i.c. both the 'delayed' groups; max 10%). The four groups are all characterized by a specific preference to delay error correction or not and their behavior differs significantly⁸. On the basis of these observations we can conclude that participants behave differently with respect to their preferred strategy to either delay errors or not, and that this behavior is significantly influenced by the occurrence of an auditory representation of the TPSF.

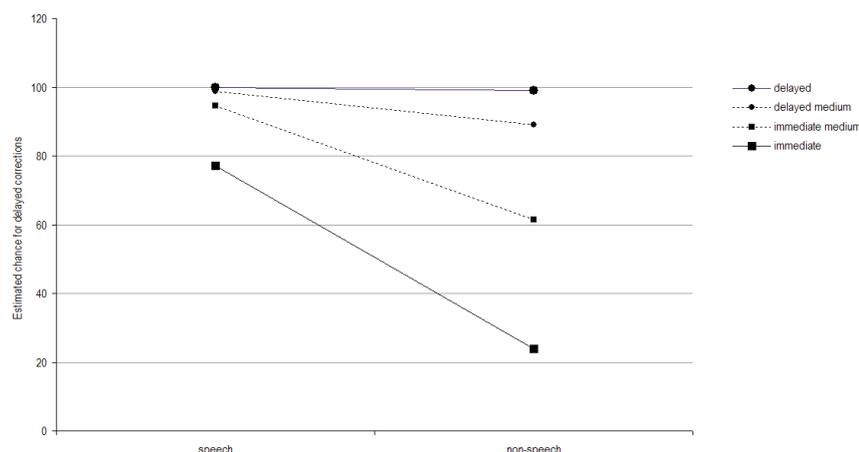


Figure 5. Preference for delayed correction of errors in the speech and non speech condition (groups).

In general, speech does not influence the accuracy of the correction significantly, that is, the participants did not detect and correct more errors in the TPSF in either condition. However, writers that prefer to delay error correction seem to be significantly

⁸ We refer to Table 14 in the Appendix for more details. Estimated differences: 'immediate medium' versus 'delayed medium': 1.62 and $\chi^2(1)=25.95$, $p < .001$; 'delayed medium' versus 'delayed': 2.94 and $\chi^2(1)=10.58$, $p < .001$; 'immediate medium' versus 'delayed': 4.11 and $\chi^2(1)=13.78$, $p < .001$.

more precise than writers who prefer to position their cursor first in the TPSF to start error correction immediately. The group that prefers to prioritize text production succeeds in 87% to correct the TPSF accurately and the group that prefers to revise first has an accuracy score of 81% (estimated difference: delay (delay + delay medium) = .436, $SE = .171$).

The addition of offering the TPSF also via an auditory channel affects the preparation time writers need to either complete the sentence or to start error correction, both in the ideal world with no errors in the TPSF and in texts that do contain deficiencies. Writers also have a strong tendency to continue text production if the TPSF is also presented in the speech mode. The next paragraphs describe the influence of error span, input type and lexicality on error correction.

5.3 Effect of error span

The interaction model of the four error types that takes the interaction between mode of presentation and error types into account for the variable reaction time are described in [4]. The estimated parameters and standard errors for the five dependent variables (reaction time, preparation time, production time, delayed error correction and accuracy) can be found in the Appendix.

For the first hypothesis on error types we compared the effect of large speech recognition errors with small speech recognition errors on the memory load and the interaction behavior with the TPSF. For instance, the estimated means of the reaction time for large and small errors were compared in combination with the speech condition (SR Large Speech_{ij} + SR Small Speech_{ij}). We expect that large error spans can be solved more efficiently than small errors. Table 8 shows the general behavior and the behavior in the speech and non-speech condition of writers in the interaction with large and small speech recognition based errors in the TPSF.

Table 8. Parameter estimates for Small and Large speech recognition errors

	SR Large Error		SR Small Error		χ^2	sign.
	Est.	SE	Est.	SE		
Reaction time	443.16	77.85	366.44	75.75	4.70	0.03
Speech ^o	394.51		344.87		1.42	0.23
Non-speech	493.55	80.32	389.14	76.42	6.27	0.01
Speech*error ^{oo}					1.55	0.21
Preparation time	2889.49	165.64	2158.87	91.02	18.90	0.00
Speech	1805.68		1611.96		0.78	0.38
Non-speech	4008.86	211.16	2712.77	106.74	34.45	0.00
Speech*error ^{oo}					12.73	0.00
Production time	26634.99	860.61	19981.18	628.64	46.35	0.00
Speech	25412.83		19788.82		25.89	0.00
Non-speech	27895.02	971.37	20172.10	685.67	48.36	0.00
Speech*error ^{oo}					4.04	0.04

Delayed error correction	1.47	0.18	1.60	0.20	0.26	0.61
Speech	3.06		2.72		0.42	0.52
Non-speech	0.69	0.22	0.98	0.23	0.88	0.35
Speech*error ^{°°}					1.44	0.23
Accuracy	2.15	0.28	0.98	0.18	17.88	0.00
Speech	1.98		1.04		7.25	0.01
Non-speech	2.37	0.36	0.92	0.21	14.29	0.00
Speech*error ^{°°}					1.17	0.28

[°] The values for speech are calculated based on the values of the estimate of the non-speech parameter.

^{°°} The significance for the interaction terms are evaluated by comparing both differences between large and small errors.

The estimated reaction time is significantly longer with large errors than with small speech recognition errors (Large Error = 443ms versus Small Error = 366ms). The addition of speech causes a significant decrease of difference in reaction time (estimated difference: Large Error: -99,047⁹; $SE = 39.017$ versus Small Errors: -44.267; $SE = 20.393$). Large errors distract more than small errors, but not when the visual TPSF is preceded by dictation.

The reaction time also shows a significant difference in variance on the sentence level, both within (estimated variance SR Large = 134578 ($SE = 11099$), SR Small = 37427 ($SE = 3056$) and between ($\chi^2(1) = 71.22$, $p < .001$) the error categories. The latter indicates that the variability within the category of sentences with larger errors is significantly larger than the variability of the sentences with a small error. So, the reaction time varies significantly more when writers are confronted with the different TPSF clauses that contained a large error. The same holds for preparation time (estimated variance SR Large = 7167918 ($SE = 538502$), SR Small = 1297077 ($SE = 106079$), ($\chi^2(1) = 114.42$, $p < .001$) and production time (estimated variance SR Large = 69411880 ($SE = 5734578$), SR Small = 27092000 ($SE = 2215746$), ($\chi^2(1) = 47.39$, $p < .001$). So, the large errors show a larger variance within sentences (nested within participants) than the small errors¹⁰.

In general, the error span does not interact with the mode of presentation, speech versus non-speech. Figure 6 shows the estimated means of reaction time for large and small errors in both conditions.

As we can see in Figure 6, large errors result in significantly slower reaction times, either when the TPSF is also presented auditory or not (speech vs. non-speech condition). The presence of large errors seems to distract the participants more intensively from their 'reading to produce' task and consequently increases the cognitive load for the writers in that stage of the writing process, especially in the non-speech condition. On top of that, the speech condition also lowers the cognitive effort significantly in

⁹ We refer to appendix Table 11 for the detailed values of the parameters.

¹⁰ In the pretest on the categorization of the error types we have controlled for differences between error types, but not on similarity within error types.

these instances. The preference to correct errors immediately or to delay error correction is not of influence on the reaction time.

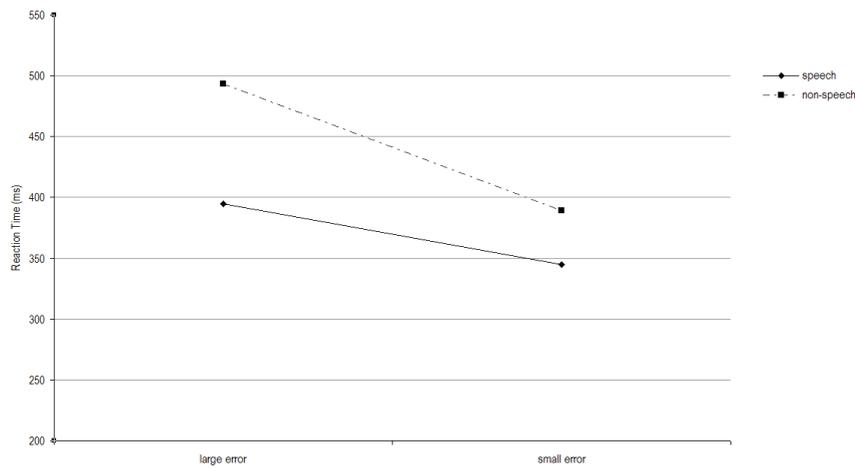


Figure 6. Estimated means of Reaction Time for large and small errors in the speech and the non-speech condition (*ms*).

The preparation time is significantly longer when writers are confronted with a large error in the TPSF (Table 8). In the non-speech condition the preparation time to start correcting large errors (or completing a sentence on the basis of a TPSF with a large error) takes significantly longer than for sentences with small errors in the TPSF ($\chi^2(1) = 19.90, p < .001$).

The total production time (completion and correction) of sentences with a large error in the TPSF also takes significantly longer than for those with a small error (Large Errors = 27 seconds, Small Errors = 20 seconds). This is probably partly explained by the larger amount of text production that needs to be performed by the writers who have to correct a larger error, but of course, also the level of distraction might take extra time to (mentally) reconstruct the correct TPSF. This result is most explicit in the non-speech condition ($\chi^2(1) = 48.36, p < .001$). If writers need to solve a large error in the TPSF, the prior dictation of the text seems to facilitate the writing process and significantly shortens the total text production time (estimated difference for speech condition with Large Errors: -2482.195 ; $SE = 888.069$). In other words, text production in these instances is more fluent if the TPSF is also provided via the auditory channel. The interaction effect is significant. If we take a closer look at the small errors, then speech does not have any effect on the production time.

In general, the size of the error does not influence the preference of writers to start with error correction or to continue with text production. However, the speech condition does influence this preference: additional analyses on the difference between speech and non-speech show a significant estimated difference between the conditions (Large Errors = 2.374 ; $SE = 0.395$ and Small Errors = 1.738 ; $SE = 0,353$). So, if the

TPSF is offered via speech, the chance that writers prefer to delay error correction and complete the sentence first is higher.

Finally, large errors are also significantly better solved than small errors, and this holds for both conditions. For instance, the odds of correcting a large error is $8.58/2.65 = 3.23$ times the odds compared to correcting a small error (probability for correct correction of Large Errors: 89.56% versus Small Errors: 72.61%).

Moreover, a more detailed analysis in which also the preference to either delay the error correction or not is taken into account, shows that writers who tend to delay more errors have a significant higher chance to correct more errors in the TPSF (estimated distance mean 0.470; $SE = 0.158$; overall chance score for delayed group: 87% versus 81% for the group that tends more to immediate correction).

5.4 Effect of input type

In the second hypothesis on error types we compare small speech recognition errors with small keyboard errors. We expected that small errors that can occur in writing with keyboard & mouse can be solved more efficiently than small errors that can occur in writing with speech recognition, because writers are more familiar with the first category.

In a first glance, the accuracy scores in Table 9 confirm this hypothesis. However, the other variables that are more directly related to cognitive effort do not seem to support this finding.

Table 9. Parameter estimates for small speech recognition errors and keyboard errors

	SR Small Error		Keyboard Small Error		χ^2	sign.
	Est.	SE	Est.	SE		
Reaction time	366.44	75.75	414.47	76.38	2.23	0.14
Speech°	344.87		412.10		3.48	0.06
Non-speech	389.14	76.42	417.99	77.49	0.64	0.42
Speech*error°°					1.39	0.24
Preparation time	2158.87	91.02	2201.66	90.19	0.20	0.65
Speech	1611.96		1780.45		1.91	0.17
Non-speech	2712.77	106.74	2636.01	102.99	0.40	0.53
Speech*error°°					2.43	0.12
Production time	19981.18	628.64	19551.00	608.67	0.32	0.57
Speech	19788.82		19137.29		0.59	0.44
Non-speech	20172.10	685.67	19967.13	651.27	0.06	0.81
Speech*error°°					0.39	0.53
Delayed error correction	1.60	0.20	1.17	0.17	2.73	0.10
Speech	2.72		2.55		0.13	0.72
Non-speech	0.98	0.23	0.39	0.21	3.52	0.06
Speech*error°°					0.77	0.38

	SR Small Error		Keyboard Small Error		χ^2	sign.
	<i>Est.</i>	<i>SE</i>	<i>Est.</i>	<i>SE</i>		
Accuracy	0.98	0.18	2.11	0.22	26.92	0.00
Speech	1.04		2.15		12.69	0.00
Non-speech	0.92	0.21	2.08	0.28	14.56	0.00
Speech*error ^{°°}					0.01	0.92

[°] The values for speech are calculated based on the values of the estimate of the non-speech parameter.

^{°°} The significance for the interaction terms are evaluated by comparing both differences between large and small errors.

The errors that are related to small speech recognition errors versus small keyboard based errors do not cause a different pattern for reaction time, preparation time, production time nor delayed error correction. The accuracy score, referring to the number of errors that were corrected successfully, is significantly better for the small keyboard errors. The chance that writers correct the small keyboard error is 11% higher than to correct the small speech recognition error in the TPSF (SR Small = 79%, Keyboard Small = 90%). In other words, the odds of correcting the SR Small error is $2.56/8.26 = 3.11$ times the odds compared to the correction of the small keyboard error (resp. 89.20 vs. 72.61). No interaction effects were found between this type of error and the auditory condition.

5.5 Effect of lexicality

In the final hypothesis on error types we focused on the effect of lexicality as a semantic characteristic of errors to compare the interaction with 'non-existing word' and 'existing-word' errors in the TPSF. We compared small errors that could either be caused by speech recognition software and keyboard based word processing with small errors that could only be caused by writing with keyboard. This latter type of error resulted in non-existent words, while the former error type consisted only of existing words. We expected that the non-existing words would be corrected more efficiently. Table 10 shows the parameter estimates for both error types.

The difference between existing words and non-existing words only causes a difference related to the accuracy of the error correction. Writers correct the errors that resulted in non-existent words significantly better than the errors that resulted in existent words. This finding that 'existent word' errors are harder to detect is consistent with literature on the topic of error correction of non-words (Hacker, 1994). However, the cognitive effort it takes to solve these errors does not seem to vary significantly which is comparable to the result of the previous hypothesis. Neither did we find any interaction effects.

Table 10. Parameter estimates for small speech recognition/keyboard errors versus small keyboard errors

	Existing words SR Keyboard Small		Non-existing words Keyboard Small		χ^2	sign.
	<i>Est.</i>	<i>SE</i>	<i>Est.</i>	<i>SE</i>		
Reaction time	425.92	77.06	414.47	76.38	0.11	0.74
Speech [°]	454.80		412.10		1.07	0.30
Non-speech	397.78	79.04	417.99	77.49	0.24	0.62
Speech*error ^{°°}					2.12	0.15
Preparation time	2233.53	87.31	2201.66	90.19	0.12	0.73
Speech	1737.05		1780.45		0.14	0.71
Non-speech	2740.57	101.48	2636.01	102.99	0.80	0.37
Speech*error ^{°°}					1.02	0.31
Production time	19654.95	568.19	19551.00	608.67	0.02	0.89
Speech	19118.92		19137.29		0.00	1.00
Non-speech	20190.78	614.80	19967.13	651.27	0.08	0.78
Speech*error ^{°°}					0.13	0.72
Delayed error correction	1.37	0.18	1.17	0.17	0.65	0.42
Speech	2.73		2.55		0.14	0.71
Non-speech	0.64	0.22	0.39	0.21	0.68	0.41
Speech*error ^{°°}					0.02	0.89
Accuracy	1.20	0.18	2.11	0.22	17.13	0.00
Speech	1.17		2.15		9.82	0.00
Non-speech	1.23	0.22	2.08	0.28	7.38	0.01
Speech*error ^{°°}					0.10	0.75

[°] The values for speech are calculated based on the values of the estimate of the non-speech parameter.

^{°°}The significance for the interaction terms are evaluated by comparing both differences between large and small errors.

In a final additional analysis we had a closer look at the differences in production time for the groups of participants that either sometimes prefer to delay certain errors or (hardly) do not. This analysis revealed a difference in production time for the four groups that vary in their preference to correct errors more or less immediately. This difference is the same for the three hypotheses on error type. The production time for the error type in which we compare small speech recognition/keyboard errors with small keyboard errors is the best candidate for a comparison. Therefore, we will illustrate the groups differences in production time by means of this latter comparison (Figure 7).

The result of this analysis is quite complex. If we compare the so called 'delayed' group with the 'immediate' group, no significant differences could be found. However, as Figure 7 shows, the production time of both the more extreme delayed group and the moderate immediate group slightly but significantly increases in comparison with the group that has the most explicit preference to correct errors immediately.

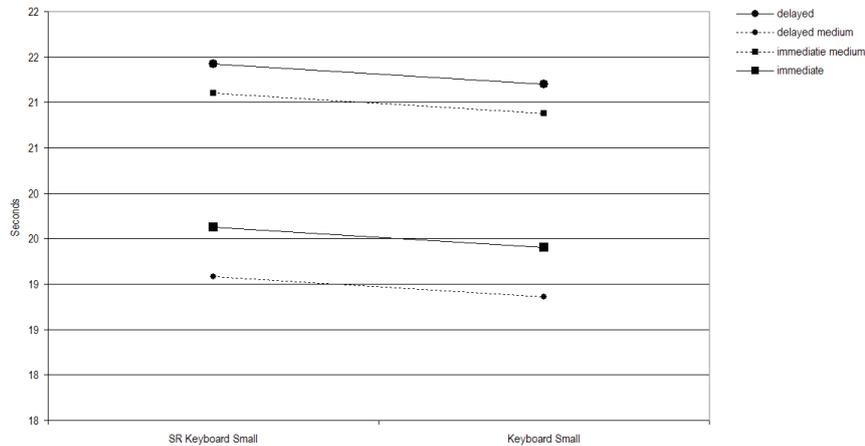


Figure 7. Estimated means of production time for error types SR Keyboard Small versus Keyboard Small (per group).

For none of the groups the difference between the different types of small errors is significant. On the basis of the data we gathered it's hard to explain these differences in production rate. The only thing we can say is that, in general, there is no difference in production efficiency between the groups with different (strategic) correction preferences. Nevertheless, a more refined categorization provides an indication of differences in production time that might be worthwhile to reflect upon in further research.

6 Conclusions and discussion

The present research isolated the effects of the writing mode that presented the TPSF (auditory vs. visual-tactile) from error span (large vs. small errors), input type (small speech recognition errors vs. small keyboard errors), and lexicality (existing vs. non-existing words). The data were explored via multilevel analysis. Although the focus of the study is mainly on error correction strategies, we have deliberately taken a step backwards and also considered the effect of the auditory channel solely in correct sentences. This enables us to focus on the effect of the presentation mode.

The isolated correct sentences did not show a significant positive effect on reaction time in the speech and non-speech condition when the TPSF was also presented auditory. However, the preparation time writers needed to complete the sentences did significantly drop. Therefore, the decrease of cognitive effort is not fully confirmed. An explanation could be that the nature of the experimental task might have influenced the participants' behavior in dealing with correct sentences. Because they knew that an error could occur in the TPSF, we might have created a situation in

which participants might have been looking more carefully to the correct sentences than they would in a normal situation, because they wanted to be sure that they had not overlooked the implemented error. In other words, it is possible that we have invoked an artificial evaluative, seeking reading behavior in the correct condition. So the attempt to create a bias towards correct TPSF sentences in the experimental design might not have worked out completely. Therefore we would like to conduct a follow-up study in which we provide only correct sentences, again both via only visual presentation versus auditory and visual presentation. That would exclude possible noise in the experimental setup and provide solid evidence on the capabilities of speech to free resources.

The experimental condition, in which the TPSF was either preceded by speech or not preceded by speech, influences the writer's strategies during error analysis. When isolating the incorrect sentences, we notice that writers more often delay error correction in the speech condition, and start writing sooner than when no speech is presented. When speech proceeds the TPSF, writers can overtly compare the TPSF when it appears on screen with the speech. However, in the non-speech condition, as when writing by keyboard and mouse, only an internal, covert conflict is possible. With speech, some of the time, the speech confirms that the TPSF is the text intended and sometimes it does not. Without speech, this kind of explicit confirmation is not possible. The present results show that writers adjust to this uncertainty in the TPSF by correcting errors immediately and by slowing down starting time. In other words, the auditory channel does cause a significant focus on fluent text production.

Compared to the mode of error presentation, error span has a more consistent effect on strategy choice. Writing without speech, and with large errors, leads to the highest cognitive effort in error analysis. Large errors lead to slower preparation time, longer production times, and slower interference reaction times, indicating that they consume more working memory resources. A positive effect of speech can definitely be found in the comparison between large and small errors. In general large errors distract more than small errors, but not when the TPSF is also offered via the auditory channel. Writers also need less preparation time when speech is present. It even does not make a difference if the error is large or small. If we describe fluency as a measure to continue text production, then the fluency is significantly higher in the speech than in the non-speech condition: writers more often prefer to continue text production in a fluent way when the TPSF is dictated first.

One benefit of the present research is that writing mode can be separated from errors that are caused by a particular mode of writing. That is, writing with speech recognition generates a type of error that is distinct from those found when writing with keyboard & mouse, and the error types can be experimentally separated from mode of writing in the laboratory. In this experiment we have made this operational on two levels: the mode of writing and the lexicality of words. The results for both are to a large extent in line with each other. Errors inherent to writing with speech recognition and with keyboard & mouse are equally distracting. But the strategy shown to be associated with lower rates of error correction success, delay of correction, is

more likely when speech recognition errors and keyboard errors that result in existing words are involved. Keyboard based errors that result in non-existing words are solved better, both in the speech and non-speech condition.

The error types used in this experiment can be divided into three categories: large errors, small errors that result in existing words and small errors that result in non-existing words. Large errors distract writers in a way that it takes them longer to continue with text production or correct the error. So, large errors are cognitively demanding, but are accurately solved, and small keyboard errors that result in non-words do not cost more cognitive effort, but also result in a higher accuracy.

In general, we expected it to be easier to compare the mental representation of the TPSF with only the visual feedback on the screen, than to compare it with visual and auditory feedback. In short, to compare two things is easier than comparing three things. The results provide evidence that writers conduct three tasks differently than two tasks. It seems that writers opt not to conduct three activities. When speech is present the preparation time is shorter and writers generally prefer text completion, which suggests that they use the auditory information merely to continue text production. A similar experiment with eyetracking can confirm this assumption. Writers can either continue text production based on the auditory information or they can use the visual information as a trigger to continue text production. Error correction is not a priority in this situation; the TPSF is just a vague visual stimulus to continue text production.

Writers can be divided in various correction profiles. Some participants prefer to correct almost all the errors before completing the text; others do just the opposite, or change their strategy. The multilevel analysis provided an adequate basis to take this preference into account on the variables. The quadripartite that we have opted for in this study is probably not feasible in more natural writing processes. The fourth group delayed almost every error correction, after they had completed the sentence first. The study in chapter 7 shows a tripartite in correction behavior, in which the latter group of delaying almost 100% to a later writing phase is not that extreme, but shows a quite comparable behavior. The preference to correct errors immediately or to delay error correction is not of influence on the reaction time. An explanation might be that writers opt for a correction strategy that is most related to their working memory capacity. If writers expect delaying error correction might cause an extra burden then they opt to correct the error first, and vice versa. When writers are asked about the rationale behind the strategy of delaying the correction of an error, they state that the subprocess of production is sometimes more important and that text completion is performed before error correction because they are afraid to lose the 'gist' of their formulation (see chapter 7). In this experimental setup we see a higher percentage of delaying error correction than in a context in which writers need to write a complete text. In that situation, writers need to balance even more between continuing text production and re-reading the TPSF. However, in this experimental setup one can

isolate the effects of various error types, while in observational research the cognitive effort of a writing task varies. Above all, the cognitive planning process related to the content development was almost removed in this experiment. In a follow up study, it may be helpful to integrate a more complex planning component by providing the context not as a full sentence, but only as keywords.

Multilevel analysis enabled us to analyze the data from a hierarchical perspective. The general results are quite equal to the description on an aggregated level. However, analyzing the data with multilevel models results in statistically more valid conclusions, because of the larger statistical power (Hox, 2002). In other words, we are more confident that the effects we found are not due to sampling errors. Also, the chance on aggregation bias in the conclusions drawn from the analyses is much smaller. A third advantage was the flexibility to control for the effect of participant (and sentence) characteristics in our models. This increases the internal validity of our conclusions. Therefore, we are more certain that the effects that we found are not due to variables not taken into account, for example participant characteristics. As a result we can be more confident in our conclusions based on the multilevel analyses.

To answer our hypothesis we mainly used the (single) description of the fixed effects. However, the variances at participant and sentence level provide an additive explanation to our data. The results related to large and small errors in the TPSF is a good example of the added value the information in the random part of the model provides us. Therefore, we have reported the variance between sentences for large speech recognition errors compared to small speech recognition errors in hypothesis 3a. Reaction time, preparation time and production time showed a significant difference on the sentence level, both within and between error types. The results of the variances also indicate that in follow-up experiments we should perhaps be more careful in distinguishing the category of large errors. Therefore, in a follow-up experiment we have opted to control the error types even more detailed (e.g. by taking care of word frequency, word length, and by inserting errors only in unstressed syllables; (Quinlan et al., in preparation).

In this experiment we described the cognitive processes of writers via the variables' reaction time, preparation time, production time, delayed error correction and accuracy. A well-known measure of cognitive load during writing processes is reaction time. In using this measure it is important to choose the exact moment the secondary task is required. Since the sentences that needed to be read were rather short, the variation in offering the secondary task was limited to a small span. As not to bias the writers, we varied the timing of the second task in the correct filler sentences on a broader scale. However, the reaction time was not as decisive as we assumed. Only in the most extreme writing situation - large speech recognition errors - did it provide more insight in the cognitive effort. In a more neutral writing situation this measure was inconclusive. The supplementary measure of preparation time seemed to be more informative for this experimental setup. The time participants needed to decide

what their next writing action would be differed as well for the mode of presentation and for the comparison between large and small errors. Since the task included the instruction 'speed on task' we see this measure as highly informative for the cognitive effort it takes to continue the writing process (whether this is production or correction). The final time measure is production time. Shorter production time is related to easier writing processes. Writers that are more fluent in text production produce longer text in the same amount of time as less fluent writers. Therefore, production time can also be taken as a measure for cognitive effort: the cognitive effort it takes to produce a text as fast and accurately as possible. The cursor position that writers choose can be described as a strategy choice to continue text production or to revise first. Again this can be seen as a cognitive effort measure too. If speech is present, writers prefer to continue text production. In a follow up study in which writers were forced to correct errors first, the production time was significantly lower (Quinlan et al., in preparation). In this study, reaction time as 'the measurement' of cognitive effort *an sich*, would not have provided as much information as the combination of the time, strategy and accuracy measures together. In our opinion the combinations of various measurements are needed to accurately describe the cognitive processes in writing.

Acknowledgements

We especially would like to thank Sven de Maeyer for his patience in explaining the principles of multilevel analyses. We appreciate his critical thinking along with our research questions. In addition, we would like to thank Isabelle De Ridder for co-designing and coordinating the experimental sessions, and data collecting. Bart Van de Velde did a great job in programming the experiment. We also would like to thank Michael Levy for adapting the Counting Span Test for Visual Basic.Net. Finally, we would like to thank Tom Van Hout for proofreading this chapter.

The project was funded as a BOF/NRI (New Research Initiatives) project by the University of Antwerp (2002-2004). The software of the experiment and the logging tool Inputlog of the experiment is available from the internet respectively at the following urls: www.ua.ac.be/marielle.leijten and www.inputlog.net.

Appendix

Table 11. Parameter estimates for reaction time per error type (speech, non-speech)

	Reaction time	
	<i>Est.</i>	<i>SE</i>
Fixed part		
SR Large	493.554	80.321
SR Small	389.139	76.415
SR Keyboard Small	397.775	79.044
Keyboard Small	417.991	77.492
Speech * SR Large	-99.047	39.017
Speech * SR Small	-44.267	20.393
Speech * SR Keyboard Small	57.027	35.017
Speech * Keyboard Small	-5.888	25.378
Median Baseline Reaction Time	0.837	0.144
Random Part		
Participant level		
SR Large	23941.970	8743.828
SR Small	22005.180	5181.527
SR Keyboard Small	21731.600	7475.973
Keyboard Small	24265.960	6243.356
Sentence level		
SR Large	134578.100	11098.730
SR Small	37427.320	3055.926
SR Keyboard Small	108928.100	8953.193
Keyboard Small	57356.120	4706.578

Table 12. Parameter estimates for preparation time per error type (speech, non-speech)

	Preparation time	
	<i>Est.</i>	<i>SE</i>
Fixed part		
SR Large	4008.858	211.156
SR Small	2712.774	106.737
SR Keyboard Small	2740.566	101.480
Keyboard Small	2636.010	102.991
Speech * SR Large	-2203.180	284.612
Speech * SR Small	-1100.811	120.222
Speech * SR Keyboard Small	-1003.516	105.776
Speech * Keyboard Small	-855.564	101.255
Group immediate medium	-526.838	98.948
Group delayed medium	-1151.583	102.350
Group delayed	-984.916	102.329
Random Part		
Participant level		
SR Large	0.000	0.000
SR Small	35835.440	49393.580
SR Keyboard Small	66859.390	44953.570
Keyboard Small	113096.400	50505.230

Sentence level		
SR Large	7167918.000	538502.100
SR Small	1297077.000	106078.700
SR Keyboard Small	1006976.000	82219.230
Keyboard Small	919969.200	75243.560

Table 13. Parameter estimates for production time per error type (speech, non-speech)

	Production time	
	<i>Est.</i>	<i>SE</i>
Fixed part		
SR Large	27895.020	976.372
SR Small	20172.100	685.672
SR Keyboard Small	20190.780	614.798
Keyboard Small	19967.130	651.265
Speech * SR Large	-2482.195	888.069
Speech * SR Small	-383.278	549.530
Speech * SR Keyboard Small	-1071.854	469.692
Speech * Keyboard Small	-829.839	462.315
Sex	-2549.793	562.819
Random Part		
Participant level		
SR Large	27167610.000	7183961.000
SR Small	13772000.000	3361718.000
SR Keyboard Small	10654100.000	2563839.000
Keyboard Small	13596650.000	3078867.000
Sentence level		
SR Large	69411880.000	5734578.000
SR Small	27092000.000	2215746.000
SR Keyboard Small	19854980.000	1621153.000
Keyboard Small	19174320.000	1568187.000

Table 14. Parameter estimates for delayed error correction per error type (speech, non-speech)

	Delayed error correction	
	<i>Est.</i>	<i>SE</i>
Fixed part		
SR Large	0.685	0.217
SR Small	0.983	0.231
SR Keyboard Small	0.644	0.216
Keyboard Small	0.392	0.213
Speech * SR Large	2.374	0.395
Speech * SR Small	1.738	0.353
Speech * SR Keyboard Small	2.088	0.348
Speech * Keyboard Small	2.161	0.326

Random Part		
Participant level		
SR Large	1.281	0.469
SR Small	1.531	0.518
SR Keyboard Small	1.321	0.456
Keyboard Small	1.340	0.443

Table 15. Parameter estimates for accuracy per error type (speech, non-speech)

	Accuracy	
	<i>Est.</i>	<i>SE</i>
Fixed part		
SR Large	2.369	0.364
SR Small	0.916	0.212
SR Keyboard Small	1.232	0.221
Keyboard Small	2.076	0.276
Speech * SR Large	-0.393	0.407
Speech * SR Small	0.120	0.245
Speech * SR Keyboard Small	-0.067	0.260
Speech * Keyboard Small	0.072	0.357
Group immediate medium	-0.159	0.199
Group delayed medium	0.264	0.219
Group delayed	0.511	0.229
Random Part		
Participant level		
SR Large	1.243	0.662
SR Small	0.065	0.175
SR Keyboard Small	0.000	0.000
Keyboard Small	0.000	0.000

Table 16. Parameter estimates for correct and incorrect sentences for delayed error correction in the speech and non-speech condition per group (immediate to delayed) [data belong to Figure 5]

	Delayed error correction	
	<i>Est.</i>	<i>SE</i>
Fixed part		
Speech - correct sentences	2.139	0.452
Speech - incorrect sentence	1.216	0.348
Non-speech - correct sentences	1.674	0.377
Non-speech - incorrect sentences	-1.155	0.201
Group immediate medium	1.624	0.256
Group delayed medium	3.247	0.321
Group delayed	5.735	0.724
Random Part		
Participant level		
Speech - correct sentences	4.952	1.676
Speech - incorrect sentence	3.086	0.972
Non-speech - correct sentences	3.385	1.145
Non-speech - incorrect sentences	0.343	0.173

Section III

Error correction strategies of
professional speech recognition users
writing business texts



7

The text produced so far in business texts

Error correction strategies of professional speech recognition users

Abstract: One of the challenges in writing research in general is to explain the structural variation in writing processes within and between subjects. More or less recursivity has been attributed to writing experience, proficiency, task characteristics and the writing mode or medium. In this study, we will focus on writers who use speech recognition as their primary tool for text production. Furthermore, we will concentrate on one significant subprocess, namely revision (repair), more specifically: error correction as one of the key factors in determining the structural characteristics of writing processes. We observed 10 professional writers who are experienced speech recognition users. The writers are observed while writing a business report. We provide 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.

In this study several converging research methods are used: (1) product analysis (the final product - a report - is analyzed on various levels), (2) process analysis (the data of the writing process are logged by the logging program Inputlog, the speech recognizer Dragon Naturally Speaking 8.1, and the usability program Morae), and (3) protocol analysis (the participants were asked questions during a stimulated retrospective interview).

The results are described on three levels: overall level, subgroup level (three writer groups and individual level (case study)). The contrast between immediate and delayed error correction is quite decisive for

This chapter is based on an article in preparation Leijten, M., Janssen, D., & Van Waes, L., (in preparation). The text produced so far in business texts: error correction strategies of professional speech recognition users.

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 revisions. However, the strategy to postpone errors is not equivalent to postponing revisions. The overall results show three distinct patterns of error correction. First, there are writers who prefer writing a first time final draft and solve technical problems immediately as well as revising the text produced so far immediately (*handle profile*). Second, writers who solve more than half of the deficiencies in the text produced so far immediately, but who also delay or postpone various technical problems and revisions (*postpone revisions profile*). Finally, writers who prefer delaying error correction and who delay technical problems to a 2nd draft (*postpone technical problems profile*). These error correction strategies are illustrated in greater detail in a case study.

Keywords: Cognitive processes, error correction, Inputlog, keystroke logging, online writing processes, pauses, pause analysis, research methods, speech recognition, text analysis, writing modes, writing observation.

1 Introduction

One of the challenges in writing research in general is to explain the structural variation in writing processes within and between subjects. Some writers seem to prefer a more or less linear path; they plan first, then formulate and finally revise with little or no recursion (Schilperoord, 1996; Selzer, 1983, 1984). Some writers go through a number of cycles of planning, translating and revising (Flower & Hayes, 1981; Flower, Hayes, Carey, Schriver et al., 1986; Hayes, Flower, Schriver, Statman et al., 1987). Others go through a more or less chaotic process with some recursion and some linearity in which it is hard to distinguish a pattern at all. But even within subjects we see differences in the way they organize their writing processes. Within the single writing process of a single writer we may find phases in which the writer works more linearly or recursively (Van den Bergh & Rijlaarsdam, 1996). Or the same writer may approach specific writing tasks differently. Simple or well-rehearsed tasks, for instance, may be a matter of mere 'knowledge telling' and thus be executed in a more or less linear fashion; other tasks may require 'knowledge transforming', which often leads to more recursion (Bereiter & Scardamalia, 1987).

Many researchers have come up with explanations for the observed differences. More or less recursion has been attributed to writing experience, monitor-configuration, proficiency, task characteristics and – of course – the writing mode or medium. In a previous study on the influence of speech recognition on the writing process (Leijten & Van Waes, 2005b, also chapters 2, 3 & 4) we have explored how novice speech recognition users adapted to the new writing medium.

We observed professional writers in their own working environment while writing 'to-be-written' business text.¹ We opted for an ecologically valid method: non-intrusive ethnographic observation in the daily writing environment. Therefore, writing tasks varied between and within writers. In this present study we opted for a more controlled approach. We observed 10 professional writers who are experienced speech recognition users. The writers are observed while writing a business text (a given task). Speech recognition is again the main writing mode. The focus of the study is on error correction strategies. We provide 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.

In this study, we focus on writers who use speech recognition as their primary tool for text production. Furthermore, we concentrate on one significant subprocess, namely revision (repair)². More specifically; error correction as one of the key factors in determining the structural characteristics of writing processes. If we compare the use of speech recognition to classical dictating it is evident that efficient and effective dictating requires linearity. The absence of the visual text produced so far (and thus the necessity to playback the tape and insert new text) makes intermediate revision cumbersome. Therefore, dictators plan their text ahead or hardly need planning because they use mental 'templates' and stock phrases and paragraphs (Schilperoord, 1996).

Speech recognition, on the other hand, induces recursion. Writers who use speech recognition simply talk to the computer. Software on the computer enables normal keyboard and mouse operations to be voice-activated. The software converts the spoken signal to visual characters on the computer screen. This mode of text production differs from classical dictation in a significant way. Firstly, the speech recognition software transforms the verbal input into visual text on screen. As a result the writer is confronted with characteristic errors that need to be corrected. The software can only translate the input into existing words from its lexicon. Normal misspellings and typographical errors cannot occur in speech recognition. Therefore, a sentence like 'this is a test' may well be recognized as 'it is a test' (and not as for instance 'tjhs is a tst', which is quite common in normal keyboard writing). Knowing that these errors can occur, puts an extra constraint on the writer. Because of the grammatical correctness and its orthographic resemblance the error 'it is a test' is easy to miss. On the other hand, speech recognition can lead to text that shows no semantic resemblance to the writer's intention: a word like 'I' may become 'eye' or a sentence like 'the case is interesting' may show up as 'the gaze is the rest thing'. Both the small, but easily to miss errors and the total misrepresentation put a burden on the cognitive pliancy of

¹ At this moment, February 2007, Dragon Naturally Speaking has reached version 9 – claiming an accuracy rate of 99% – and is available in seven languages. Dutch is one of those. In addition to the general packages, their main focus is on the medical and legal market, since these professions were already used to (analogues) transcription and dictation devices. Also, real-time subtitling of television broadcasting is an important domain for speech recognition.

² In this study, we elaborate on the notion of revisions. Because of the typical speech recognition based errors, we make a distinction between technical problems and revisions. The definition that we use to indicate the combination of both is repairs (cf. section 3.3.1).

writers (cf. chapters 5 & 6). Writers must not only plan and monitor their speech but they must also constantly evaluate whether the text-produced-so-far matches their mental representation. And, since recognition errors occur all the time, writers are compelled to monitor the text continuously. Moreover, since some errors are so easy to miss, writers may well feel the need to revise their texts immediately, which makes their writing processes highly recursive.

#	Time (minutes)	WritingMode	Output	StartClock	StartTime (ms)	EndClock	EndTime (ms)	ActionTime (ms)	Pause (ms)	X	Y
1	4,20	1	ENTER	0:17:28	1048984	0:17:29	1049078	94	203		
2	4,36	3 - dict	binnen de afdeling	0:17:44	1064852	0:17:46	1066558	1706	15868		
3	4,40	3 - dict	ligt de werkelijke bijzonder hoog .\punt	0:17:49	1069242	0:17:51	1071945	2703	2684		
4	4,49	3 - dict	look de	0:17:57	1077457	0:17:58	1078674	1217	5512		
5	4,50	3 - dict	motivering bij de medewerkers	0:17:58	1078604	0:18:00	1080878	2274	0		
6	4,52	3 - dict	wachten tot voor kort erg hoog	0:18:01	1081036	0:18:03	1083769	2733	158		
7	4,55	3 - dict	.\punt	0:18:03	1083897	0:18:05	1085004	1107	128		
8	4,58	1	HOME immediate error correction	0:18:07	1087031	0:18:07	1087125	94	2027		
9	4,59	1	DEL (w)	0:18:08	1088171	0:18:08	1088250	79	1140		
10	4,59	1	DEL (a)	0:18:08	1088343	0:18:08	1088406	63	172		
11	4,60	1	DEL (c)	0:18:08	1088468	0:18:08	1088531	63	125		
12	4,60	1	DEL (h)	0:18:08	1088609	0:18:08	1088671	62	141		
13	4,60	1	DEL (t)	0:18:08	1088734	0:18:08	1088796	62	125		
14	4,60	1	DEL (e)	0:18:08	1088843	0:18:9	1089046	203	109		
15	5,00	1	DEL (n)	0:18:9	1089140	0:18:9	1089265	125	297		
16	5,01	1	l	0:18:10	1090234	0:18:10	1090312	78	1094		
17	5,01	1	a	0:18:10	1090328	0:18:10	1090453	125	94		
18	5,02	1	g	0:18:10	1090593	0:18:10	1090656	63	265		
19	5,03	1	END	0:18:12	1092171	0:18:12	1092265	94	1578		
20	5,04	1	SPACE	0:18:12	1092812	0:18:12	1092968	156	641		
21	5,08	3 - dict	maar de	0:18:16	1096609	0:18:18	1098235	1626	3797		
22	5,11	3 - dict	toegenomen werklast	0:18:19	1099656	0:18:21	1101401	1745	1421		
23	5,14	3 - dict	stelt die	0:18:22	1102467	0:18:23	1103674	1207	1066		
24	5,17	3 - dict	motivering danig onder druk .\punt	0:18:25	1105459	0:18:28	1108262	2803	1785		
25	24,13	2	Left Button	0:37:21	2241406	0:37:21	2241468	62	286	389	367
26	24,14	1	LEFT delayed error correction	0:37:22	2242875	0:37:23	2243015	140	1469		
27	24,14	1	DEL (e)	0:37:23	2243187	0:37:23	2243296	109	312		
28	24,15	1	DEL (k)	0:37:23	2243375	0:37:23	2243484	109	188		
29	24,15	1	DEL (i)	0:37:23	2243593	0:37:23	2243671	78	218		
30	24,15	1	DEL (j)	0:37:23	2243765	0:37:23	2243875	110	172		
31	24,15	1	DEL (k)	0:37:23	2243984	0:37:24	2244109	125	219		
32	24,16	1	DEL (e)	0:37:24	2244437	0:37:24	2244515	78	453		
33	24,17	1	d	0:37:25	2245484	0:37:25	2245609	125	1047		
34	24,17	1	r	0:37:25	2245734	0:37:25	2245781	47	250		
35	24,17	1	u	0:37:26	2246031	0:37:26	2246093	62	297		
36	24,18	1	k	0:37:26	2246609	0:37:26	2246687	78	578		
37	24,19	2	Movement	0:37:28	2248171	0:37:39	2259875	11704	1562	553	383
38	24,31	2	Left Button	0:37:40	2260046	0:37:40	2260156	110	11675	553	383

Figure 1. Example of immediate and delayed error correction.

However, observations of writers using speech recognition have shown that there are still huge differences in the way in which writers deal with revision (Leijten & Van Waes, 2005, also chapters 2, 3 & 4). Figure 1 shows two excerpts from one of our protocols.

(02) within the department
 (03) the real [pressure] of work is very high
 (04) also the
 (05) motivation of the employees
 (06) waiting [was] until recently very high
 (07) .\full stop
 (21) However, the
 (22) increase in pressure of work
 (23) puts the
 (24) motivation under pressure .\full stop

The English translation of the text is (the underlined parts are incorrectly represented on the screen: we give a literal translation of the word that appeared in Dutch – the translation is given between square brackets).

In the first part of the writing session (4'20"-5'17") we see a writing process in which the writer is confronted with two errors in the text produced so far (#3: 'pressure of work' and #6: 'was', are both represented incorrectly). Apparently the writer detects the second error, immediately stops, makes a repair via keyboard and mouse and continues producing text via speech (# 8-18). In the second part of the same session (24'13"-24'31"), we see a different pattern: the repair is postponed. The writer writes various paragraphs (for about 20 minutes), and then opts to return to the beginning of the text to start with a first error correction. He then continues to revise the content of the text and to correct formal errors in a repair episode for a while, after which he continues text production. We call the first pattern 'immediate' error correction and the second type 'delayed' error correction.

Both patterns often occur in speech recognition. It is evident that both instances make the writing process recursive. In both cases error correction interrupts the text production. Instead of finishing a complete version of the text and revising/repairing the whole text at once, the writer goes through cycles of planning, formulating and revising. The difference lies in the time interval: immediate repair or delayed. Immediate repairs obviously lead to a higher recursive pattern than delayed repairs (because delayed repairs are often carried out in episodes). In this contribution we would like to explore the rationale behind this: why are some errors repaired immediately and others later on in the process.

1.1 Reflection on the text produced so far in speech recognition

As mentioned in the introduction, error correction in the text produced so far (TPSF) in speech recognition may interrupt the linearity of the writing process in a different way than it does in for instance traditional computer based word processing. Focusing on error correcting is interesting for several reasons. On the one hand the correction of errors reveals an explicit awareness of the writing modes. This is shown by the difference in using speech input during error correction, and the related switches to keyboard & mouse. On the other hand, the focus on error correction enables us to analyze the writer's mental interaction with the TPSF in a particular way. Writers who need '(re)reading to write' (writers who use the TPSF to prompt new ideas and guide planning) could experience difficulties making a correct representation due to deficiencies in the text on the screen. In other words, speech recognition can be seen as a perspective to gain better insight in the cognitive processes that are related to writing in general. Moreover, speech recognition is particularly interesting because of its flaws and its imperfectness. It brings certain processes to the surface. In other words, we deliberately created a task with speech recognition to create a high standard for cognitive effort. We try to provide more insight in error correction strategies as writers

are pushed to the edge by the rather difficult task and the inherent difficulties related to the writing mode. The between-writer variation in this situation is therefore very interesting to gain insight in how writers juggle constraints while writing. Of course, we should be careful in generalizing across tasks and writing modes.

In previous studies we have described speech recognition as a hybrid writing mode since it combines characteristics of both classical dictating and computer writing (Leijten, Ransdell, & Van Waes, submitted; Leijten & Van Waes, 2005b). Next to the input of the writing medium, the representation of the TPSF is one of the distinguishing characteristics. To explain this, we will elaborate on the similarities and differences of the TPSF in classical dictating, keyboard based and speech recognition based word processing. Like in computer writing, the TPSF in the speech recognition mode is visible on the screen; similar to classical dictating, it is also audible via text-to-speech. Like in computer writing, the emerging text appears almost immediately on the screen, not letter by letter, but in text segments or phrases. After training, the text on screen is a more or less correct representation of what has been dictated. This is the main difference with classical dictating, where the writer - in writing a first draft - has no visual representation of the text at all.

The correctness of the representation differs too; it can be placed on a continuum from correct to incorrect. We consider the representation of the classical dictating device to be correct, because the audio on the tape is identical to the input; of course, the transition to the paper is dependent on the typist. In computer writing, we classify the representation on the screen as being (semi) correct, because of the typing errors that might occur. However, the misrecognitions of the speech recognizer - and consequently the incorrect representations on the screen - are of a different kind. There is a possible overt conflict between the TPSF and speech in speech recognition and a possible covert conflict in computer writing.

So, writers using speech technology get immediate written feedback on the computer screen that may overtly conflict with the TPSF they dictated. Consequently, speech recognition causes a different dependency for the writer on the TPSF. Writers not only need to keep a mental model of the text, they can also be easily distracted by an incorrect visual representation on the screen. They need to balance between a mental and a visual representation, since the output of the speech recognizer is still less predictable than when writing with keyboard & mouse. This makes speech recognition a suitable writing instrument to bring the role of the TPSF into focus.

The predictability of the TPSF is inherent to the input of the writing medium. Experienced typists that use keyboard based word processing are physically closely connected to the text they are making. Writers 'feel' most of the typing errors that they make. In speech based word processing this physical relation is less clear. An experienced speech recognition user will be able to predict some of the errors that appear on the screen, for example, if the writer stutters or hesitates. But not all of the errors are equally predictable. So the physical connectedness to the visual representation differs in both writing modes.

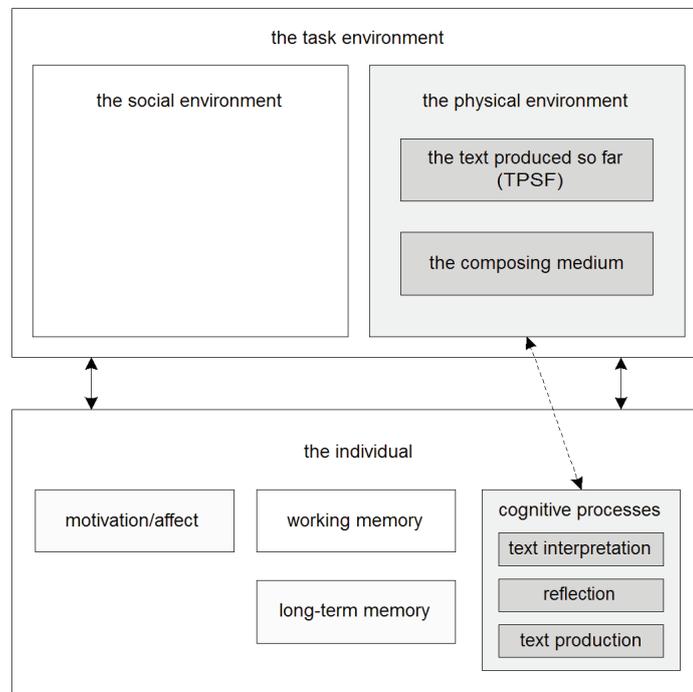


Figure 2. Synthesis of the writing model of Hayes (Hayes, 1996, p. 4).

As the Hayes & Flower writing model in Figure 2 shows, the text produced so far is an important component of the task environment in monitoring the writing process. The composing medium is largely responsible for the amount of discrepancies in the TPSF. This might lead to strategies that either encourage a continuation of text generation or a delay in correcting errors.

The text interpretation and reflection on the 'text produced so far' is a very complex cognitive task, both in the classical dictating mode and in the computer mode. The original process model of revision proposed by Flower et al. (1981), which has been derived from thinking-aloud protocols, globally consists of three main processes (Flower & Hayes, 1985; Flower et al., 1986; Hayes et al., 1987):

1. Text evaluation: writers reread the TPSF for comprehension, evaluation and problem definition. At this stage writers need to detect and diagnose problems in the text.
2. Strategy selection: (a) writers can opt to ignore a problem, (b) decide to solve the problem at a later stage, (c) search for more information before making changes, (d) rewrite the text while preserving the gist of a (text) segment or finally (e) revise the text while preserving the already produced text.
3. Revision execution: writers can opt to make fundamental meaning changes to the

text (that alter the plan) or they can opt for a more basic change that only alters the formulation (cf. *infra*).

We assume that in speech recognition the same revision subprocesses will play a role. However, as stated before, in text evaluation another subprocess occurs. Since revising is a very complex process in either writing mode, speech recognition enables us to focus in more detail on the complexity of revising/repairing. For instance, typing errors in keyboard based word processing leads to less explicit revising processes. The demands placed on the working memory are rather low for typing errors, because writers know how to solve them and because they often 'feel' the typing errors made. Furthermore, they may know that the spelling checker will take care of most of them. However, the typical errors that are generated by speech recognition do lead to visible processes. The demands on the working memory here are much larger during the repairing processes, because of the obvious discrepancy between the mental representation and the physical manifestations of the TPSF. A previous case study showed that novice speech recognition users needed to interrupt the writing process to solve errors in the TPSF. Apparently, certain error types were so distracting, that they undo the main benefit of speech recognition (*viz.* high speed text production). A second study in which we isolated - among others - the effect of error span on error correction strategies (Leijten, Ransdell, & Van Waes, submitted, also chapters 5 & 6), showed that large errors resulted in an increase of cognitive effort while writing.

In Figure 3 we have tried to model the writing process in general and more specifically the interaction between the physical and the mental TPSF and other cognitive processes. The upper two levels (1-2) represent basic but fundamental knowledge needed for writing: the situation model and the discourse model. The situation model consists of world knowledge and information about the writer's rhetorical situation (goals, intentions, audience, etc.). The term situation model comes from reading research where it is used to describe and explain how readers use and more specifically integrate information from texts. A situation model of a football game for instance

calls for a temporal sequence of events at various locations, for causal relations between the events, and for the interaction of individuals, interacting physically and socially, governed by physical laws and constrained by the 'laws' of the game and social conventions and motivated by various intentions. (Johnson-Laird, 1983, p. 414)

A writer needs to translate the information from the situation model into a discourse model. Discourse models contain information about content and structural characteristics of genres, paragraphs, sentences, tone, etc. Sometimes that transformation is easy, for instance, when a lawyer needs to write a simple standard letter to a client as Schilperoord's (1996) subjects did all the time. In those instances the situation model is 'fixed' and so is the discourse model. In other instances the situation model is easily translatable into a discourse model, namely in all instances of knowledge telling. But if the writing tasks are new or complex, translating the situation model into a discourse model may be a challenge. This is usually the case in all types of knowledge

transforming writing (Bereiter & Scardamalia, 1987). The third level (3) consists of a mental representation of the texts produced so far; a mental image of the paragraphs, sentences and words that the writer has produced. Levels 1-3 represent the mental, cognitive domain. The level 3 representation can differ from the actual text at level 4 the physical representation. This is for instance the case when writers overlook errors. Writing in this model is basically a matter of translating information to a more concrete level and vice versa constantly mapping the different levels. Of course, information becomes more concrete and elaborate from top to bottom. One element from the situation model may lead to several elements in the discourse model, which in turn leads to more elements in the TPSF.

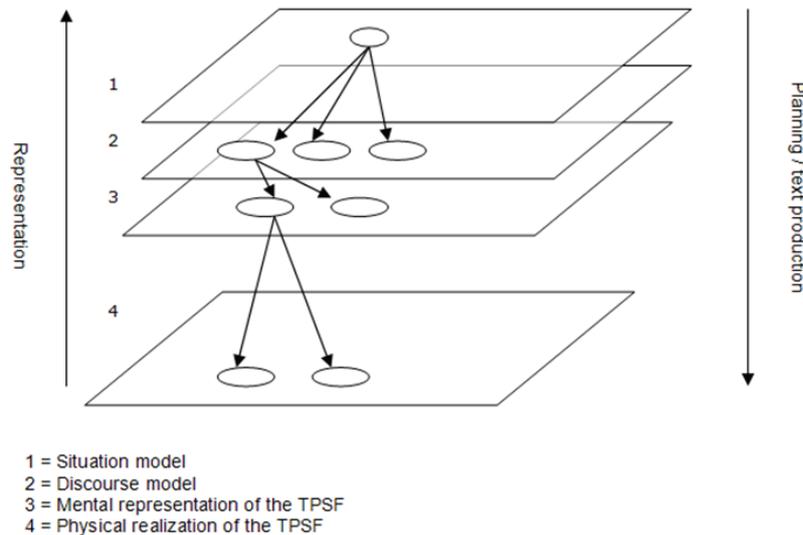


Figure 3. Interaction between physical and mental representation of text produced so far.

Levels 1 and 2 are basically the same in all writing modes: keyboard, dictating and speech recognition. In dictation level 4 is non-existent. That makes dictating less constrained than writing with speech recognition. The dictator does not need cognitive energy for mapping levels 3 with 4. Repairing in speech recognition on the other hand seems to be mentally more demanding than keyboard writing. Due to the software the writer must pay extra attention to levels 3 and 4. As we have seen earlier the software generates errors that a typist will never make and that are undetectable by spelling checkers. It may well be that speech recognition writers heavily monitor their writing on this level and revise more instantly to free mental space for planning and evaluation at the higher level.

Writers who postpone or delay error correction may not always be able to make

a correct representation of the information they want to convey. If so, they need to 'search' through the discourse or situation model for the necessary information. Writers may also choose to proceed more quickly with speech recognition than during computer writing because in principle the spoken input allows for a faster text generation than with keyboard & mouse. And although typing is highly automated for most writers, it will still absorb some cognitive energy that the speech recognition writer can use for other planning and revision purposes.

With this model we might be able to explain why speech recognition writers on the one hand decide to interrupt the linearity of the writing process to solve problems and on the other hand why they continue text production and ignore problems or delay their correction. As long as they think that they can make a correct representation of the TPSF and match that with the discourse model, they can continue writing. If the mismatch between levels 3 and 4 is too big, representation and mapping on this level requires so much cognitive energy that higher level planning and evaluation are disrupted. In those cases it becomes necessary to solve the local problems first and get them out of the cognitive system to free resources for other processes.

So, we state that production is easier and faster with speech recognition, but that revising/repairing in speech recognition differs from keyboard based word processing and classical dictating because the writer constantly has to answer the additional question 'Is the text produced so far correctly presented?' (level 4, Figure 3) In other words, is the text that the writer has dictated to the speech recognizer correctly represented in the text that appears on the screen? In Figure 4 we describe the process that follows this question in the revising process. This action model has been adapted from the model that we introduced in chapter 5. Because the former model mainly dealt with technical problems, we opted in the new model for a more explicit focus on the interaction between technical problems and revisions.

The text interpretation and reflection can lead to two questions about the text produced so far. The specific order of addressing these questions is not predetermined. Writers will have different strategies for reflecting on the text that they have produced. One question is: 'Is the representation of the text produced so far correct?' (= mapping on the levels 3 and 4). The other question is: 'Is the content of the text produced so far correct?' (= mapping on the levels 2 and 3 and 1 and 2). If the text produced so far is correct then writers can either continue with text production or they can monitor the text for the unanswered question. For example, if the representation of the text is correct, writers can subsequently monitor the content of the text. As stated before, we are aware that the complexity of the writing task is increased by the speech recognition writing mode. Due to the higher task demands and the higher cognitive load as a result of that, writers are forced to set priorities in how to continue their writing process. Although this may be questionable from the viewpoint of validity, it generates the kind of data that we are interested in.

If the *content* of the TPSF is incorrect, writers can choose to prioritize revision (Flower et al., 1986; Hayes, 1996; Hayes et al., 1987) and then reflect again or

continue with text production. When the *form* of the text is presented incorrectly, because of a technical misrecognition by the software or a typing error, two possibilities remain. The writer can either detect or not detect the error. If the writer does not detect the error, he can choose again between text production and revision of the content. If a technical problem is detected by the writer, then he can opt for ignoring the problem, solving the problem at a later stage or searching for more information. In this scenario the writer again has two options: text production or content revision.

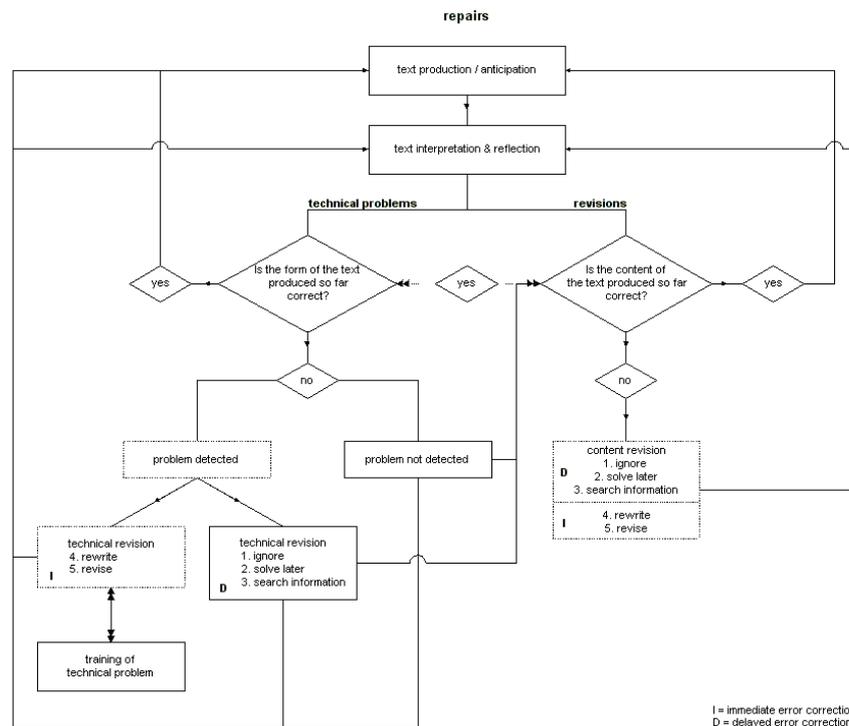


Figure 4. Theoretical action model of repairs.

Writers can also opt to correct the error immediately again by rewriting or revising the TPSF. In correcting technical errors, writers can opt to make fundamental meaning changes to the text (rewrite) or they can opt for a more basic change that only alters the initial formulation (revise). In other words, writers can choose to change the text while they correct a technical problem or they can just 'fix' the problem and stick to the original formulation.

Writers can either perform the technical revision immediately or they can train the speech recognizer to correctly recognize the word the next time they use it. The speech recognizer does not necessarily 'know' every word that writers use and, therefore, one can make the software smarter by adding words to the lexicon and by

practicing the pronunciation of words and word combinations. Ultimately, speech recognition could lead to a strategy of (minimal) error correction. The reason for this strategy might be that writers prefer freeing their working memory by producing new text that is already planned internally.

1.2 Error correction strategies in writing with speech recognition

The model in Figure 4 shows several moments of decision related to the writing medium. We assume that the moment when writers switch between writing subprocesses is one of the trigger moments to decide whether to switch between writing media or not. However, we already know that there is no one-on-one relation between switches and subprocesses.

In the previous study on the influence of speech recognition (chapters 2, 3 & 4) we did not find a causal relation between mode switches and solving technical and content problems. In 65% of the repairs the participants (selection of 10 participants with and without previous dictating experience - different learning styles) did not switch between writing modes. Writers switched about one time per minute from the speech recognition mode to keyboard & mouse. Repairs are apparently not the main trigger to switch between writing modes. Since the writers had to perform about 3 repairs per minute, they opt to switch only in one third of the repairs. This leaves two questions unanswered: 'which repairs trigger mode switches?' and 'why do writers prefer to switch from speech to another mode in those instances?'

The error types also leave some questions unresolved: Why does a writer who prefers a first time final draft leave certain errors in the text? Are those errors left uncorrected on purpose or are they simply overlooked? So, on the one hand the interaction with the text on the screen can lead to a highly recursive writing process in which every error is repaired almost immediately, but on the other hand it can also lead to a less recursive writing process in which repairs are made at the end of a section or a text and initially left unnoticed.

Another aspect refers to the correction profiles of the writers. As described in Figure 3, the participants in our case study (Leijten & Van Waes, 2005b, 2006b) differed in the way they preferred correcting errors during writing. This observation also holds for the participants in the follow-up experiment (Leijten, De Maeyer, Ransdell & Van Waes, in preparation; Leijten, Ransdell, & Van Waes, submitted, also chapter 5 & 6) in which a (multilevel) analysis is made of the working memory requirements that lead to strategy change during writing. In sum, some participants prefer correcting most errors before completing the text; others do just the opposite, or change their strategy at different stages in the writing process. Especially the behavior of the last two groups opens perspectives for further analysis. What is the rationale behind the strategy of delaying the correction of an error? Which cognitive aspects related to the working memory influence this behavior? Is the strategy dominated by an absolute preference, or is it related to the type of interaction with the text produced so far and/or the type of error encountered during that interaction? A more refined analysis, in

which the data are studied from the perspective of the strategy of the participants and complemented with more detailed information of the cognitive (sub)process might reveal complementary factors.

1.3 Interaction with the text produced so far

Generally, five steps are characteristic of the interaction process with the (deficient) TPSF aimed at the formulation of new text (based on Flower & Hayes, 1985; Flower et al., 1986; Hayes et al., 1987):

- (a) reading of the TPSF – (b) detection of the error – (c) diagnosis – (d) editing – (e) completion of the text.

Writers have several reasons to read the TPSF. Two important reasons for this study are: reading as input to continue with text generation and reading to compare the mental representation with the visual representation. In keyboard based word processing the distinction between both reasons might be 80% to generate new text and 20% to monitor the visual representation. However, we assume that in speech recognition based word processing this division might be more equally divided, for example 50-50%. On the one hand writers might prefer frequently monitoring the TPSF, since they do not need to monitor their fingers typing. On the other hand, writers can opt to take advantage of their free hands to hold various papers at hand and focus on notes and other written material that might be helpful in text generation.

Based on this information we can distinguish two diverse profiles:

1. Handle profile [ab-c d e]: Writers who do not delay the error correction follow this process linearly in the 'Handle profile'. Writers read the TPSF to evaluate and then choose a procedure to fix the detected text problem on a continuum from 'ill-defined' to 'well-defined' problems (Hayes & Flower, 1986). A small difference can be made in the initial text completion between writers who only detect the problem (they will prefer 'saying it again'/'rewriting' [abd(ce)]) and writers who also make a thorough diagnoses of the problem (they prefer 'saying it differently'/'revising' [abcde]).
2. Postpone profile [a (bc) e (ab) cd]: When following the 'Postpone profile' writers do not really 'read' the TPSF in order to comprehend, evaluate and define problems, but they are mainly focused on grasping the main gist of the text in order to produce new text. Detecting problems is of secondary importance. Only after completing (part of) the text do they decide to diagnose possible errors and correct them. The postpone profile can be gradually divided in three profiles: gestalt (a e (a) bcd), detection (ab e cd) and diagnosis (abc ed) (see also chapter 5). All three profiles postpone the revision process, but the extent to which writers detect and diagnose the formal and content errors in the TPSF differs.

So, both strategies consist of substrategies. One of the main goals of this study is to describe why writers choose to solve problems (both technical problems and revisions) in their text immediately or why they prefer postponing this process. Another

issue related to the strategy writers use, and which might shed a light on the cognitive load of problems in the TPSF, is the use of writing modes during the execution of repairs. What strategies do writers use to make modifications to the TPSF? To answer this question we will describe the repairs in detail based on our categorization model (Leijten & Van Waes, 2005b, also chapter 3).

2 Method

In this research project we have combined converging research methods. In order to be able to capture the richness and complexity of what is involved in written composition it is important to cross-check the consistency and completeness of different data (Greene & Higgins, 1994). Next to a general questionnaire, the observation methods that we used are:

1. Product analyses: the final product of the writing session, a report, is analyzed on various levels.
2. Process analyses: the data of the writing process are logged by the logging program Inputlog, the speech recognizer Dragon Naturally Speaking 8.1 and the usability program Morae. The logging of the writing process data did not interfere with the writing process. Next to this, we have observed the writer and the writing process concurrently from another room.
3. Protocol analyses: after the participants had finished their writing task they were interviewed in a retrospective thinking aloud protocol. On the basis of the findings of the concurrent observation, we have used active prompts during the stimulated retrospective interview.

The findings from the product, process and protocol analyses should corroborate each other. In this section we will describe the participants of the study (section 2.1), the task they conducted (section 2.2), the data collection of the various observation methods used (section 2.3) and the procedure of the experiment (2.4).

2.1 Participants

We conducted our study in 10 sessions, which we organized at the Flemish radio and television channel 'VRT'. In 2001 VRT started to incorporate speech recognition for simultaneous live subtitling. The main reason to use speech recognition for subtitling is the speed with which text can be generated. At the division subtitling about 14 employees frequently write with speech recognition. Being both expert speech recognition users and professional writers, they were very suitable candidates for this study. In this experiment 10 employees cooperated voluntarily. They were allowed to take part in this study during working hours. As a gesture of appreciation they received a book by a professor emeritus of the section Business Communication: R. Haest 'Taal in Stukjes' [Language in Columns].

7 Male and 3 female participants took part in the observations, ranging in age from 25 to 52 years old (mean age is 35.2). Most participants have a university degree

in translating/interpreting, communication or the arts. At the time of the studies, they had at least ten years of experience in computer writing. They use speech recognition for about 2 to 10 hours a week (work related). The episodes in which they normally write with speech recognition last maximum 20 to 30 minutes, because of the high cognitive load involved in re-speaking for simultaneous subtitling.

2.2 Task

The writers were provided with a realistic situation model of a semi-familiar theme. The participants knew how to perform this kind of task, but did not necessarily have to perform it frequently: they were asked to write a report about 'well-being at the workplace' of one and a half page (A4). We provided the writers with enough information about the topic and the situation (2½ pages). Thanks to this writers were comparable with respect to their prior knowledge about the topic. Each writer was also invited to relate the topic to his or her own work situation.

The writing task combined components that were intended to induce both knowledge telling and knowledge transforming processes (Bereiter & Scardamalia, 1987). The participants were asked to paraphrase information about legal issues related to well-being (knowledge-telling). On the other hand they had to formulate advice to members of the board (knowledge-transforming). The preferred structure of the final report, a part of the discourse model, was given to the writers.

2.3 Data collection

The data collection started a week before the actual writing experiment. The participants received - via their management - an introductory questionnaire. Next to this, they received general information on the writing task ('prior knowledge' information about well-being at the workplace). The participants were asked to read this before the writing session. The actual task description was given at the start of the experiment. In this section we will elaborate on the questionnaire, the process and the protocol data.

2.3.1 Questionnaire

A week prior to the experiment the participants filled out a short questionnaire. The questionnaire addressed the background of the participants. The participants were questioned about their qualifications, previous experience in professional writing, and experience with computers. We paid extra attention to their experience level in writing with speech recognition and their attitude towards this writing modus. Finally, we questioned them about their preferred writing styles when composing professional texts.

2.3.2 Process data: Inputlog, Dragon Naturally Speaking 8.1 and Morae

To register the process we opted for a combination of Inputlog, Dragon Naturally Speaking and Morae. Inputlog is a logging tool that enables researchers to record the

data of a writing session in MSWord, generate data files for statistical, text, pause and mode analyses, and integrates the input of dictation processes using speech recognition software (i.c. Dragon Naturally Speaking 8.1; see chapter 8 for a more detailed description of the program).

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	410		
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	11549		
1	BS	0:05:01	301594	0:05:01	301688	94	6543		
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		
Linear Logging File									
interval	Output								
0:00:00	[Movement] {16672} [Left Button] [Movement] [Left Button] NUM LOCK NUM ...								
0:03:40	NUM + [Movement] <het tekst> {3548} NUM LOCK NUM LOCK {2344} [Movement] [Left Button] [Movement] <context in> {3199} ENTER ENTER <de algemene verplichtingen van de werkgever met betrekking tot de bescherming van gezondheid en veiligheid van werknemers op de werkplaats> {11549} <\punt> {6543} BS · <kaderen in de ontwikkeling ervan een preventiebeleid \punt> {6973} <de werkgever is verplicht te								

Figure 5. Merged logging data of Inputlog and Dragon Naturally Speaking.

At the time of the study Inputlog 2.0 Beta was the only logging tool that enabled researchers to integrate the input of speech recognition into a detailed process logging file. In this experiment we have used the Dutch version of Dragon Naturally Speaking 8.1 Professional³. The especially customized logging add-on in the speech recognizer (in combination with a Python script) enabled us to integrate the dictated text with the data logged by Inputlog. Comparable to the general logging file generated by Inputlog, the logging file of Naturally Speaking relies on timestamps. Via a Python script that was developed by Nuance (Boston) for this purpose, we generated an XML-file of a recorded speech session that is basically structured in the same way as the basic file of Inputlog. Consequently, both logging files can be combined and integrated into one general logging file which is based on the convergence of timestamps (see Figure 5). The result is a single file that can be used for further analysis of multi modal writing sessions in which speech input is combined with keyboard & mouse.

³ More information about Dragon Naturally Speaking products can be found on the Nuance website <http://www.nuance.com/>

Inputlog is 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)⁴. 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. Just like Inputlog it also logs very detailed timestamps which enabled us again to integrate the additional data registered by Morae into the output of Inputlog.

Next to this, Morae integrates a webcam that allows us, for instance, to also view the writers' actions in combination with the text developing on the screen. In this experiment we are mainly interested in the way writers interact with the TPSF on the computer screen. Figure 6 illustrates the observation setting of the experiment.

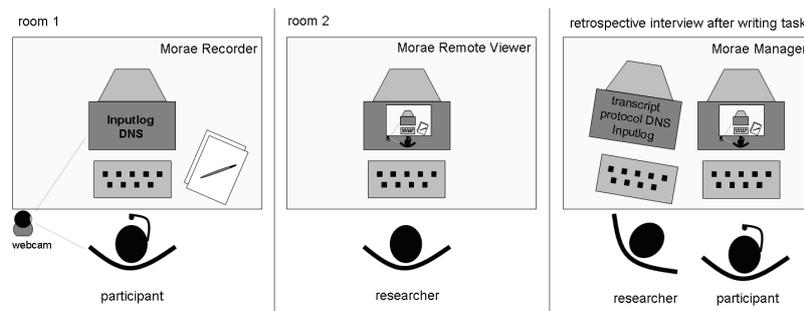


Figure 6. Observation setting of experiment.

As Figure 6 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 the retrospective interview (cf. protocol analysis). An illustration of this view (in the Morae Manager) is presented in Figure 7.

In this study we have used four main categories to mark in the remote viewer during the concurrent observation:

1. immediate – technical problem,
2. immediate – revision,
3. delay – technical problem,
4. delay – revision.

⁴ More information about Morae can be found on the Techsmith website <http://www.techsmith.com>

These markers are at a later moment elaborated in the Morae Manager with all the items of the categorization model (section 3.3.1).

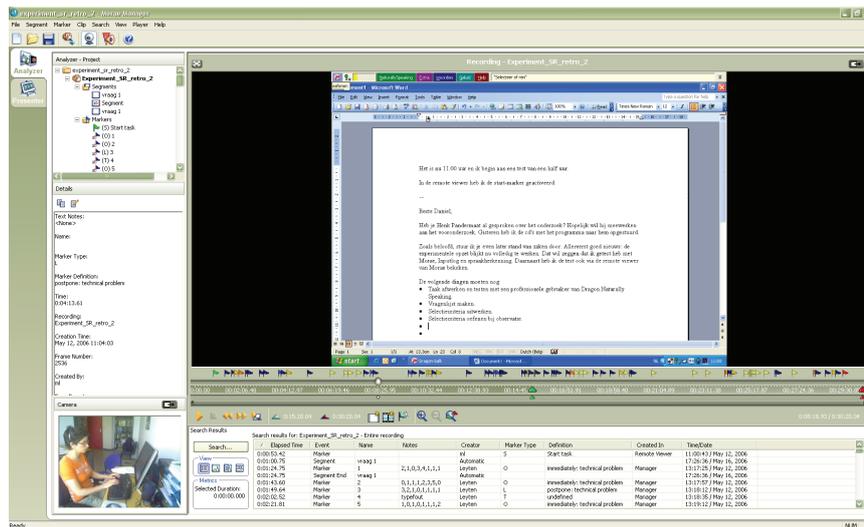


Figure 7. Example of view in Morae: screen, writer and markers (flags).

2.3.3 Protocol analysis

In this study we collected additional data using (stimulated) retrospective interviews. Retrospective interviews allow researchers to get a glimpse of the writers' strategies and decisions after the writing process is finished. They have the advantage of allowing writers to explain and reflect on the decisions without interfering directly with their attention to the task, freeing a writer from the cognitive load that the concurrent verbalization of the think aloud would require (Greene & Higgins, 1994).

Of course, the concurrent thinking-aloud method is impossible in our study since our subjects are already using speech recognition. Therefore this description needs to be read as evidence for using the stimulated retrospective interview as a valid method to describe why writers solve errors in the TPSF immediate or delayed.

Both the gathering of concurrent thinking aloud as well as retrospective thinking aloud protocols have been criticized as a research instrument. The negative influence which the thinking-aloud task has on the main writing task is crucial. This phenomenon is called reactivity (Janssen, Van Waes, & Van den Bergh, 1996), that is, the double cognitive workload in the concurrent thinking-aloud condition. Van den Haak, De Jong & Schellens (2003; 2004; 2006) show that this reactivity is not present in retrospective interviews. They compared retrospective and concurrent thinking-aloud protocols for the evaluation of an online library catalogue and a municipal website. Because the retrospective thinking aloud participants only had to perform one task at a time, they were given the opportunity to verbalize more problems, both

task-related and non-task-related. Results also showed that there were no differences in terms of number of problems detected in both methods. The conditions they compared largely reveal similar types of problems in similar frequencies. This means that the research methods proved equally useful in detecting relevant problems, but that the quality of the task-performance is negatively influenced when concurrently thinking-aloud, but not in retrospective interviews.

The most general retrospective verbalizing instruction asks the subject to report everything he or she can remember about the writing process. If the participant is interviewed immediately after performing the process, the model predicts that some previously heeded information will still be in the short term memory, permitting direct reporting by the processes described earlier and facilitating retrieval of additional information stored in long term memory in episodic associations that were formed when the information was heeded (Ericsson & Simon, 1980, p. 226).

The retrospective thinking aloud method is a relevant research method if properties of the solution process are relevant to the theory. This method is a means to validate or construct theories of cognitive processes, in particular of problem-solving. (Van Someren, Barnard, & Sandberg, 1994, p. 9)

Since writing is generally known as a continuous problem-solving process and since we are interested in *why* writers opt for various strategies, the retrospective interview is a valid instrument. Ericsson and Simon (1980) have shown that verbal reports, elicited with care and interpreted with full understanding of the circumstances under which they were obtained, are a valuable and thoroughly reliable source of information about cognitive processes. However, a variety of remarks that need to be taken into account are mentioned (Ericsson & Simon, 1980; Van Someren, Barnard, & Sandberg, 1994). The short term memory has limitations as to what can be stored. If the delay between the task and the interview is too long, the information is not stored anymore in the short term memory. On the other hand, writers may use a single experience to generalize from this instance to typify their approach to writing. In that case, the working memory constructs cognitive processes that rely on a single instance. Finally, questions in the retrospective interview need to be formulated carefully and must not be leading.

The degree to which retrospective verbalization must rely on retrieval from LTM can be minimized by studying cognitive processes of short duration, where the verbal responses lag the task processes by only a brief interval. (Ericsson & Simon, 1980, p. 226)

Another solution to the limitation of short term memory is to direct the participants to concrete situations by showing them various fragments of the writing process again, so called stimulated recall (Lindgren & Sullivan, 2003; Smagorinsky, 1989). In other words the probes that are given to participants need to be specific enough to elicit the information actually sought. If they are too general the subjects will use inferential processes to fill out and generalize incomplete or missing memories.

As we have stated before, we have used converging methods to be able to cross-

validate the consistency of our data from the different methods (general questionnaires, logging of writing process (Inputlog), locating critical incidents (Morae) and finally retrospective interviewing). In the next section we describe the procedure of the experiment.

2.4 Procedure

The participants received the background information about 'well-being' related to the task a week beforehand, together with the questionnaire. At the beginning of the experiment they received the actual task 'write a report that contains advice to pay more attention to well-being at the workplace'.

Before they started writing, the participants activated their speech profile of Dragon Naturally Speaking on the experimental computer (each computer and working environment needed to be 'linked' to the speech profile). The researcher started Inputlog, the logging add-on in Dragon Naturally Speaking, Morae Observer and Morae Remote viewer. In order to be able to calibrate the Inputlog data with the speech recognition data the participants needed to create a basis for this calibration⁵. After the participants received a short verbal clarification on the procedure a small timer was started, because the task was restricted to half an hour. The participants could keep an eye on the time via this timer (this was the only visual interference for the writers). After 15 minutes it gave a little beep as a reminder that the writing session was halfway.

Based on the theoretical implications, known problems and suggestions in the literature (Ericsson & Simon, 1980; Greene & Higgins, 1994; Van Someren, Barnard, & Sandberg, 1994) we collected the data for the retrospective interview immediately after the writing session. The interview was structured as follows.

First, we started with some more general questions about the writing process. The participants were asked to describe the general development of the writing process and they were asked to elaborate on what went well and what caused difficulties in more detail. On the printed copy of their text they indicated for each section if it was easy or difficult for them to write it. They also indicated possible errors in the final text.

The participants answered these questions via speech recognition in a Word document. While the participants answered the open questions, the researcher selected about 5 to 6 important fragments of the writing process that contained a combination of technical problems and revisions that were both immediately solved and delayed (in Morae Manager, cf. section 2.3.2).

In the next stage, we focused on critical incidents by showing the participants fragments of the writing process that we were interested in. The focus was based on a selection of fragments that were concurrently classified during the observation

⁵ The participants were asked to press the arrow up key, release it and say the word 'test' as quickly as possible. This procedure was repeated at least three times.

(cf. description of Remote Viewer Morae). We asked the writers in a semi-structured way about the reason *why* they chose different error correction strategies at various moments during the writing process. Theory suggests that writers who are asked to reflect on concrete examples of writing are more likely to provide more detailed information (Ericsson & Simon, 1980). Therefore, we showed the participants six fragments in which they either solve a technical problem or carry on a revision. The participants were asked to elaborate on their strategy on the basis of a very open question: 'Can you describe what happens here?' If necessary we continued asking questions about 'why' they solved the problems the way they did. For each participant we started with the same type of error to familiarize the participants with the technique of retrospective interviewing (The Dutch word 'welzijnswet' always appeared incorrect on the screen).

We ended the interview with some general questions about their use of the speech recognition software. For instance: Do you monitor the 'yellow bar' in Dragon Naturally Speaking? (This shows the progress of the dictated text and already reveals errors prior to appearing on the screen). Do you anticipate a slip of the tongue in any way? And finally, we inquired directly into their opinion about the influence of the TPSF on the screen. The retrospective interview is also logged by Inputlog and the add-on of Dragon Naturally Speaking.

3 Data analyses

The data analyses focused on three areas: (1) final output of writing task: a report of about one and a half page, (2) process data of the writing session, and (3) data from the retrospective interviews.

3.1 Product analysis

We have analyzed the final product of the writing task. The participants wrote a report of about one and a half page. We counted the number of words in the final text (Word tool). We also counted the number of errors in the final text.

3.2 Process analysis

To prepare the logging data gathered by Inputlog and Dragon Naturally Speaking 8.1 for statistical analyses we have made three major adaptations to the general logging file generated by Inputlog: (a) merging of logging data Inputlog and Dragon Naturally Speaking 8.1, (b) cleaning up the merged data by filtering double dictates, and (c) adapting the action times generated by Inputlog to the addition of speech data. Next to this we have merged the marker categories generated by Morae into the general logging file. In this section we describe the technical steps to merge the data files and the underlying theoretical choices. Subsequently, we describe various categories that we derive from the process data.

Inputlog and Dragon Naturally Speaking both generate very detailed process data. In computer terminology this is called the 'event-level'. An event is an action performed by the computer. Each event is characterized by several variables. If a writer types the letter 'm', the computer provides information about the type of event (keyboard press in), the related timestamp etc. (more detailed information about events can be found in chapter 8: Inputlog). Dragon Naturally Speaking 8.1 provides the same kind of data for the dictated segments: the content, the starting time and end time of dictated segments. However, yet there is no common reference point between both logging data sets. Therefore, we needed to create a calibration point to match the logging data of keyboard & mouse by Inputlog and the speech recognition data by Naturally Speaking. Figure 5 shows an example of the merged process data.

The shown data have been adapted from the original file in two ways⁶. Various dictated segments were incorrectly distributed over various events, causing double action and pausing times. So, first we deleted these double speech recognition events that were generated by Dragon Naturally Speaking⁷. Second we have recalculated the pausing times. Originally, Inputlog data are calculated on 'key in' actions. Next to technical reasons, we assume the actual 'key in' press to be the most distinguishing cognitive action in keyboard writing. However, this reasoning is not adequate enough when combining keystroke logging data with speech recognition data. Therefore, we recalculated the action times of the speech events.

The marker data entered in Morae during the 'real-time' observation sessions is the final process data that is merged into one overall process data file.

To sum up, the process data of the writing sessions consist of keyboard & mouse data from Inputlog, speech recognition data from Dragon Naturally Speaking and higher level analysis data from Morae. This combination of data allowed us to categorize the writing process data on a highly detailed level.

We have extended the analyses that are generated by Inputlog automatically as the current version of Inputlog provides only basic information about the writing modes. This enabled us to distinguish between various kinds of mode switches. We have categorized the transitions between computer events in 7 items:

1. no switch (event before and after transition is conducted in the same writing mode),
2. keyboard to mouse,
3. keyboard to speech,
4. mouse to keyboard,
5. mouse to speech,

⁶ In the next version of Inputlog these calculations will be automated.

⁷ Dictated segments were represented in various events instead of one event. We have opted to delete the double overlapping events and the connected action and pausing times. To calculate the new action time we have subtracted the beginning of the next event from the beginning of the dictated segment.

6. speech to keyboard, and
7. speech to mouse.

This enabled us to distinguish between mode switches that are related to a repair and the general mode switches.

The action times and pausing times were also calculated per writing mode. The data also helped us to focus on actions and pauses of various lengths. We have opted to analyze the pausing data above the threshold value of 1500 milliseconds. The data showed a trade-off point for this value: exploratory data analysis showed that most pauses under this threshold value level were situated within words (between letters), an analysis level that was less important for this study. We are aware that this level might vary slightly between participants (Wengelin, 2006). Pauses above the threshold value are probably related to higher cognitive activities than the automated processes of writing that are of no particular value in this study.

We have divided the writing processes in intervals by dividing the total process into ten equal parts (time based). This enabled us not only to describe the process data as such, but also the evolution during the writing processes.

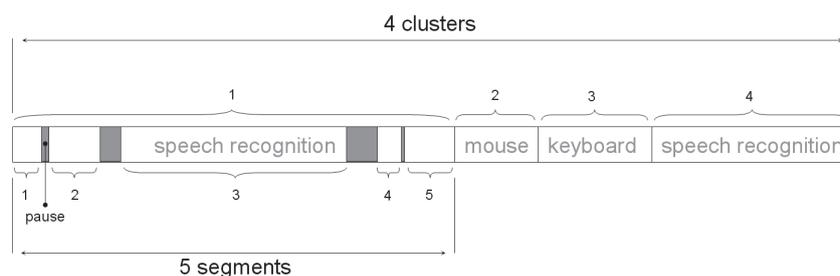


Figure 8. Clusters and segments in writing processes.

As mentioned before, Inputlog provides data per event. Although separate events are very informative, we would like to add an extra dimension to the relatedness of the events⁸. Writers can have a fluent writing process or it can be rather fragmented. To gain insight in fluency we have divided the writing process in several clusters and segments (Figure 8). Clusters are larger units of text production in one and the same writing mode. Clusters consist of smaller units that we will call segments (comparable to bursts as described by Hayes & Chenoweth (2006). Segments are smaller units within the clusters that again might – but need not be – split up by a threshold value (e.g. pausing time is larger than 1500 milliseconds)⁹. This subdivision gave insight in the way writers fragment their writing processes by switching between writing

⁸ The next version of Inputlog will integrate this functionality. In this study we have conducted the analysis semi-automatically, and we have developed the framework for the integration.

⁹ In this study clusters are the same as events, since we did not use a threshold value to split the clusters in various parts.

modes on the one hand and enabled us to take a closer look at fluency by analyzing the length of the dictated clusters on the other hand. In our opinion longer clusters stand for higher fluency, as well as longer texts (written in the same time span) stand for higher fluency (Quinlan, 2004; Ransdell & Levy, 1996). Fluency is one of the widespread positive characteristics of speech recognition.

3.3 Protocol analysis

Since this study deals with the writing process of professional speech recognition users, we also decided to use the combination of Inputlog and Dragon Naturally Speaking as a logging instrument to record the retrospective interview. Therefore, the participants kept on their headset during the retrospective interview (see Figure 6). Inputlog normally starts logging in Microsoft Word. However, in this situation we have used DragonPad provided by Dragon Naturally Speaking, because it has less built-in functionalities that could interfere during the interview. While the participants paid attention to the Morae video, the researcher simultaneously monitored the spoken output by the participants. From time to time the spoken output stopped or activated an item in the task menu. So the researcher had to make sure that the interview was recorded as accurately as possible. Of course, we also used a digital sound recorder to complete the transcriptions (and as a back-up for the logging with Dragon).

Between each question the researcher entered a number to indicate the successive episodes in the retrospective interview (Morae uses the term 'segments' to describe this functionality). The time it took to transcribe the retrospective interviews was substantially reduced by this procedure. Of course, the dictated text had to be checked, but this took far less time than transcribing the complete protocols. It takes researchers a lot of time to accurately transcribe every stuttered utterance or paraphrasing by the participants. Dragon Naturally Speaking, on the other hand, provides this information already quite accurately.

3.3.1 Repairs

In this section we describe the categorization model that we used to describe the repairs. The most important variables in this study are the moment the repair is performed (immediate or delayed) and the type of repair (technical problem versus revision). These main variables were analyzed in two instances. The first one was during the actual writing session of the participants. The second analysis was done afterwards in order to describe the repairs in more detail. During the writing session the researcher marked all the errors using the Morae Remote Viewer.

Four levels were distinguished:

1. immediate solving of technical problem,
2. immediate solving of revision,
3. delayed solving of technical problem, and
4. delayed solving of revisions.

Writers either corrected errors immediately or delayed this correction. Errors are categorized as 'delayed' if two criteria were met (a) the segment needed to appear on the screen, and (b) the participant had taken a look at the TPSF and continued with text production. For this latter purpose we used the registration of the webcam view in the remote viewer of Morae.

As we have seen before (Leijten & Van Waes, 2005b, also chapters 2, 3 & 4) speech recognition causes various errors in the TPSF that we define as 'technical problems'. These technical problems differ from the well-known typing errors in keyboard based word processing, because not only the writer is responsible for the error, but the speech recognition technology as well. Speech recognition creates error types that are sometimes difficult to grasp for writers. Therefore, we distinguish between technical problems on the one hand and regular revisions on the other hand. Revisions are changes to the TPSF.

Revisions in this study are based on the traditional revision taxonomies. We only focused on externalized revisions (Lindgren & Sullivan, 2006). Immediate revisions are made at the point of utterance or in the near vicinity. Delayed revisions are made at a later stage. As a consequence, revisions that we categorize as delayed might in fact be immediate. For instance, revisions that are corrected in the second draft to revise problems that were created in the first draft are seen as 'delayed revisions', because we consider them to be delayed with respect to the point of utterance. We are aware that some of these revisions might be the result of an evaluation made during (re)reading at a later stage and that not all of the delayed revisions are necessarily strictly related to the initial text production. A refinement of the classification can only be based on more elaborate mental traces (simultaneous thinking aloud protocols). To be consistent over the various revisions, we have opted to define 'delay of revision' in terms of its manifestation in the text production, that is the point of utterance and not purely the mental representation.

The repairs can be described by eight characteristics: (1) use of writing mode to solve errors, (2) switches between writing subprocesses, (3) dealing with errors in the first draft or the second draft, (4) the level of the error, (5) the distance between the point of utterance and the correction, (6) the location of the error in the sentence, (7) the error type and finally, and (8) the goal of the revision.

(1) Use of writing modes

Writing with speech recognition makes it still possible to use keyboard and mouse. Therefore, we have analyzed the writing mode which the participants used for error correction. Next to that, we have taken the writing mode into account before they started error correction. Because of this we did not only categorize the writing modes, but also the switches between writing modes. Switching between writing subprocesses could stimulate or force writers to simultaneously switch modes.

(2) Switching between subprocesses

The writing process can be divided into several subprocesses: planning and translating, programming and execution and reading and editing (Kellogg, 1996). In this study we are mainly interested in the subprocess of (re-)reading and editing. More specifically, we focus on the effect of errors on the writing process: do deficiencies in the TPSF have consequences for the structure (i.e. fragmentation) of the writing process? Therefore, we have analyzed the switches from text production to text correction (either a technical problem or a revision). Secondly, we have looked at the switches from one type of error correction to the other, for example, solving a technical problem and subsequently performing a revision or vice versa. Thirdly, we have examined the instances in which writers do not switch. Those instances are defined as revision episodes, or in this study repair episodes¹⁰ (Severinson Eklundh, 1994; Severinson Eklundh & Kollberg, 2003).

(3) Error correction in first and second draft

An extension of the distinction between immediate and delayed error correction is analyzed by the number of errors corrected in the first draft and the second draft. We distinguished the first draft as the process time of the writing process that leads to the first draft of a text. The period of the second draft is the explicit rereading phase at the end of the writing task.

(4) Level of error

Previous research on computer based word processing and classical dictating has distinguished various levels of revision. In computer based word processing writers revise a lot at a very low level. Writers that use classical dictating on the other hand conduct more high level revisions (Lindgren, 2005; Schilperoord, 1996; Van Waes & Schellens, 2003). Since speech recognition is a very specific writing mode we describe the various levels of error correction in detail: punctuation, letter, word, segment, sentence, paragraph, voice command.

(5) Distance between point of utterance and error correction

Writers have several options to initiate error correction: in the same sentence, but also across the sentences or even across the paragraphs. Therefore, we used a measure to express the distance between the point of utterance and the location of the revision. Next to this, we added a 'visibility' factor to this category to indicate whether the location of the revision was visible (on screen) when it was initiated. In other words, this is the visibility on the screen relative to the point of utterance.

(6) Location of the error in the sentence

Since we are interested in the location of an error during text production, the error location is a process measure. We have categorized the error location in the sentence

¹⁰ In a repair episode only the first errors that are located in the same sentence are coded as immediate repairs. The subsequent errors are delayed. This way, we isolated the most 'dominant' errors from the less important ones.

based on mathematical reasoning. If a sentence produced so far contained three words, the first is 'beginning', the second is 'middle' and the final is 'end'. If a sentence contained an even number of words, the middle part of the sentence contained the 'extra' word.

(7) Type of revision

In general, revision types can be divided into formal and content revisions. Formal revisions are related to the form of the text, like putting headings in bold. Content revisions change the content of the TPSF.

(8) Goal of revision

Revisions have several goals. Faigley & Witte (1981) have distinguished five: (1) writers add new text to the TPSF (addition), (2) writers delete incorrect text, (3) writers replace one text by another (substitution), (4) writers rearrange texts (permutation) and (5) writers contract text segments (consolidation). Next to these categories we have added cosmetic changes. These are changes that do not actually alter the goal of the text, but sharpen or lessen the focus of the text, for example putting a header in bold, inserting a bulleted list (see also Lindgren, 2005; Van Waes & Schellens, 2003).

4 Results

In this chapter we describe several characteristics of the writing process of professional speech recognition users that write a business report. We are mainly interested in the cognitive processes it takes to deal with errors in the TPSF. To answer this question we have analyzed the writing processes of 10 professional writers. The duration of the writing sessions varied between 26'38" and 37'41" minutes. The mean time of the writing sessions is 30'51" ($SD = 3'01''$). The distribution between the actual writing time and the pausing time is 32% versus 68%. In other words, writers pause for about 20 minutes and actually write for 10 minutes. Of the actual writing time writers use speech recognition on average for about 6 minutes, and keyboard & mouse¹¹ for about 4 minutes.

In this section we describe the results from three perspectives. We start with a general analysis of the overall data. Previous research on writing processes shows that these processes often differ between various writers. Therefore, we also distinguish between writer groups that are characterized by similar writing (repair) strategies. Finally, we will illustrate the data by means of a case study.

¹¹ The addition of speech recognition led to the following calculation of total action time in keyboard and mouse: total process time - pausing time (threshold value of > 1500msec) - action time speech = action time keyboard + action time mouse. Time in speech sometimes overlapped with keyboard and mouse. Therefore, we have taken the speech recognition times as a starting point and computed the action time in keyboard and mouse on the basis of the speech recognition time.

4.1 General

In this study we have coded 521¹² repairs during the 10 writing sessions of about half an hour. In Table 1 the number of repairs per minute is presented. The repairs have been divided in two groups: repairs that are solved immediately and repairs that are delayed, that is, solved at a later stage. Errors are categorized as 'delayed' if two criteria were met (a) the segment needed to appear on the screen, and (b) the participant had taken a look at the TPSF and had already continued producing new text.

Table 1. Number of corrections of technical problems and revisions per minute

	Immediate (<i>n</i> = 368)		Delayed (<i>n</i> = 153)	
	Number	<i>SD</i>	Number	<i>SD</i>
Technical problems (<i>n</i> = 309)	0.83	0.45	0.19	0.22
Revisions (<i>n</i> = 212)	0.38	0.28	0.32	0.20
Total repairs (<i>n</i> = 521)	1.21	0.59	0.50	0.23

The table shows that the writers on average conduct 1.71 repairs per minute (*SD* = 0.55). Of all repairs, technical problems represent 59% and revisions about 41%. In absolute numbers, the number of revisions in each writing session varied from 8 to 35. The number of repairs that are either performed immediately or delayed differed. The participants preferred performing about 70% of the repairs immediately and delayed 30% of the repairs. Particularly technical problems are solved immediately more often (80%). For revisions this percentage is significantly lower: 55% (χ^2 (*df*=1) = 43.7, *p* < .001).

Table 2. Number and percentage of technical problems and revisions

	Immediate		Delayed	
	Number	%	Number	%
Technical problems (<i>n</i> = 309)	252	82%	57	18%
Revisions (<i>n</i> = 212)	116	55%	96	45%
Total repairs (<i>n</i> = 521)	368	71%	153	29%

The technical problems and revisions are performed at different levels. In general the distribution for the repairs is 70% immediately performed and 30% postponed. However, if we split the data and take a look at the error level (ranging from letter to paragraph) we see an even more detailed pattern. Technical problems and revisions that are smaller than whole sentences are more often solved immediately (letter, word, segment: 76%), sentences and sections are in 60% of the cases delayed.

Most of the errors repaired immediately are located at the point of utterance as measured in the on-line process. This might be at the end of the sentence, but also in the middle of the sentence. Error detection might lead to an interruption of the

¹² Although we know that typing errors might also influence the course of the writing process, we have opted to filter these errors that were immediately solved and interpreted them as noise. These types of errors were unequally distributed over the participants (58% of these error types were made by one participant).

production of an utterance. This is also the location that contains most of the errors (back of the utterance 60% vs. middle and beginning 40%). Errors in the final part of the sentence are corrected immediately in 82% of the cases. In general, writers prefer solving the technical problems and revisions with keyboard and mouse. The immediate repairs are solved with keyboard and mouse in 83% of the cases, and the delayed repairs in 86%. In 48% of the instances writers switch from speech to keyboard or mouse to make a repair. Only in 15% is it the other way around. In 30% they are satisfied with the writing mode that they are using for text production to perform a repair. The remaining 7% of the switches are between keyboard & mouse. If we analyze the instances in which writers prefer to keep on writing in the same writing mode we see a contrastive pattern. Writers are most willing to continue writing with keyboard or mouse to perform a repair. In 95% of the cases they will not switch between writing modes. However, for speech recognition this percentage is 5%. In only 5% of the instances writers continue writing with speech recognition to solve a technical problem or to revise their text.

Next to switches between writing modes (total of 1486) we have analyzed the switches between writing subprocesses (Table 3). The most frequent switch is from text production to revision ($n = 294$, $M = 56\%$). Next are the repair episodes in which solving one technical problem leads to solving another technical problem, or a revision to another revision ($n = 145$, $M = 28\%$, this is also called a revision episode). Finally, solving a technical problem can trigger a revision, and a revision can lead to solving a technical problem ($n = 82$, $M = 16\%$). If we ignore the repair episodes in which the correction of a technical problem leads to another correction of a technical problem (the same holds for revisions), then we see that 66% of the switches between writing subprocesses are done immediately. Table 3 shows the switches for the various writing subprocesses.

Table 3. Number and percentage of switches between subprocesses

	Immediate		Delayed	
	Number	%	Number	%
Text production to revision				
Production to technical problem ($n = 194$)	182	94%	12	6%
Production to revision ($n = 100$)	76	76%	24	24%
Switch repair subprocesses				
Technical problem to revision ($n = 53$)	22	42%	31	58%
Revision to technical problem ($n = 29$)	14	48%	15	52%
Repair episode				
Technical problem to technical Problem ($n = 86$)	56	65%	30	35%
Revision to revision ($n = 59$)	18	31%	41	69%
Total ($n = 521$)	368	71%	153	29%

Writers solve most repairs in the first draft of a text (87% first draft ($n = 453$) versus 13% second draft ($n = 68$), see Table 4. If we take a closer look at the repairs solved immediately this percentage increases to 99%. Only 1% of the repairs that is solved

immediately ($n = 2$) occurred in the 2nd draft. The repairs that are delayed by the writers are more equally distributed: 57% in the first draft ($n = 87$) and 43% in the second draft ($n = 66$).

Table 4. Number and percentage of technical problems and revisions in the 1st and 2nd draft

	Immediate		Delayed	
	Number	%	Number	%
General				
1st draft ($n = 453$)	366	81%	87	19%
2nd draft ($n = 68$)	2	3%	66	97%
Technical problems				
1st draft ($n = 268$)	252	94%	16	6%
2nd draft ($n = 41$)	0	0%	41	100%
Revisions				
1st draft ($n = 185$)	114	62%	71	38%
2nd draft ($n = 27$)	2	7%	25	93%
Total ($n = 521$)	368	71%	153	29%

We analyzed the level of repairs by dividing the repairs in low-level versus high-level repairs (Table 5). The low-level repairs (letter & punctuation) are mostly performed immediately (70%). For the higher-level words and segments repairs this preference increases to 78%. This is caused by the preference to solve the majority of technical problems at this level right away.

Table 5. Number and percentage of level of repairs

	Immediate		Delayed	
	Number	%	Number	%
Letter & punctuation ($n = 176$)	123	70%	53	30%
Word & segment ($n = 222$)	173	78%	49	22%
Sentence & paragraph ($n = 48$)	19	40%	29	60%
Total ($n = 446$)	315	71%	131	29%

Writing with speech recognition can cause errors ranging from letter to segment level. It is almost impossible to write a whole sentence completely incorrect. Therefore, the largest number of delays (60%) of high-level repairs (sentence & paragraph) is based on revisions. More than two-thirds of their revisions are content based (content based: $M = 68\%$ vs. formal: $M = 32\%$).

The revisions are equally distributed over formal and content revisions (immediate: 54% vs. delayed 46%). However, the kinds of revisions that are executed immediately differ from those that are delayed. Table 6 shows a selection of the most common type of revisions and the moment that they are performed. The type of revision seems to influence the preference for immediate delay or not to a certain extent (addition vs. rest of the revisions).

Writers show a preference to immediately delete text, substitute text and make cosmetic changes. The pattern for additions is different. Additions to the text are more often delayed.

Table 6. Number and percentage of type of revision

	Immediate		Delayed	
	Number	%	Number	%
Addition (<i>n</i> = 46)	12	26%	34	74%
Deletion (<i>n</i> = 23)	17	74%	6	26%
Substitution (<i>n</i> = 51)	38	74%	13	26%
Cosmetically (<i>n</i> = 78)	44	56%	34	44%
Total (<i>n</i> = 198)	111	56%	87	44%

Writers prefer repairing most of the errors visible on the screen related to the point of inscription. Only 5% of the repairs are conducted after a scroll or mouse movement outside the immediate visibility of the screen. For example, a writer starts a revision episode at the beginning of the previous screen and consequently revises outside the visibility of the screen. Obviously, all these instances are delayed.

The errors that are visible on the computer monitor while the participants are writing are repaired with a delay in 25% of the instances. Table 7 shows the location of the errors that are visible on the screen.

Table 7. Number and percentage distance between error and repair

	Immediate		Delayed	
	Number	%	Number	%
Within sentence (<i>n</i> = 263)	241	92%	22	8%
Within paragraph (<i>n</i> = 132)	84	64%	48	36%
Outside paragraph (<i>n</i> = 49)	9	18%	40	82%
Total (<i>n</i> = 444)	334	75%	110	25%

Writers repair more than half of the errors below the sentence level (59%). About 30% of the errors is solved within the paragraph and 11% beyond the paragraph level. Errors below the paragraph level are preferably solved immediately; errors beyond the paragraph level are in 82% of the cases corrected with a delay. This pattern is consistent for the technical problems as well as for the revisions. However, errors within the paragraph are more often delayed if the errors are revisions (immediate: *n* = 22, 45% vs. delayed: *n* = 27, 55%).

To conclude, we would like to summarize the characteristics of both immediate and delayed repairs. If a writer detects a technical problem in the TPSF that occurs close to the end of the utterance (within the same sentence) he will probably prefer correcting it immediately. Especially if that problem occurs in the first draft and is a low-level error. It is most likely that these errors will be corrected by keyboard & mouse, even

if that requires a switch from speech recognition. In most instances writers will delete text or substitute text preferably in small repair episodes. On the other hand, text problems that lead to content related revisions have a higher chance to be delayed. Quite often these revisions will be delayed till the second draft and lead to additions of new text to the TPSF beyond paragraph boundaries.

As is the case with immediately corrected errors, the preferred writing mode is also keyboard & mouse, and again repair episodes arise, i.c. one revision leads to another revision.

4.1.1 Evolution of writing process over time

The writing process can be divided into smaller parts to gain more insight in the evolution of different processes during the writing process (Van den Bergh & Rijlaarsdam, 1996). We have divided the total process time of each writing session in ten equal intervals. Figure 9 shows the evolution of the writing process by showing the total action time of speech recognition versus keyboard & mouse per interval (left Y-axis). Next to this, the mean pause length per interval is presented (right Y-axis, both measures are on a logarithmic scale).

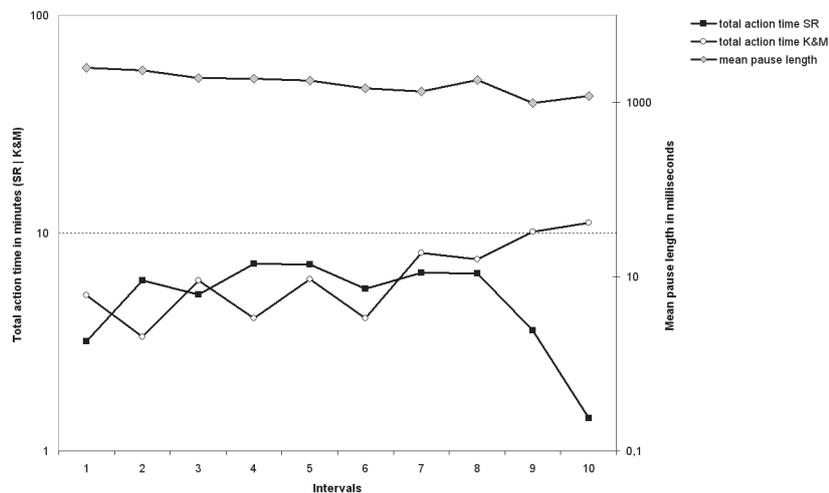


Figure 9. Evolution of writing process into ten intervals.

The total action time of speech recognition, keyboard and mouse, and the mean pausing length show a relatively constant pattern in these 10 intervals. If we combine the action times of speech and keyboard and mouse then we do see a difference ($F(9, 10059) = 2.56, p < .006$). The mean pausing length per interval also differs ($F(9, 10059) = 5.45, p < .001$). Especially the first two intervals differ from the last two intervals. In the beginning of the writing process the pauses are significantly longer

than in the final part of the writing process (Interval 1: $M = 2.49$, Interval 2: $M = 2.35$, Interval 9: $M = 0.98$ and Interval 10: $M = 1.18$; intervals 1 and 2 both differ from 9 and 10, according to Scheffé analysis, $p < .05$). Of course, initial planning in writing processes does lead to longer initial pauses

An exploratory analysis of the data shows two breaking points, after interval 3 and after interval 8. Therefore we have divided the writing process into three parts: begin, middle and end. The data show that the participants show a constant production pattern in the middle of the writing process (intervals 4-8).

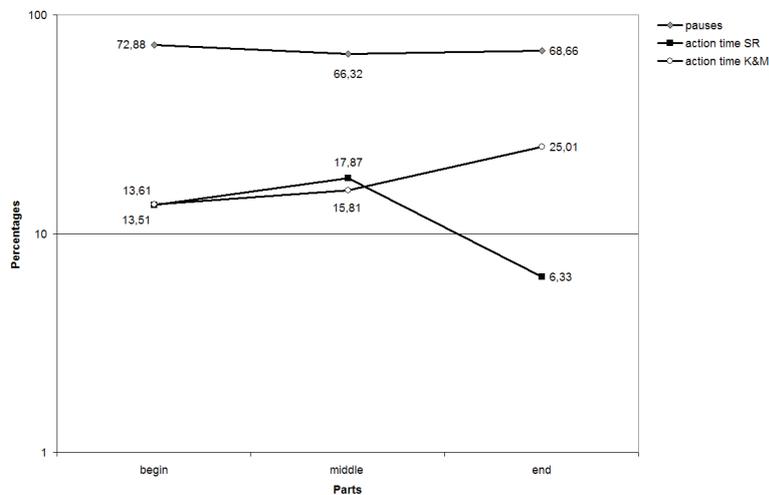


Figure 10. Evolution of writing process into three parts.

The aggregated data show a comparable use of speech recognition and keyboard & mouse in the beginning and middle of the writing process (Figure 10). In the final phase the main writing mode is keyboard & mouse. The total length of the pause duration per phase differs. In the production phase the total length of all the pauses is higher than in the initial phase and the final phase. The mean pausing time drops from the beginning to the end of the writing session (2.2 seconds to 1 second). The threshold value of 1500 milliseconds provides the same pattern: the mean pause length in the beginning is 9 seconds, in the middle part it is 7 seconds and in the end it is 5 seconds.

If we take a closer look at the relation between switching between writing modes and the number of repairs then we see that the number of switches is rather equally distributed over the 10 intervals, and so are the number of repairs (Figure 11: Y-axis, left). The final 2 intervals have twice as much of repairs as the first intervals (43 versus 84). If we divide the number of switches by the number of repairs then we clearly see the relation between both variables. The closer the ratio is to 1, the larger the one on one relation is between a mode switch and a repair.

The right Y-axis of the bar-graph in Figure 11 shows the ratio between the general switching behavior and the number of repairs per interval. On the left Y-axis the final two intervals show an increase in number of repairs, and also a decrease in the number of general switches in the final interval. This points at a very tight relationship between both variables in the final interval (1.51). That is, almost every switch is related to a repair.

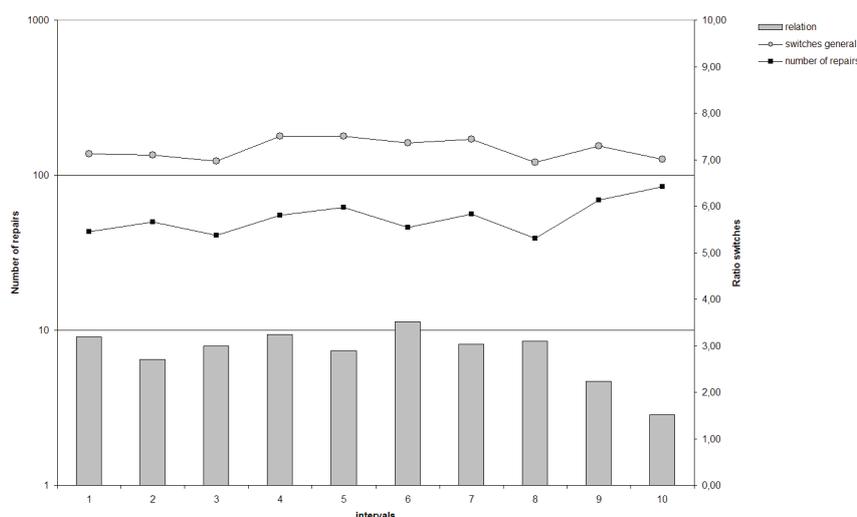


Figure 11. Switching behavior, number of repairs and the relation between both variables.

4.1.2 Some perspectives

Next to the process data, we also gathered protocol data. To finalize the general description we provide information on the various perspectives of the participants on speech recognition based writing. In general, the retrospective interviews show that the participants are very involved with the speech recognizer. Often the participants said ‘Normally, the speech recognizer would ...’. They often related their behavior to the professional context they work in, for instance, they formulated expectations about errors and anticipated on various levels.

The research context restricted them to train the speech recognition software systematically. In their working environment they are used to training the speech recognition software before they start a subtitling session. In this context they felt limited in time and therefore opted not to train the software. However, they were very much aware of the consequences. Several times they stated that if they had trained the software, various errors could have been prevented.

In Dragon Naturally Speaking writers get the opportunity to ‘pre-view’ the text that is to appear on the screen via a yellow bar. This bar is shown on top of the screen

and enables writers to take a look at the TPSF before it is really transferred to the Word document. Apparently, the distribution of participants that keep an eye on the bar and those who do not track this information is fifty-fifty. Some participants actually read the content of the bar, while others only check if the speech recognizer is responding. A few participants indicated that the yellow bar provided them with information to anticipate on speech misrecognition errors.

Writers mostly anticipate on errors in the TPSF if they 'feel' a slip of the tongue. In those instances they all claim to interrupt text production and to switch to monitoring the TPSF, in order to be able to correct the expected error. Another type of anticipation which the participants mentioned was the rephrasing of words prior to dictation: for example various writers claimed to have deliberately used synonyms for the company name used in the report.

4.2 Writer groups

As stated before, previous research on writing processes shows a variation in writing profiles (Galbraith, 1996; Kieft, Rijlaarsdam, Galbraith, & Van den Bergh, to appear; Lindgren & Sullivan, 2006; Van Waes & Schellens, 2003). At first glance, the writers in this study also show variation in the number of technical problems and revisions (e.g. standard deviations number of repair, Table 1). This might be related to various writing preferences on how to deal with errors in the TPSF.

The total number of repairs per person varies from minimum 30 to maximum 70 in their writing session. Table 8 shows the number and percentages of immediate versus delayed error correction per participant. The number of immediate corrections varies from 9 to 62, and the delayed corrections from 5 to 25. In addition to this general distribution of repairs, we have further explored the data by conducting a hierarchical cluster analysis in which we included the most important variables in this study: the number and distribution of the repairs (moment, type & density), action time per mode & mode switches, pausing times, fluency). The cluster analysis shows three groups. We have also tested this tri-partition on the main variable, immediate or delayed error correction. Chi-square tests confirmed that the three groups differed significantly ($\chi^2(df=2) = 35.2, p < .001$; $\chi^2(df=1) = 18.3, p < .001$).

Table 8. Number and percentage of immediate and delayed error corrections

	Immediate		Delayed	
	Number	%	Number	%
Eva (<i>n</i> = 69)	62	90%	7	10%
Ivan (<i>n</i> = 32)*	27	84%	5	16%
Sven (<i>n</i> = 54)	45	83%	9	17%
Norah (<i>n</i> = 69)*	54	78%	15	22%
Will (<i>n</i> = 53)	41	77%	12	23%
Gerard (<i>n</i> = 53)	35	66%	18	34%
Ethan (<i>n</i> = 70)	45	64%	25	36%

	Immediate		Delayed	
	Number	%	Number	%
Andrea (<i>n</i> = 57)	36	63%	21	37%
Thomas (<i>n</i> = 30)	14	47%	16	53%
Ben (<i>n</i> = 34)	9	27%	25	74%
Total (<i>n</i> = 521)	368	71%	153	29%

To describe the results of the experiment in more detail, we opted to divide the total dataset into three groups¹³. Group 1 consists of Eva, Sven and Will. Their profile can be defined as ‘handle’; they prefer solving repairs immediately¹⁴. Group 2 consists of Gerard, Ethan and Andrea. They delay more repairs than the handle group. Further analyses will show that the difference in delaying preference is mainly based on the revision category. Therefore, this second group will be defined as ‘postpone (r)’, in which the (r) stands for revision. The final group consists of Thomas and Ben, who both delay more repairs, than that they correct them immediately. They also perform fewer repairs in total than the two other groups. The repairs that they delay most are related to technical problems. That is why, this group will be defined as ‘postpone (tp)’, in which (tp) stands for technical problem.

To summarize, we have described three distinctive writer groups. Group 1 ‘handle’ corrects 83.5% of the repairs immediately. For group 2 ‘postpone (r)’ the percentage of immediate repairs is 64.5% and group 3 ‘postpone (tp)’ delayed 64% of their repairs.

Table 9. Number and percentage of immediate and delayed repairs per writer group

Groups	Immediate		Delayed	
	Number	%	Number	%
Handle (<i>n</i> = 176)	148	84%	28	16%
Postpone (r) (<i>n</i> = 180)	116	64%	64	36%
Postpone (tp) (<i>n</i> = 64)	23	36%	41	64%
Total (<i>n</i> = 420)	287	68%	133	32%

In this section we will compare the characteristics of the three writing profiles that we have distinguished. The writers differ mainly in their preference to solve errors in the text immediately or not. In general, the ‘handle’ and ‘postpone’ profile have a comparable number of errors to correct. The writers with the ‘postpone profile (r)’ have a larger number of delayed error corrections (more than twice as many as the handle group). The final profile, which we call ‘postpone-profile (tp)’ also has a larger number of delayed errors than the handle group, as well as a lower number of errors

¹³ The names of the participants are fictive: the sex and the initial are kept the same.

¹⁴ At first sight, it may look as if Ivan and Norah complement this profile. However, Norah showed a very different profile in the cluster analysis. Since Ivan’s logging data of Dragon Naturally Speaking are missing due to technical problems, we have opted to treat him in the profile analyses as a missing value. Consequently, we have disregarded the data of Norah and Ivan in the following analyses.

in general. Table 10 shows the distribution of the technical problems and revisions over the categories immediately and delayed.

Table 10. Distribution of the technical errors and revisions per moment

Groups	Immediate		Delayed	
	Number	%	Number	%
Handle (<i>n</i> = 176)				
Technical problems	83	93%	6	7%
Revisions	65	75%	22	25%
Postpone (<i>r</i>) (<i>n</i> = 180)				
Technical problems	95	82%	21	18%
Revisions	21	33%	43	67%
Postpone (<i>tp</i>) (<i>n</i> = 64)				
Technical problems	14	35%	26	65%
Revisions	9	38%	15	62%
Total technical problems	192	78%	53	22%
Total revisions	95	54%	80	46%
Total (<i>n</i> = 420)	287	68%	133	32%

The writers with the 'handle profile' solve technical problems as well as revisions immediately. The 'postpone (*r*)' group on the other hand, prefer solving technical problems immediately, but they postpone revisions. The number of errors that they delay is twice as high as the 'handle' group (handle: *n* = 28 versus postpone (*tp*): *n* = 64). The postpone (*r*) group has a rather high number of technical problems that they delay. The 28 errors that they delay number to 63% of all the delayed errors.

The use of writing modes seems quite comparable for the three profiles. Repairs are preferably performed in keyboard & mouse and on average writers switch from speech recognition to keyboard & mouse to solve a problem in 50% of the cases. Only the postpone (*tp*) group has a higher number of 'no-switches'. They also switch less from speech recognition to keyboard & mouse. These findings are consistent with the assumption that they keep more repairs for the second draft of the TPSE. Consequently, they prefer staying in the keyboard & mouse modus to correct their text in the second draft.

The switches between subprocesses shed light on the preferential difference in immediate correction or delayed correction (Table 11). Most switches between the subprocesses are of course from 'production' to 'error correction'. The next category is 'no switch' and finally the switches within a repair episode. The handle group has a small number of delayed repairs that are clustered and, therefore, the percentage of immediate repairs in an episode is rather high (70%). The postpone (*r*) group, on the other hand, does have a large number of repairs delayed.

Table 11. Number and percentage of switches between writing subprocesses

Groups	Immediate		Delayed	
	Number	%	Number	%
Handle (<i>n</i> = 176)				
Production to error correction	106	91%	10	9%
Repair episode	12	71%	5	29%
No switch	30	70%	13	30%
Postpone (r) (<i>n</i> = 180)				
Production to error correction	83	82%	18	18%
Repair episode	10	30%	23	70%
No switch	23	50%	23	50%
Postpone (tp) (<i>n</i> = 64)				
Production to error correction	18	82%	4	18%
Repair episode	3	27%	8	73%
No switch	2	6%	29	94%
Total (<i>n</i> = 420)	287	68%	133	32%

The 'handle' group delays 30% of the repair episodes (revision to revision, technical problem to technical problem, revision to technical problem and vice versa). The 'postpone' group delays 58% of the episodes. This larger number (and percentage) is caused by the higher number of delayed revisions. The 'handle' group deals with revisions within an episode more or less in the same way as they conduct isolated repairs (immediate: *n* = 10, delayed: *n* = 11); the postpone (r) group delays these episodes four times more than that they conduct them immediately (immediate: *n* = 4, delayed: *n* = 16). The total number of delayed repair episodes is twice as high for the postpone (tp) group (postpone: *n* = 30 versus handle: *n* = 15).

In general, writers prefer solving about 90% of the errors in the first draft of the text. Those errors that are delayed are taken to the second draft in 43% of the instances. The three groups have a different preference in solving errors in both drafts. Table 12 shows the number and percentage of errors corrected immediately versus delayed corrections in both drafts.

Table 12. Number and percentage of error correction in first and second draft

	Immediate		Delayed	
	Number	%	Number	%
<i>General</i> (<i>n</i> = 420)				
1 st draft	285	80%	70	20%
2 nd draft	2	3%	63	97%
Handle (<i>n</i> = 176)				
1 st draft	147	88%	20	12%
2 nd draft	1	11%	8	89%
Postpone (r) (<i>n</i> = 180)				
1 st draft	115	76%	37	24%
2 nd draft	1	4%	27	96%

	Immediate		Delayed	
	Number	%	Number	%
Postpone (tp) (<i>n</i> = 64)				
1 st draft	23	64%	13	36%
2 nd draft	0	0%	28	100%
Total (<i>n</i> = 420)	287	68%	133	32%

The 'handle' group solves 95% of the problems in the first draft, leaving only 5% for the second draft. The 'postpone (tp)' group leaves 16% for the second draft and the 'postpone (r)' group 23%. If we focus on the delayed error correction then we see that the 'handle' group solves 70% (*n* = 20) in the first draft and 30% (*n* = 8) in the second draft. The errors that are delayed are all (but one) revisions in the first draft. In the second draft apparently several new technical problems occurred since this distribution is not this strict. The 'postpone (r)' group solves 42% (*n* = 27) in the second draft and the 'postpone (tp)' group even 55%. The preference to delay error correction can be extended to delaying error correction to the second draft. In general, the errors that need to be solved in the 2nd draft are both technical problems and revisions.

Since the 'handle' group has only very few instances in the second draft, we compared the delayed corrections of the postpone (r) group and the postpone (tp) group (Table 13).

Table 13. Number and percentage of delayed error correction in second draft

	Immediate		Delayed	
	Number	%	Number	%
Postpone (r) (<i>n</i> = 64)				
1 st draft	11	30%	26	70%
2 nd draft	10	37%	17	63%
Postpone (tp) (<i>n</i> = 41)				
1 st draft	1	8%	12	92%
2 nd draft	25	89%	3	11%
Total (<i>n</i> = 105)	47	45%	58	55%

Table 13 shows that the 'postpone (r)' group and the 'postpone (tp)' group have a comparable number of errors that they correct with delay in the second draft, 27 versus 28. However, both groups deal quite differently with these errors. The 'postpone (r)' group mostly delays revisions, while the 'postpone (tp)' group prefers revising the TPSF in the first draft. In the second draft only 3 revisions were performed, while 25 technical problems needed to be resolved. The 'postpone (tp)' group feels comfortable to take technical problems over the barrier of a first draft. However, this does not hold for the revisions. An explanation for this behavior could be that these writers assume that they will remember the technical problems but that they predict that this will be more difficult for revisions.

The percentage of time spent on the second draft varies between 4% and 11% of the total writing process time. The 'postpone (r)' group spends the most time on the second draft (11%) and the 'postpone (tp)' group follows with 10%. For the 'handle' group the percentage of time spent on the second draft is 4%.

If we take a closer look at the type of revisions per group then we see that the formal and content revisions are equally distributed for the 'handle' group (Table 14). The preference is again to revise both formal revisions and content revisions immediately (74%). The 'postpone (r)' group also has an equal number of formal and content revisions and they prefer delaying both types of revisions. The 'postpone (tp)' group shows a rather irregular pattern in a sense that formal revisions are preferably to be delayed. This relates to the above-mentioned explanation why certain errors are corrected within the second draft. Because formal revisions are postponed more often, we assume that they are found to be less cognitively demanding than content revisions.

Table 14. Number and percentage of formal versus content revisions

Groups	Immediate		Delayed	
	Number	%	Number	%
Handle (<i>n</i> = 87)				
Formal revision	34	74%	12	26%
Content revision	30	73%	11	27%
Postpone (r) (<i>n</i> = 63)				
Formal revision	13	42%	18	58%
Content revision	8	25%	24	75%
Postpone (tp) (<i>n</i> = 22)				
Formal revision	2	15%	11	85%
Content revision	5	56%	4	44%
Total (<i>n</i> = 172)	92	53%	80	47%

The 'postpone (tp)' group delays cosmetic changes to the TPSF. The other groups show a more equal distribution of the delayed revisions over the categories addition, deletion, substitution and cosmetic changes. In line with the other results the 'handle' group prefer revising 86% of the deletions & substitutions and 74% of the cosmetic changes immediately. Additions are done immediately as well as delayed. The 'postpone' groups prefers delaying about 65% of the changes. Deletions and substitutions, as well as cosmetic changes are almost as often performed immediately as they are delayed. Almost all additions, on the other hand, are carried out with delay.

Finally, we compare the levels and the distance at which the writer groups prefer correcting errors. In general writers prefer correcting errors at a lower level immediately and they postpone correction of higher-level deficiencies in the TPSF. The 'handle' group enhances its profile for immediate correction by scoring about 10% higher on each level than in comparison with the overall average (low-level letter & punctua-

tion = 82%, word & segment = 91%, high-level sentence & paragraph = 50%). On the other hand, 'postpone (tp)' group scores higher than the other groups on the delayed error correction of the lower-level errors. The explanation for this finding is that they simply do not carry out any high-level corrections. The 'postpone (r)' group is also consistent with the general findings on the lower-level errors. However, they postpone twice as many high-level errors compared with the high-level errors that they correct immediately (immediately = 8, delayed = 16). This is different from the 'handle' group, whose errors on sentence and paragraph level are more equally distributed.

Next, we analyzed the location of the error correction related to the point of inscription. In general most errors are corrected within the sentence (59%), second within the paragraph (30%) and finally beyond the paragraph (11%). The writers with the 'handle' profile have a higher number of errors that they correct within the sentence (72%). In 97% these errors are corrected immediately. The writers of the 'postpone (r)' group have a higher percentage of errors beyond the paragraph level (18%). In 86% of the cases they prefer delaying these errors. Since the writers in the 'postpone (tp)' group delay most of their errors, the distribution of the location where the error is corrected is more equally distributed than in the other groups. The distribution within the sentence and within the paragraph is also equally distributed over immediate and delayed error correction. The errors that go beyond the paragraph are all delayed.

To summarize, we have described the characteristics of both immediate and delayed repairs in the three writing groups. In the 'handle' group the errors are characterized by a high number of repairs in the writing session. Technical errors and revisions (each level and both formal and content) are immediately corrected. Repairs are immediately performed by switching from production to repair and also in small episodes (within the sentence or segment). Immediate is for this group of writers both at the point of utterance and in the first draft. Apparently, texts are 'sculptured' at the point of inscription. In the 'postpone (r)' group - who prefer delaying revisions - the general number of repairs is quite high, more specifically the number of technical problems (64%) in relation to revisions (36%). Technical errors are corrected immediately switching from production to repair in the first draft. Revisions are delayed and corrected in repair episodes (switching from technical problems to revisions). In general, the 'postpone (tp)' group who prefer delaying technical problems is characterized by a low number of repairs and more specifically a low number of revisions. Errors are solved immediately one at a time switching from text production to repair and then continue with text production again. So, only a limited number of errors are solved, and other errors are deliberately kept in the text. Technical problems that are delayed are solved in technical problem episodes. Delayed errors in the first draft are mainly revisions and delayed errors in the second draft are mainly technical problems.

4.3 Case studies

The description of the writer groups shows three distinct patterns. First, there are writers who prefer writing a first time final draft and solve technical problems immediately as well as revising the TPSF immediately. Second, writers who solve more than half of the deficiencies in the TPSF immediately, but who also delay or postpone various technical problems and revisions. Finally, writers who prefer delaying error correction and who have a preference to delay technical problems to a 2nd draft.

In this section we will illustrate the preferences of the groups by describing the most dominant writer of each profile. Table 15 provides information on the preference of the three selected writers to correct errors immediately or with a delay.

Table 15. Number and percentage of immediate versus delay strategy of three writing groups

	Immediate		Delayed	
	Number	%	Number	%
<i>Handle-profile: Eva (n = 69)</i>	62	90%	7	10%
<i>Postpone-profile (r): Ethan (n = 70)</i>	45	64%	25	36%
<i>Postpone-profile (tp): Ben (n = 34)</i>	9	27%	25	74%

The handle profile will be illustrated by Eva, who has the largest number of immediate repairs. The postpone (r) profile will be illustrated by Ethan, who has the largest number of delayed repairs. And finally, the postpone (tp) profile will be described by Ben, who has the most distinctive percentage of delayed repairs.

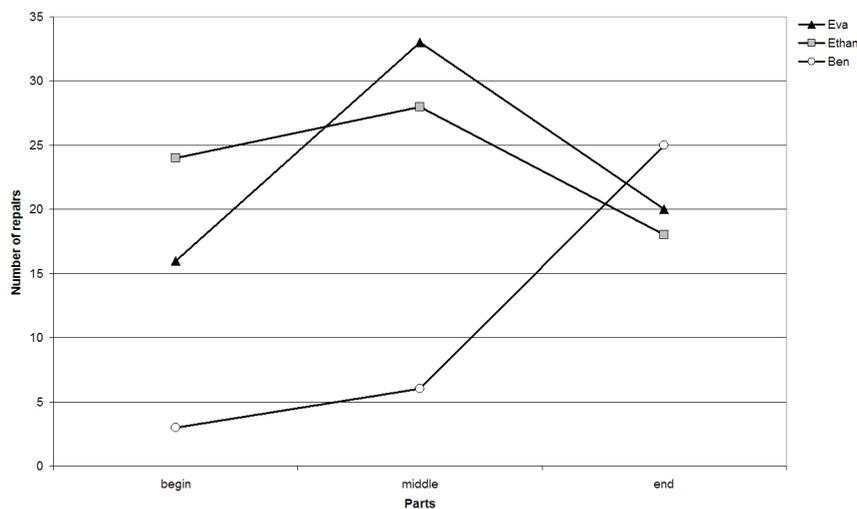


Figure 12. Distribution of number of repairs per part.

The writing processes of the three participants show a different progression. Eva and Ethan perform most repairs in the beginning and middle of the writing process. Ben,

on the other hand, solves most problems towards the end of the writing session. Figure 12 shows the distribution of the number of repairs after dividing the writing process in three intervals: beginning, middle and end (cf. section 3.2 process data). Eva and Ethan have a comparable number of repairs in their session. However Eva has more repairs at the beginning of the writing session and Ethan more at the end. In general, Ben has a low number of repairs, in particular at the beginning of the writing process. Like Ethan, he prefers delaying error correction until the end of the writing process. If we compare the number of technical problems and revisions that are either solved immediately or delayed, then we see a variation in preference. Table 16 shows the distribution between technical problems and revisions per preferred correction moment.

Table 16. Number and percentage of immediate and delayed repairs per individual

	Immediate		Delayed	
	Number	%	Number	%
<i>Handle-profile: Eva (n = 69)</i>				
Technical problems (n = 41)	37	90	4	10
Revisions (n = 28)	25	89	3	11
<i>Postpone-profile: Ethan (n = 70)</i>				
Technical problems (n = 44)	40	91	4	9
Revisions (n = 26)	5	19	21	81
<i>Postpone (tp) profile: Ben (n = 34)</i>				
Technical problems (n = 26)	4	15	22	85
Revisions (n = 8)	5	63	3	37

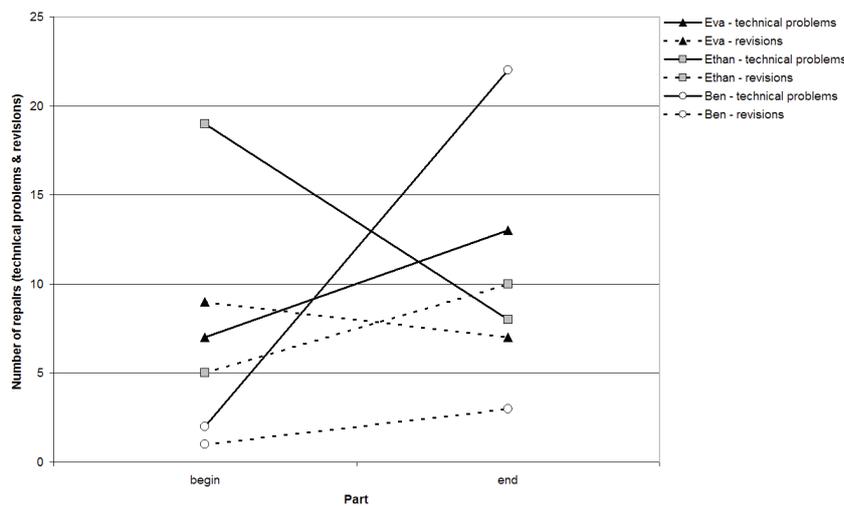


Figure 13. The number of technical problems and revisions in the beginning and the end of the writing process.

Eva and Ethan have a comparable number of technical problems and revisions (Eva: technical $n = 41$, Ethan: technical $n = 44$, Eva: revisions $n = 28$, Ethan: revisions $n = 26$). However, they differ in the moment they make revisions to their text. Eva prefers revising her text immediately. Ethan saves revisions for a later moment. And Ben, who delays just as frequently as Ethan, prefers solving technical problems at a later stage. Figure 13 shows this pattern in the course of the writing process. The number of technical problems and revisions are visualized for the beginning (interval 1 to 3) and the end (interval 9 and 10) of the writing process.

These results are reinforced if we take a closer look at the density of the repairs. We analyzed the number of seconds between two repairs.

Table 17. Mean density of repairs per interval (seconds between repairs)

Intervals	1		2		3	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
<i>Handle-profile: Eva (n = 69)</i>	35.06	30.20	27.92	28.62	14.43	19.35
<i>Postpone-profile (r): Ethan (n = 70)</i>	23.58	14.84	30.77	26.09	17.50	14.47
<i>Postpone-profile (tp): Ben (n = 34)</i>	193.65	214.16	93.50	106.36	8.40	7.28

The reason for the low density of Ben's repair behavior is his writing fluency. If we compare the various cluster durations in speech recognition Ben is extremely fluent. His clusters, that is, various segments of speech that are joined together, in the first part of the writing session have a mean length of 43 seconds. Eva and Ethan dictate for about 4.5 seconds in a row. The mean length of a single speech segment is 2.2. In the middle Ben is still far more fluent than Eva and Ethan. In the final part of the writing process he does not use speech recognition at all. He has switched to keyboard & mouse to correct his text.

Table 18. Mean length of speech cluster (in seconds)

Intervals	1		2		3	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
<i>Handle-profile: Eva</i>	4.71	7.36	5.78	12.01	7.25	21.93
<i>Postpone-profile (r): Ethan</i>	4.43	4.19	6.04	7.54	1.87	0.88
<i>Postpone-profile (tp): Ben</i>	42.61	61.30	29.40	25.67	0.00	0.00

Eva becomes more fluent during the writing process (mean speech length 4.71 versus in the first cluster and 7.25 seconds in the last). Ethan's writing process, on the other hand, gets more fragmented.

Table 19 shows the mean pause length in seconds of pauses that were longer than 1500 milliseconds. Eva's pauses are twice as long in the beginning compared to the end. Ethan shows a very equal distribution over his writing process. Ben's mean pauses are almost three times as long in the beginning.

Table 19. Mean pause length per interval

Intervals	1		2		3	
	M	SD	M	SD	M	SD
Handle-profile: Eva	8.97	18.64	5.84	6.54	4.11	7.48
Postpone-profile (r): Ethan	4.44	5.67	4.73	5.90	4.70	5.32
Postpone-profile (tp): Ben	10.60	23.01	5.67	6.04	4.17	4.73

4.4 Handle profile: Eva

Eva starts her writing sessions with the structure of her report. She formulates every heading and subheading and then starts with the first heading (Advice; knowledge transforming). This is a rather difficult paragraph to write because all the information needs to be synthesized in a formal advice to the board. Eva has rather long pauses in the beginning of her writing session.

Eva prefers writing a first time final version. She spends very little time on the second draft, only 1'31" minutes. She opts to correct her errors immediately. She does not feel the need to finish her sentence first. If an error occurs she opts to correct it immediately. Figure 14 shows a short excerpt from her writing session. This fragment is typical of the way in which she interacts with the TPSF and deals with errors. Figure 14 shows each event individually numbered, to translate each segment into English.

#	Time (minutes)	WritingMode	Original output	Intended output	ActionTime (ms)	PauseTime (ms)
1	0:11:50	3 - dict	ter preventie van beroepsrisico 's ,komma		3491	443
2	0:12:00	TP 1-2 3 - dict	en gg die voordurend	en <u>gm</u> die voordurend	2035	6121
3	0:12:05	1 LEFT			454	3555
4	0:12:06	1 LEFT (5)			1251	1625
5	0:12:08	1 BS			94	578
6	0:12:08	1 BS			78	172
7	0:12:08	3 - dict om			1147	630
8	0:12:11	1 END			62	1848
9	0:12:12	1 BS			765	297
10	0:12:13	1 BS			62	859
11	0:12:13	3 - dict	voordurend		1456	458
12	0:12:20	3 - dict	te evalueren en te verbeteren .lpunt		3072	5332
13	0:12:31	3 - dict	deze maatregelen		1775	7696
14	0:12:34	3 - dict	worden gebundeld		1735	1864
15	0:12:39	1 LEFT			906	2768
16	0:12:40	1 LEFT (3)			202	1391
17	0:12:41	1 SPACE			78	812
18	0:12:41	R 1 3 - dict	moeten		1207	244
19	0:12:44	1 END			63	1174
20	0:12:45	1 SPACE			47	1703
21	0:12:46	3 - dict	in een globaal preventie plan		2464	319
22	0:12:46	2 Movement			28219	0
23	0:12:50	3 - dict	en een jaarlijks actieplan		2294	3711
24	0:12:53	TP 3 3 - dict	moet	.	718	1288
25	0:13:14	2 Left Button			78	20333
26	0:13:14	2 Movement			359	78
27	0:13:15	1 ENTER			94	781

Figure 14. Fragment of writing process Eva.

The texts are translated as literally as possible from Dutch. To gain insight in the size of the errors the Dutch text should be compared to see visually how large the error is (underlined part in column intended output). This might differ from English

because of the rather literal translation¹⁵. Furthermore, the writing mode is shown: 1 for keyboard actions, 2 for mouse activities and 3 – dict for dictated segments. The technical problems are indicated with 'TP' and the revisions with an 'R'. So, the figure gives information on how the text appeared on the screen (original output) and the intended output. Also the action time and pause time of each event are given. Eva's writing process is quite fragmented. She writes in short clusters and segments. In this example two technical problems occur one after the other. The translated output of the dictated segments is shown below. The technical problems in the TPSF are underlined and the revision is printed in italics.

<p>(01) to prevent occupational hazard, (02) and at that co stant (07) to (11) constant (12) to evaluate and to improve. (13) these measures (14) are bundled (18) must (not literal: need) (21) in a global prevention plan (23) and a yearly action plan (24) must</p>
--

Eva dictates the first segment, pauses after the comma for about 6 seconds and dictates the second segment. She interacts constantly with the TPSF on the screen and with her papers at hand. She opts to correct the first error first. She switches from speech recognition to keyboard to navigate to the error and re-dictates the word. The second error 'co stant' (Dutch: voor dierend) is corrected in the same way. Although the error is actually at letter level, she opts to correct it at word level, by dictating the whole word again. And not, since she navigated to the error by keyboard, solve it by keyboard. A possible explanation for this strategy might be that she expects words to be added to the lexicon correctly if she tries to dictate them again (REv143). Eva stated that the recognition rate during longer writing sessions sometimes improves. She expected the same effect in this writing session. (REv143)

After correcting both errors Eva continues with a new segment, after a pause of 5.3 seconds. After completing this sentence she again pauses rather long: 7.7 seconds, before she continues with the final sentence of this paragraph. She starts this sentence but then realizes that the formulation 'these measures are bundled' is not fully adequate and she adds the verb 'must' in between. The correct formulation to her is 'these measures need to be bundled' (Dutch: deze maatregelen moeten worden gebundeld). She continues and finishes the sentence and rereads the TPSF on the screen for about 11 seconds. However, she does not correct the error 'must' (24). She glances in her papers and then positions her mouse in the next paragraph.

During the retrospective interview she commented on this fragment. We showed her the whole fragment.

¹⁵ Speech recognition based errors are often errors that result in phonological similar words. This relation is difficult to reproduce in English. Therefore the Dutch example should also be taken into account.

REv110	Researcher	Did you see both errors (*op/at, voor duren/con stant-2) immediately?
REv111	Eva	'hums' confirmatively
REv114	Researcher	What did you pay attention to? (*related to error moet/must-24)
REv115	Eva	Everything. I was mainly focusing on the whole paragraph. If the information was reflected in what I had written. I was checking if my summary was adequate.
REv119	Eva	I think I was looking eh is this what I want to say and not how it is written
REv120	Researcher	Why do you insert 'must' at this moment?
REv123	Eva	To formulate, I saw that the text said it was an obligation to write these plans, so that is content. This and that need to happen.
REv124	Researcher	You opt not to finish your sentence first. Why?
REv125	Eva	No, with me, if I decide this needs to be added, this needs to be added.

As mentioned before, Eva preferred correcting errors with speech recognition to increase efficiency in the course of the writing process. Another example of this strategy is the paraphrasing of 'full stop'. In Dutch 'punt' has at least two meanings: 'full stop' and 'point' in, for example, an enumeration. If this type of error occurs, Eva opts to paraphrase her text instead of correcting the error by keyboard.

Eva's writing process is characterized by a large number of repairs and she made most repairs in the first draft. In the beginning of the writing process she did pause longer than in the end. Her writing process had the highest fragmentation rate. We have divided the number of repairs by the number of words in the final text and Eva scored 13.31 repairs per 100 words. If we compare this to results reported by Van Waes & Schellens (2003) this is actually quite a low score. In their study the mean fragmentation rate was 15.32 (revisions without typing errors). However, in our study only half of the rate is related to actual revisions (41%). The other part has to do with technical problems caused by the speech recognition software. Eva also had the highest percentage of pausing time (67%) of the three participants in the case study. Eva indicated in the questionnaire that she takes little time to construct ideas about the text.

Once she has come up with the rough text, then she writes the text almost in one flow. She hardly ever pauses and changes little to the written text. This self-report is quite consistent with the way in which she organizes her writing process during this observation. She claimed to spend a bit more time on the revising subprocess than Ethan and Ben reported; this is also consistent with the observations in this writing session. She also had the highest mean pause length before repairs, namely, 5.41 seconds. On the other hand, her distribution was closely related to the general distribution of pausing and production time, respectively 68% versus 32%. She produced about 17 words per minute and ended up with 518 words in her final text.

In sum, Eva has a rather fragmented writing process because of her constant inter-

action with the TPSF and her preference to write a first time final draft. In the writing session she only delayed 4 technical errors. In other words, she prefers correcting each error immediately. On the other hand, the content of the text is her main concern. If this takes too much attention, an error might be overlooked.

4.5 Postpone (revision) profile: Ethan

Ethan started his writing session with the context of the report. He is the only writer that opted to start with the more knowledge telling part of the task. The context of the report could be quoted or paraphrased from the extra information which the participants received. During the interview he said: “I thought I would understand it all better, in relation to the content I mean. I thought I’ll write the two easy parts first.” (RE130) This enables him to think about the rest of his text. Characteristically for his writing session is that he switches to keyboard after each line to create a new sentence (space) or a new paragraph (enter). He does not use speech recognition for these actions. Figure 15 shows an excerpt of the first part of his writing process. It provides insight in his postpone profile.

#	Time (minutes)	WritingMode	Original output	Intended output	ActionTime (ms)	PauseTime (ms)
1	0:04:38	2	Left Button		110	1031
2	0:04:38	2	Movement		2562	94
3	0:04:39	TP 1 3 - dict	context in	context	1526	410
4	0:04:42	1	ENTER		78	1673
5	0:04:42	1	ENTER		63	328
6	0:04:43	3 - dict	de algemene verplichtingen van de werkgever met betrekking tot de bescherming van gezondheid en veiligheid van werknemers op de werkplaats	de algemene verplichtingen van de werkgever met betrekking tot de bescherming van gezondheid en veiligheid van werknemers op de werkplaats <u>kaderen in de ontwikkeling van een preventiebeleid</u>	8420	924
7	0:04:55	3 - dict	.punt		877	3129
8	0:05:01	1	BS		94	5666
9	0:05:02	1	SPACE		78	703
10	0:05:02	TP 3 3 - dict	kaderen in de ontwikkeling <u>er</u> van een preventiebeleid .punt	kaderen in de ontwikkeling <u>van</u> een preventiebeleid .punt	3561	390
11	0:05:09	3 - dict	de werkgever is verplicht te zorgen voor de veiligheid en gezondheid van werknemers		4978	3412
12	0:05:18	3 - dict	inzake alle met het werk verbonden aspecten .punt		3671	4226
13	0:05:26	TP 4 3 - dict	de werkgever <u>moeten</u> de nodige maatregelen treffen .punt	de werkgever <u>moet</u> de nodige maatregelen treffen .punt	3681	3762
14	0:05:32	TP 5 3 - dict	ook ,komma <u>trainen</u> en ter preventie van beroepsrisico 's'is'r	ook ,komma <u>maatregelen</u> ter preventie van beroepsrisico 's'is'r	3771	2106
15	0:05:39	1	LEFT (2)		1938	4098
16	0:05:42	1	DEL (12)		704	2015
17	0:05:44	3 - dict	maatregelen		1745	500
18	0:05:45	1	SPACE		78	0
19	0:05:47	1	END		47	2719

Figure 15. Fragment of writing process Ethan.

Ethan wrote lines 3 to 7 (#) almost in one flow. In the Dutch text errors appeared in segments 3, 6, 10, 13 and 14 (the underlined words in the translations). In the pause related to the full stop (3.1 seconds) Ethan is reading the TPSF and notices that the final segment does not appear on the screen. After a pause of 5.6 seconds he opts to switch writing modes to delete the full stop with his keyboard and subsequently types a space. He proceeds with speech and dictates the final segment once again. It is striking, though, that he does not correct the other smaller errors in the text, for example ‘context’ instead of ‘context in’. The continuation of the observation shows

that he prefers continuing with text production and finishing his paragraph. In the final sentence the second error that he corrected immediately appeared. Instead of 'measures' (Dutch: maatregelen), 'train' (Dutch: trainen) appears on the screen. After a pause of 4 seconds he chooses to correct the error by dictating the correct word a second time.

The English translations of the text that appeared on the screen are:

- (03) Context in
 (06) The general obligations of the employer related to the protection of the employees health at the working place x
 (10) fit in with the development of them the prevention policy.
 (11) The employer is obligated to take care for the safety and health of the employees
 (12) with regard to al work related aspects.
 (13) The employer needs to take measures.
 (14) also, train to prevent occupational hazards

So, Ethan opted to immediately correct the error in segments 6 and 14. The three other errors are delayed. He only detected the errors in segment 10 and 13 when rereading the printed text version. During the retrospective interview he commented on this fragment. We showed him the fragment until he starts error correction and asked him if he knew what he did during the writing session.

RE113	Researcher	Do you remember what you did at this moment?
RE114	Ethan	I think I start correcting errors by keyboard.
RE115	Researcher	What are you going to correct?
RE116	Ethan	Perhaps I left this error (*moeten/needs-10) in the and that one I have corrected (*trainen/train-14)
		show fragment
RE119	Researcher	Now you immediately point out the error 'moeten', did you see it at that moment too?
RE120	Ethan	No definitely not, it is that what I mean to say. In the beginning I was far more focused on the text ...

After Ethan has finished the draft of this first paragraph he revises a few formal elements. He inserts a general heading for the report and adds a paragraph heading to the context paragraph. Subsequently, he writes a second paragraph without rereading his first paragraph. In the interview he comments on this writing behavior by saying: "I never reread in between. I will always do that in the end, rereading" (RE128). Apparently Ethan's strategy consists of delaying various errors, but he is very precise in the errors that he will definitely correct immediately. Since he is very familiar with speech recognition he can predict certain types of errors and he can anticipate on them. For instance, the assignment mentioned several company names. Ethan stored the name he used most in his clipboard. Each time he used this name, he pasted it into his text by using the shortcut control + v. Besides this anticipation, he stated that utterances in which he expected technical problems, for example quotation marks,

a word with a capital letter and difficult words like ‘psycho-social’, were carefully monitored by him. “My expectations will trigger me to look at those instances.” Other errors that he did not predict easily escaped his attention. In his final text he found 8 errors that he had overlooked. Of these 8 errors, 5 were large errors like ‘structural’ (Dutch: structurele) appearing as ‘start to arrange’ (Dutch: start te regelen). The other errors are small errors and knowledge based (whether to write a word as a single word or not).

The writing process of Ethan is less fragmented than Eva’s. The fragmentation rate is 10.97 repairs per 100 words. The final text of Ethan contained 638 words and he produced almost 20 words per minute. He had the lowest percentage of pausing time (55%). In general, Ethan has twice as many pauses as Eva and Ben (almost 1500). If we focus only on pauses longer than 1500 milliseconds, this number is even higher, 30 for Eva and 60 for Ben. His pauses are of equal length over the various intervals. He seems to prefer building the text while writing.

4.6 Postpone (technical problems) profile: Ben

Before Ben started with his writing session he had a very long initial pause. Once started, he writes in very large clusters. He prefers continuing with text production and hardly interrupts his writing process to correct technical errors or to revise the text. Figure 16 is a typical example of an uninterrupted cluster, which does not contain any speech recognition based errors.

#	Time (minutes)	WritingMode	Original output	Intended output	ActionTime (ms)	PauseTime (ms)
1	0:28:03	1	ENTER		94	140
2	0:28:13	3 - dict	Hoofdletter Monti moet		1755	9895
3	0:28:16	3 - dict	ook		808	1240
4	0:28:18	3 - dict	veel aandacht besteden aan de psychosociale belasting		3082	1197
5	0:28:21	3 - dict	bij de afdeling .punt		1666	209
6	0:28:34	3 - dict	ook hier		957	11238
7	0:28:36	3 - dict	speelt een goede organisatie en tijdsindeling		4130	449
8	0:28:40	3 - dict	een grote rol .punt		1666	39
9	0:28:44	3 - dict	nu wordt de organisatie als niet echt adequaat ervaren door de medewerkers .punt		2204	2574
10	0:28:47	3 - dict	De toegenomen werkdruk en dus ook toegenomen stress		3591	800
11	0:29:26	3 - dict	toegenomen stress		4090	35767
12	0:29:31	3 - dict	zorgt de		1446	114
13	0:29:33	3 - dict	laatste jaren ook voor toegenomen spanning tussen de medewerkers onderling .punt		2364	561
14	0:29:37	3 - dict	en dat zorgt dan weer voor stresssituaties .punt		4310	2061
15	0:29:47	3 - dict	De afdeling zit dus een beetje in een vicieuze cirkel .punt		2454	5852
16	0:29:57	3 - dict	het is belangrijk voor de medewerkers en ook voor het bedrijf dat die cirkel wordt doorbroken .punt		1905	7786
17	0:30:00	3 - dict			4170	499
18	0:30:06	3 - dict			6574	2413
19	0:30:15	1	ENTER		63	1806
20	0:30:15	1	ENTER		93	157

Figure 16. Fragment of writing process Ben.

Furthermore, the fragment in Figure 17 shows him struggling with the writing medium. It shows one of the instances when he does opt to correct an error immediately and it also shows how his error correction strategy is influenced by the speech recognition software.

#	Time	WritingMode	Original output	Intended output	ActionTime	PauseTime
	(minutes)				(ms)	(ms)
21	0:30:22	3 - dict	het		788	7224
22	0:30:23	3 - dict	tennis instituut \Hoofdletter	Kennisinstituut	2105	36
23	0:30:25	3 - dict	present	Prevent	1107	371
24	0:30:30	1 LEFT			94	3275
25	0:30:30	1 LEFT (2)			109	312
26	0:30:31	1 RIGHT			109	438
27	0:30:31	1 RIGHT (2)			157	312
28	0:30:31	1 RIGHT			110	125
29	0:30:33	1 BS (6)			469	2078
30	0:30:34	3 - dict	kennis		1207	28
31	0:30:35	3 - dict	-streepje-links die ik	streepje instituut	1825	394
32	0:30:37	3 - dict	\Hoofdletter present	Prevent	1775	172
33	0:30:42	1 BS			1328	3365
34	0:30:44	3 - dict	het		798	2049
35	0:30:47	3 - dict	bureau \Hoofdletter present R 1	Prevent	2574	2004
36	0:30:52	1 LEFT			62	2247
37	0:30:52	1 LEFT (3)			219	390
38	0:30:53	1 DEL			78	282
39	0:30:53	1 v			32	718
40	0:30:54	1 RIGHT			79	453
41	0:30:54	1 RIGHT (2)			203	438
42	0:30:55	1 SPACE			78	500
43	0:30:55	3 - dict	heeft naam en faam		2214	764
44	0:30:58	3 - dict	waar het aankomt op de preventie van beroepsrisico was \punt		4709	281
45	0:31:04	3 - dict	het bureau		1137	1652
46	0:31:08	3 - dict	heeft heel wat knowhow verworven		2444	2057
47	0:31:14	3 - dict	wat het bevorderen		2324	3909

Figure 17. Second fragment of writing process Ben.

Ben pays a lot of attention to this error correction. As stated before, this is not characteristic of his profile. In general he delays error correction and uses his keyboard to perform these actions. However, the way he deals with the writing medium is characteristic. We zoom in on at his error correction strategy to show this. The English translation of the text that appeared on the screen is:

- | |
|--|
| <p>(01) The
 (02) tennis institute \Capital
 (03) present
 (10) knowledge
 (11) hyphen – left that I
 (12) \capital present
 (14) the
 (15) bureau \capital present
 (23) has a good report
 (24) in the area of preventing occupational hazards
 (25) the bureau
 (26) has gained a lot of know how</p> |
|--|

This fragment is taken from the beginning of the second page of his report and until then he has interrupted his text production only 4 times to make a revision or to solve

a technical problem. His goal is to dictate ‘the knowledge instituut Prevent’ (Dutch: *Het kennisinstituut Prevent*). Since Prevent is a company name, this causes some problems and requires some (speaker-specific) training (cf. Figure 4)¹⁶. If a word is not available in the lexicon of the speech recognizer, it will choose a word that is closely related. Ben opts to train the software by simply repeating the dictated segment. In Dutch ‘knowledge institute’ is written in one word. To avoid the same error again, he opts to split the word in two parts by using a hyphen. This strategy is not successful. The solution Ben uses to continue writing with speech recognition is to change his phrasing. Instead of using ‘knowledge institute’ he revises it by paraphrasing it with a less specific synonym ‘bureau’. In this case the technical problem leads to a revision. The company name is still misrecognized. After three trials with speech, he opts to correct the word by using his keyboard. It took Ben about half a minute to solve this problem. In the next few lines he combines the terminology of ‘bureau’ and ‘institute’ to refer to the company. Four lines later he uses the same strategy when an error occurs in the text. Instead of solving the error at word level, he rephrases the whole sentence.

In the retrospective interview Ben indicates that he solved errors if they met two criteria. The first criterion is that he expected misrecognition of the dictated text because he made a slip of the tongue. The second criterion is that the error caused a large content error. In the excerpt below he explains his error correction strategy for the shown fragment.

RB138	Researcher	Can you describe what happened?
RB139	Ben	Apparently If I trip over a word I correct it immediately and otherwise I don't
RB147	Researcher	Why do you opt to correct the error with speech recognition?
RB150	Ben	This is the tool we need to rely on so...
RB152	Ben	Normally I would train the computer beforehand, before I start to write a text.
[RB157-167]	Researcher	[Discussion about the influence of the recognition on the text content (*paraphrasing of sentence after technical error)]
RB168	Researcher	Would you have preserved the sentence if it had appeared correctly on the screen?
RB169	Ben	Definitely.

Ben mentioned various times that the speech recognizer influences his writing process. He can predict what will be difficult for the speech recognizer and he tries to avoid errors in the text. Therefore, he uses different words and different constructions. For example, in Dutch past particles are very difficult. If possible Ben opts to use synonyms. Furthermore, Ben indicated that in his opinion it is more efficient to continue with text production and correct the text afterwards. He assumed that otherwise he

¹⁶ Besides training on the spot, writers can also add various texts that they have written beforehand. The software ‘learns’ the words which the writers use frequently.

would forget the gist of his text. Moreover, he feels comfortable enough with his first draft for him to rewrite the text in a second draft even if the TPSF is not flawless. Ben spent 4'52" minutes in the second draft and he corrected 22 technical problems. From time to time he did not accurately remember what he had tried to formulate in his first draft, but in those instances he used the TPSF as a basis to complete his text. In his final text only one technical error was left unresolved.

In the questionnaire writers self-reported on their pausing and planning behavior. We asked them to indicate how they would describe their writing process when they were asked to write a letter to an organization in which they request their cooperation on a project. Ben indicated that he would perform this task as follows: "I start almost immediately with text production. I take time to think about the content of the text. In particular, in the beginning of the text I pause before I produce new text". Without any doubt, this profile holds for Ben in this situation. Ben has the lowest number of pauses of the three writers described in this case study. However, his pauses are quite long, especially in the beginning of the writing process. Pauses before repairs, on the other hand, are the shortest, 2.03 seconds on average. Ben is the most fluent writer of the three that we have described in detail. He produced almost 26 words per minute and his final text was 731 words long.

5 Conclusions and discussion

In this study we described cognitive processes during text production, and our main focus was on error correction strategies. To study this theme we have analyzed the writing processes of 10 professional speech recognition users while writing a business report. We were interested in the questions: Do writers prefer continuing with text production even if the TPSF is inaccurate because of errors in the text? Or, do they prefer interrupting their writing process frequently to keep the visual representation of the TPSF as correct as possible. The correct answer to these questions is 'both'. If a writer detects a technical problem in the TPSF that occurs close to the end of the utterance (within the same sentence) he will probably prefer correcting it immediately. Especially if that problem occurs in the first draft and is a low-level error. On the other hand, text problems that lead to content related revisions have a higher chance of being delayed. Quite often these revisions will be delayed till the second draft and lead to additions of new text to the TPSF beyond paragraph boundaries.

To explain the strategies writers prefer, we have described three types of error correction strategies: handle, postpone revisions and postpone technical problems. The first strategy is the most frequently used. Five out of ten participants preferred correcting errors in the text immediately. Three of these writers are described in the handle profile. The handle group mainly corrects errors that were characterized by a high number of repairs in the writing session. Both technical errors and revisions are solved immediately. Repairs are immediately performed by switching from production to repair and also in small episodes (within the sentence or segment). Immediate stands for both at the point of utterance as well as in the first draft. In other words,

texts are 'sculptured' at the point of inscription. The second group, the 'postpone (r)' group, prefers delaying revisions. The general number of repairs is quite high, more specifically the number of technical problems (64%) in relation to revisions (36%). Technical errors are likely to be corrected immediately, switching from production to repair in the first draft. Revisions are often delayed and corrected in repair episodes. The third group, the postpone (tp) group, who preferred delaying technical problems, has in general a low number of repairs and more specifically a low number of revisions. They prefer correcting errors immediately, one at a time, switching from text production to repair and then continuing with text production again. So, they correct only a limited number of errors, and other errors are (deliberately) kept in the text to be corrected later. Delayed errors in the first draft are mostly revisions and delayed errors in the second draft are mostly technical problems.

5.1 Writing profiles

The variety between writers has been the subject of many studies. Process data and writer characteristics can be very informative to distinguish writing profiles. Lindgren (2005), for instance, mentions three approaches in her study on analyzing on-line revisions. We would like to relate our study to this description. Lindgren states that:

The writing process is not a static process; it varies both between and within writers. The way in which text is planned is one example of a process that is undertaken differently depending on the writer, the text type and the writing mode. (Lindgren, 2005, p. 82)

In this study we focused on business texts that were written with speech recognition which created a high cognitive effort even though the participants were expert speech recognition users. We are aware that we should be cautious in generalizing to other tasks and writing modes. We tried to provide more insight in error correction strategies as writers need to fulfill a demanding task, especially in the knowledge transforming episodes. Although the writers are expert speech recognition users they still had to deal with the inherent difficulties related to the writing mode. The between-writer variation in this situation is therefore very helpful to gain insight in how writers juggle with constraints during writing processes.

Lindgren associates the typologies of Galbraith (1999), Kieft, Rijlaarsdam, Galbraith & Van den Bergh (to appear) and Van Waes and Schellens (2003) as factors of influence on the study of on-line revisions: writing profiles influence the organization of the writing process and the treatment of revisions. We now briefly describe the writing typologies and then relate our data to them. Galbraith distinguishes profiles based on personality characteristics. Writers can either be high or low self-monitors: high self-monitors are mainly concerned with the situational appropriateness of their selves in social presentations. Low self-monitors are much less concerned with this social appropriateness; they are more driven by internal dispositions (Galbraith, 1996, 1999). In a writing study and a self-report study, Galbraith mainly focuses on drafting strategies and planning behavior. To measure self-monitoring, the self-monitoring scale of Snyder (1974) was used. High self-monitors appeared to have a higher

degree of idea generation before text production. Low self-monitors have fewer ideas prior to text generation, but they carefully construct the text while writing.

Kieft, Rijlaarsdam, Galbraith & Van den Bergh (in preparation) have built upon these ideas in a study about student characteristics in which the students wrote argumentative texts. Kieft et al. also used a self-reporting questionnaire. The main focus was on planning and revising strategies. They developed a writing course to meet the needs of various writers. They expected that high self-monitors would benefit most from planning strategies (Kellogg, 1994, 1996) and low self-monitors from revision strategies (Galbraith & Torrance, 2004). However, in contrast with what they expected, writers benefited from instruction that complement their initial profile. In addition, the questionnaires on writing strategy and self-monitoring did not show any significant correlation. Because of this, Kieft et al. are hesitant to confirm a single direct relation between self-monitoring and the writing process. However, in our opinion these findings are complementary and not restrictive, certainly since in general the writing skills of the writers did not improve. Only low self-monitors did as in Galbraith, Torrance, & Hallam (in preparation) benefit from a complementary instruction. Kieft et al. state that the planning behavior of writers does not account for the strategy they employ in their complete writing process.

Therefore, we will also further explore typologies as described by Van Waes and Schellens (2003) who have constructed typologies on the basis of a wide spectrum of variables describing different subprocesses and characteristics of the complete writing process. They created five writing profiles that distinguished writing strategies in pen-and-paper and computer-based writing modes. Initial planners pause a lot and revise little. Average writers are average related to other profiles. Fragmentary stage I writers revise more than others in the first draft. Stage II writers made most of the revisions in the second draft. Non-stop writers revise less often than other writers. We will relate the participants writing behavior to the high and low self-monitoring typology by Galbraith, and the writing profiles of Van Waes et al.

Writers with the handle profile can be related to the low self-monitoring profile described by Galbraith (1999). They carefully construct their text while writing. Consequently, they have a lot of characteristics in common with the Fragmentary Stage I writers described by Van Waes & Schellens (2003). Eva's writing process, for instance, is characterized by a large number of repairs and she made most repairs in the first draft. Her writing process had the highest fragmentation rate.

Writers with the 'postpone technical problems' profile can be related to high self-monitors. Their writing process is described by idea generation prior to text production. Their typology can be constructed from the Stage II writers and the Non-stop writers. Stage II writers make most of their revisions (technical problems in this case) in the second draft. They spend some time on planning, but once they start writing, they have quite a fluent writing process. Ben, as an example of this profile, had a long initial pause before he started text generation. Once started, his writing process was very fluent, almost without any fragmentation.

Writers with the postpone-profile who prefer delaying revisions can be placed on a continuum between high and low self-monitors. Their profile cannot adequately be related to any of the writing profiles described by Van Waes & Schellens. Therefore, we would like to elaborate on this profile based on the theoretical distinction that we have made in the introduction. We stated that the handle profile might be divided into two subprofiles. This profile actually meets the criteria for the writers who also make a thorough diagnosis of the problem and prefer 'saying it differently'. However, according to our categorization, they prefer delaying (or postponing) this instead of performing the repair immediately. Apparently, the contrast between immediate and delayed error correction is quite decisive for the way in which writers structure their writing processes. Besides, the distinction between technical problems and revisions plays an important role, although most writing typologies ignore this type of correction. Most writers prefer solving technical problems immediately. The same does not necessarily hold for revisions. However, the strategy to postpone errors is not equivalent to postponing revisions.

Apparently, it is difficult to directly relate the writing profiles in our study to the various existing profiles (as Kieft et al. (in preparation) also described). The approach of each writer profile is quite unique. Galbraith mainly focuses on the planning component in the writing process, Kieft et al. focus on planning and revisions and Van Waes and Schellens take - besides planning and revising - various other process and product characteristics into account. In this study we have focused attention on another aspect of the writing process: immediate and delayed error correction. Comparable to the detailed focus of Galbraith on self-monitoring as an explanatory characteristic, this study focuses on another aspect, the moment of error correction as a clarifying variable for writing profiles. The moment when writers prefer correcting errors reveal distinct patterns that seem to affect other writing process characteristics.

On the one hand this diversity in approaches produces insight in the most decisive characteristics of writing profiles. On the other hand it makes it more difficult to generalize across various writing processes, writing tasks, etc. In future studies on writing processes it would be appropriate to study the most decisive characteristics in various settings in order to obtain a more general profile description. Besides to the further refinements of the existing profiles, a combination of profiles could reinforce the description of writing profiles. We feel that a general view on profiles could be obtained via a combination of writer characteristics (as in self-monitoring), process characteristics (fragmentation of writing process in general and those caused by error correction strategies - both technical and revisions - and fluency) and product characteristics (number of words in final text).

In this study we have described the writing process from various perspectives. The converging research methods that we have used (product analyses, process analyses and protocol analyses) enabled us to elaborate on writing process data in great detail. Writing strategies and writing typologies are extended because of this approach. The error correction strategies that emerged from this study clearly show that writ-

ers use various successful strategies to deal with errors in the TPSF. The strategies are described in detail and also the reasoning behind the strategies are explained. We are convinced that the observational methods and the research approach in this study can be very useful in future writing process research: especially to describe writing profiles.

5.2 Comparison with previous studies

In the introduction to this chapter we described the distinction between the handle and postpone profiles at various levels. In general, we see that writers employ various strategies to deal with errors in the TPSF. As Figure 3 shows, writers are forced to choose between content revisions and more technical problems. On the one hand writers prefer revising the content of the text, on the other hand writers are more focused on the form of the text. It also shows the pattern that solving a technical problem might lead to a content revision (and vice versa). Writers clearly start from the two questions presented in the model: 'is the form or the content of the TPSF correct?' In the retrospective protocols, writers made regular remarks like: 'I expected that an error would occur' and 'I was not sure if that was what I really wanted to say', indicating the conscious decisions of interrupting the writing process. If we compare the error correction strategies of novices (Leijten & Van Waes, 2006b, also chapters 2, 3 & 4) versus expert speech recognition users, we see quite a consistent pattern. The total number of repairs per minute is a bit lower for the expert writers. In general the expert writers conducted 1.71 repairs per minute. The novice speech recognition users had to correct about one error more per minute ($M = 2.90$). The number of revisions remains the same for both groups and has about the same variation. Consequently, the distribution between technical problems and revisions is different. Novice speech recognition users spent 20% of their error correction on content revisions, while expert speech recognition users spent 40% on content revisions¹⁷. One explanation could be that when the subprocess of technical error correction is too demanding, writers do not have enough capacity to perform revisions. In terms of Figure 1, the interaction between the physical realization of the TPSF and the mental representation on the TPSF is too demanding to still have cognitive energy left for the other two levels (discourse model and situation model).

This study confirms the findings of Leijten, Ransdell & Van Waes (submitted, chapter 5) and Leijten, De Maeyer, Ransdell & Van Waes (in preparation, chapter 6) in which various error types were presented in a single sentence production task. In addition, we found that in an ecologically more valid setting (speech recognition) errors are also more often corrected immediately. Large errors may cause the writers to lose the overview over the various levels of their representation (cf. Introduction,

¹⁷ In this study we coded 2538 repairs in the four final observation sessions in which the participants (with classical dictating experience and without classical dictating experience) either solved a technical problem in their text or made a revision (+ dictating experience: technical errors $M = 2.11$ per minute, revisions $M = 0.79$; -dictating experience: $M = 2.38$ technical problems per minute and $M = 0.40$ revisions per minute).

Figure 3). We assume it is important that the discrepancy between the TPSF and the discourse model is not too large for writers to be able to map the different levels of their cognitive representation of the text.

In the latter study we also described the profile of writer groups based on their preference to immediately correct errors or to postpone this correction. The profiles can be transferred to this study. The main difference is that in the previous study four profiles could be distinguished. The handle and two postpone profiles can be discerned also in this study. However, in the experimental setting a group of writers was formed that delayed almost every error correction (99%). Apparently, this is not realistic in a more real-life situation with a complex task.

Moreover, writers confirm that - based on their experience - they often 'feel', or suspect that errors will occur. In these instances errors are preferably solved immediately; unless the text that needs to be generated is too demanding. It is obvious that writers are confronted with a dilemma during their writing process. So far, we have focused mainly on the role of the TPSF and we see that writers deliberately choose between immediate error correction if they feel the error is too disturbing to continue text production and delay error correction if they assume they will lose the gist of the text. Not only is the complexity of the TPSF of importance but also the complexity of the text to produce. Therefore, in a follow-up study we have varied the complexity of the text that needed to be produced (Quinlan, Loncke, Leijten, & Van Waes, in preparation).

Leijten, Ransdell & Van Waes (submitted) also stated that texts of which the TPSF is presented visually and in the auditory mode are preferably completed first and the error correction is postponed. However, in 'real' text production this pattern is more diverse. In the experimental condition it is easier for writers to maintain the overview of the representation (we could perhaps even state that is negligible). The representation of a single sentence in an experiment is less complex compared to the fairly complex representation of a real business text. In the experiment the 'writers' had little planning to do. In this writing task the writers have to deal with all the aspects of the situation model and translate that to written text. The participants - in both studies - have different coping strategies in this respect. Some prefer immediate error correction. This way they make sure that they are able to make a correct representation of the TPSF at each single moment of their writing process (bottom-up). On the other hand writers prefer fluent text production to keep the gist of the text (top-down).

5.3 Methodological considerations

In this final section we discuss the methodological considerations made in this study and choices made within the various research projects on writing and speech recognition. As stated before, we have opted to use converging research methods in this study. The methods used are very complementary. Each element of the product, process and protocol analyses added up in this study. The writing processes were fully logged: keystrokes, mouse movements, speech input, and time-related process data.

The logging did not interfere with the writing processes. During the writing processes the writers were also observed; not only the text that appeared on the screen, but the writers themselves too. On top of that, after finishing the writing task, the writers were asked to reflect on their writing process in a stimulated retrospective interview. More specifically, they were stimulated to recall why they employed various error correction strategies.

The process data were analyzed on three levels: overall level (aggregated data), subgroup level (aggregated data on three writer groups), and individual level (highly detailed case analysis). Both the methods used as well as the analyses performed showed to be converging and enabled us to create multiple perspective on the description of the writing process.

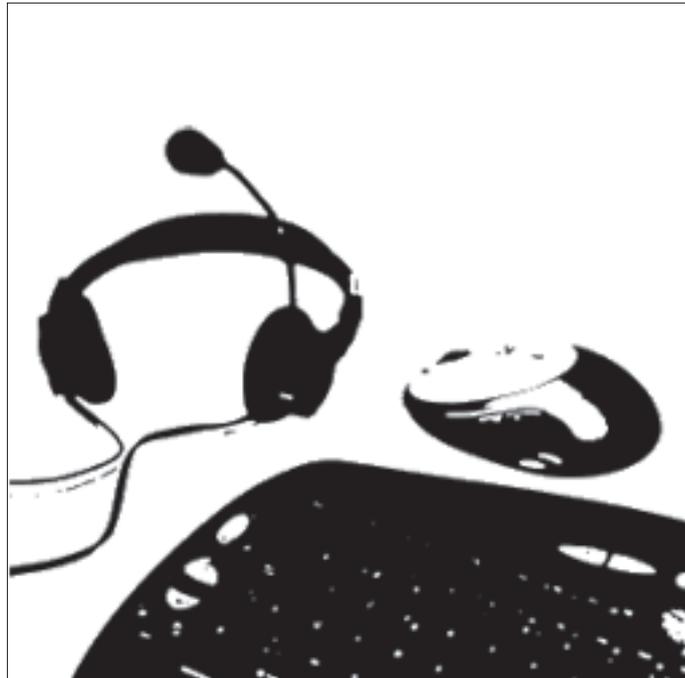
The aim of this book was to describe the relation between writing and (influences of) speech recognition. To explore this relation, we have used complementary research methods. The quasi-experimental design in the last study allowed us to compare the writing processes of the participants related to a specific task. This is different from the first project that was more ecologically valid. In that study writers performed various writing tasks in their own professional writing environments (chapters 2, 3 and 4). In the present study, we also took the complexity of the writing task in consideration. This is quite different from the experimental study in which error types were compared in a more restrictive environment (i.c. isolated sentence; chapters 5 & 6). In isolated contexts very specific low-level information on error types and working memory can be gathered, but of course the complexity of the writing task also has an impact on the organization of writing processes. The complexities of writing tasks provide information about characteristics of the writing process that are of importance. In isolation, however, the exact influence of that characteristic can be measured in more detail. Therefore, we would certainly like to encourage further research in which quasi-experiments are complemented with – or related to – more controlled experiments, and more explorative, ethnographic studies.

Acknowledgements

We would like to thank the staff members of the VRT for their cooperation in the project. We are particularly grateful to Bernard Dewulf and Erik Desnerck for their enthusiasm and critical attitude towards speech recognition as a writing medium. In addition, we would like to thank Hans Geuns and Stijn Van Even of Nuance for their professional and warm-hearted cooperation. Mathia Van De Poel did a great job in programming the integration between Inputlog 2.0 Beta and Dragon Naturally Speaking 8.1. Finally, we would like to thank Geert Jacobs for proofreading this chapter.

Section IV

Logging writing processes in a Windows environment



8

Inputlog

A research tool for observing and analyzing multimodal writing processes in a Windows environment

Abstract: Not only has the use of computers as writing instruments had a profound effect on the writing practice and the attitudes towards writing, it has also created new possibilities for research on writing. In the field of cognitive writing research especially, keystroke logging programs have become very popular. In this chapter we describe a logging program called Inputlog. Inputlog 2.0 Beta 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; (4) a playback module that enables researchers to review the writing session. In this chapter we describe the technical and functional characteristics of Inputlog 2.0 Beta and give advice on applying Inputlog as a research tool. We conclude the paper with a preview of the plans for further developments.

Keywords: Inputlog, keystroke logging, registration tool, on-line writing processes, pauses, writing modes, text analysis, pause analysis, research methods, writing observation, cognitive processes.

This chapter has been elaborated from Leijten & Van Waes (2006). Inputlog: New perspectives on the logging of on-line writing processes in a Windows environment. In K. Sullivan & E. Lindgren (Vol. Eds.) & G. Rijlaarsdam (Series Ed.), Studies in Writing: Vol. 18. Computer Key-Stroke logging and Writing. Methods and Applications (pp. 73-94). Oxford: Elsevier; Van Waes, L., & Leijten, M. (2006). Logging writing processes with Inputlog. In L. Van Waes, M. Leijten, & C. Neuwirth (Vol. Eds.), & G. Rijlaarsdam (Series Ed.), Studies in Writing: Vol. 17. Writing and Digital Media (pp. 158-166). Oxford: Elsevier; Van Waes, L., & Leijten, M. (2006). Schrijfprocessen registreren met Inputlog. Een data-analyse van de interactie met de 'reeds geproduceerde tekst' [Observing Writing Processes with Inputlog: A Data Analysis of the Interaction with the Text Produced so Far]. Tijdschrift voor Taalbeheersing, 28(2), 198-219.

1 Introduction

As most writers nowadays produce nearly all of their texts on a word processor, the computer has not only become the major writing instrument, its use as a research tool has also increased. The computer enables researchers to collect detailed information about the writing process that were hardly accessible before. Or, as Spelman, Miller and Sullivan (2006) state:

As an observational tool, keystroke logging offers the opportunity to capture details of the activity of writing, not only for the purposes of the linguistic, textual and cognitive study of writing, but also for the broader applications concerning the development of language learning, literacy, and language pedagogy. (p. 1)

In this day and age, researchers make frequent use of keystroke logging tools to describe online writing processes in detail. These logging programs enable researchers to exactly register and accurately reconstruct the writing processes of writers who compose texts at the computer. The basic concept of the different logging tools that have been developed is more or less comparable. First, the keystroke logging tools register all keystrokes and mouse movements. During the writing process these basic data are stored for later processing. This continuous data storage does not interfere with the normal use of the computer, creating an ecologically valid research context. At a later stage, the logged data can be made available for further analysis, either within the program environment itself or as exported data in statistical programs such as SPSS or SAS. Depending on the research question, researchers can choose to analyze different aspects of the writing process and the writing behavior by combining, for instance, temporal data (e.g. time stamps or pauses) with process data (keystrokes or mouse movements). Computer-based data collection (and processing) is much faster and more accurate than manual data collection.

Depending on the research question, researchers can study different aspects of the writing process or the writing behavior by combining temporal data (e.g. absolute time and pauses) and writing process data (keystrokes and mouse movements) for analyses. In this chapter we describe Inputlog, a logging tool for writing process research developed for Windows environments. First, we present the most important characteristics of Inputlog 2.0 Beta. Then we give a more detailed description of the technical and functional characteristics of Inputlog. Next, we elaborate on the applications of Inputlog. To conclude we preview the further developments of the program.

The chapter describes Inputlog in great detail. Readers who are mainly interested in the general characteristics of Inputlog should read section 2, 4 and 5. Readers who are interested in the technical aspects of Inputlog should also read section 3. Readers who already are familiar with Inputlog, and are interested in specific techniques of data analyses find more information on the topic in section 6.

2 Characteristics of Inputlog

For the development of Inputlog we were able to fall back on the functionality of two existing programs: JEdit and Trace-it on the one hand (Kollberg, 1998; Severinson Eklundh, 1994; Severinson Eklundh & Kollberg, 1992, 1996, 2003; Spelman Miller & Sullivan, 2006), and ScriptLog on the other hand (Strömquist & Karlsson, 2001; Strömquist, Holmqvist, Johansson, Karlsson, & Wengelin, 2006). Both have serious limitations. JEdit and Trace-it are designed for Macintosh personal 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). Trace-it features an extended interactive revision module, while ScriptLog is the first program that combines logging data with recorded eye-tracking data.

Most logging-tools are either developed for a specific computing environment, or not adequately adapted to the current Windows environment¹. As such, they cannot be used for writing studies in which 'natural' writing and computer networks employ commercial word processors (e.g. Microsoft Word or WordPerfect). This discrepancy was one of the main reasons for deciding to start with the development of Inputlog in 2003.

Another impetus for the development of Inputlog has been the emergence of speech recognition as a new writing mode for word processors. In our first research study on the influence of speech recognition on the writing process we analyzed the writing process data manually (Leijten & Van Waes, 2003b; 2005b, see also chapters 2, 3 & 4). At that moment it was impossible to register keyboard input in combination with speech mode data with any of the existing logging tools. To collect the writing process data, we combined two digital observation instruments: a digital screen cam (i.c. Camtasia)² and a digital sound recorder (i.c. Quickrecord)³. The study showed that although the chosen observation instruments and analyzing methods did enable us to analyze and describe the specific speech recognition writing processes, the data analysis was very time-consuming. In follow-up experiments we registered and analyzed these multi-modal data with Inputlog (Leijten, Janssen, & Van Waes, in preparation; Leijten, Ransdell, & Van Waes, submitted; also chapters 5, 6 & 7).

In sum, Inputlog allows researchers to:

- record (keyboard, mouse and speech) data of a writing session in Microsoft Word, I-pad and other Windows based programs (cf. section 4.2);
- generate data files for statistical, text, pause and mode analyses (cf. section 4.3);
- integrate various types of data from other programs (cf. section 4.4);

¹ In June 2006 the company Noldus Information Technology released a keystroke logging program called Ulog that also registers in a Windows environment. The tool is an addition to The Observer software (www.noldus.com).

² Camtasia was renamed to Camtasia Studio. It is a commercial software package for recording, editing and sharing high-quality screen videos (more information: <http://www.techsmith.com>)

³ Quickrecord is a compact application for recording and playing sound on a Microsoft Windows PC (more information: <http://www.ptpart.co.uk/>)

- playback the recorded session at different speeds (cf. section 4.5).

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.

2.1 Word processor independency

Contrary to Trace-it and ScriptLog, Inputlog registers every keystroke and mouse movement independently of the word processor used. Inputlog is designed to log and analyze writing data produced in Microsoft Word. However, the program also logs keyboard and mouse actions in other Windows based programs⁴. In other words, not only writing processes as such can be observed with Inputlog, but also basic processes like consulting websites or programming in any Windows based language.

2.2 Parsing

Generally speaking, parsing refers to the process of analyzing an input sequence⁵. 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 (e.g. statistical, text, pause, and mode analyses) from the source logging file. The processing of input data depends on what has preceded, and even on what follows. As such, keeping track of these dependencies in order to verify how to process the data is called parsing. In order to facilitate the implementation of new functionalities and to better control program maintenance, we have opted to technically restructure version 2.0 of Inputlog by using a specific parsing technique: syntax directed parsing. Inputlog uses the tools Flex and Bison to implement the parsing (Aho, Sethi, & Ullman, 1986; Donnelly & Stallman, 1995; Levine, Mason, & Brown, 1992; Paxson, 1995). These parsing tools generate parsers in C++ code which are then integrated in the Inputlog program code.

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, one does not only get a more readable program structure, but one also obtains a framework in which it is possible to get the different analyses in just one or two passes. This makes the program considerably faster.

In *syntax directed parsing*, a standard algorithm automatically constructs the input part of a program from a high level description of the input data structure. Code to perform any required processing of the data is then attached to the description in a convenient way. This non-procedural description is generally easier to write and to

⁴ Researchers should keep in mind that for certain analyses (e.g. analyses on sentence or paragraph level) only the data logged in Microsoft Word can be used. For this kind of analyses the basic data are interpreted with a set of algorithms based on Microsoft Word.

⁵ We would like to thank Nico Verlinden for his effort in integrating Parsing in Inputlog.

modify than the equivalent program code, and much less likely to harbor bugs. It is also easier to read, and easier to maintain.

2.3 XML structure of output files

In the previous versions of Inputlog the output data were generated as Excel files. However, we have chosen to integrate the more universal XML structure in the current version of Inputlog. 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.

```
<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>
```

The advantage of XML is that researchers can easily adapt the data to their own research needs. Research data can be built in the same way as the Inputlog output and can then be easily integrated into one another. The example above shows how the XML structure of Inputlog is built. Each <tag> states the information between the tags: for example writing mode 2 is a mouse movement or click. In the future we are hoping to develop a standard for keystroke logging tools and exchange data XML structures that are compatible with other keystroke logging programs⁶. This would enable merging between XML structured data of various programs. We will give another example of these possibilities in more detail in the next section.

2.4 Speech recognition

As mentioned above, Inputlog is currently the only logging tool that can integrate the input of dictation devices using speech recognition software. Since the analyses of multi-modal speech recognition data turned out to be extremely laborious, computer assisted logging is very helpful in recording and analyzing writing sessions in which speech technology is used. For this reason we chose the most widely used speech recognition software, that is, Dragon Naturally Speaking. The specially designed logging add-on in the speech recognizer (in combination with a Python script) enabled us to integrate the dictated text with the data logged by Inputlog. Comparable to the general logging file generated by Inputlog, the logging file of Naturally Speaking relies on timestamps. Via a Python script that was developed for this purpose, we generated

⁶ Data collected with ScriptLog then becomes exchangeable with Inputlog.

an XML file of a recorded speech session that is basically structured in the same way as an Inputlog data file. By also converting the output files of Inputlog 2.0 Beta into XML, we were able to combine both logging files and integrate them into one general logging file based on the convergence of timestamps. The result is a single file that can be used for further analysis of multimodal writing sessions in which speech input is combined with keyboard & mouse.

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

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 1. Example of a writing session with keyboard, mouse and speech recognition.

In this excerpt the writer dictates the next segments following each other closely (translated from Dutch):

04.39	<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 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 key presses, 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). Next to this, 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. The implementation of speech recognition in Inputlog will stimulate research on the effect of this new technology on the writing process (of both professional writers 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 (see section 5.3 further applications).

We have developed an interface in which researchers can select the analyses they need for a specific source file. This user-friendly interface allows researchers to adapt Inputlog to their research needs. This basic functionality is described in section 4. In section 3 we first give some more information on the technical background of Inputlog.

3 Technical description

In this section we describe Inputlog from a more technical perspective. Technical terminology will be further explained in the glossary (see Appendix 1). First, we illustrate the flow of the program. Then we describe the program structure. Finally we formulate several ‘best practices’.

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, namely:

- 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 (Inputlog Data Files) that are converted to different output files afterwards, preparing the rough data for qualitative and quantitative analyses (cf. functional description in section 4). Figure 2 visualizes the flow of the most recent version of the program: Inputlog 2.0 Beta.

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 either as input for the analyses, or as input for the playback module (cf. *infra*). In step 3 the binary data of the source file are converted to text data.

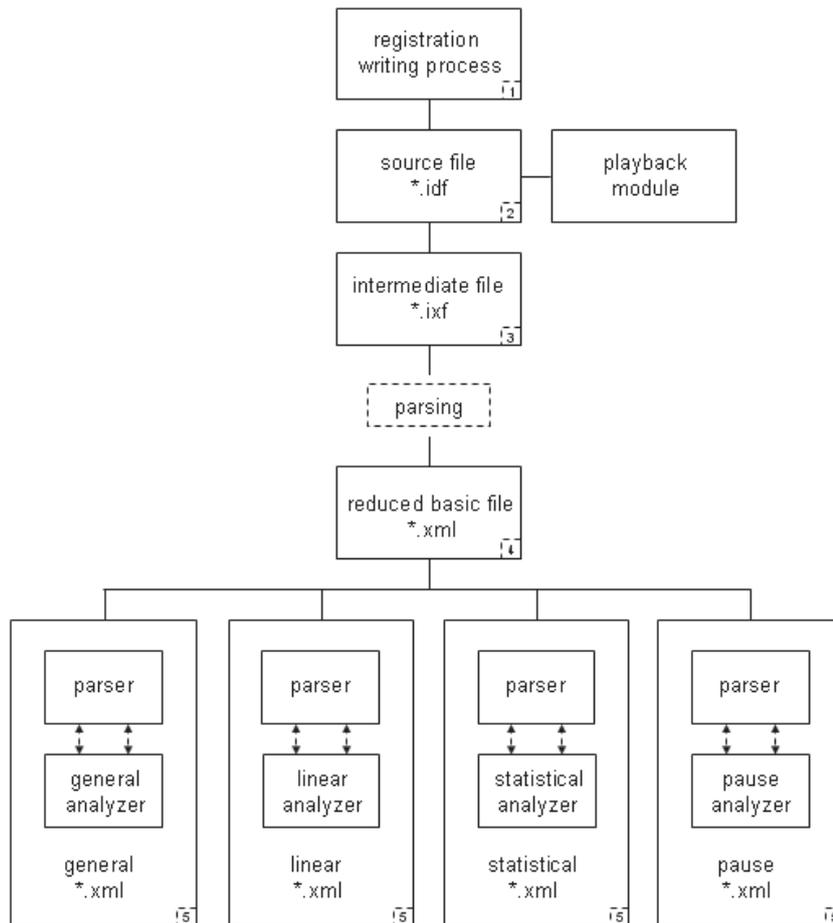


Figure 2. Basic flow of the logging and analysis conversion in Inputlog (without merging of speech).

This conversion is necessary because of the syntax directed parsing technique that we use (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 that enables researchers to analyze 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) is also used as input for the play back module⁷. 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.

3.1 Programming language

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.

It is very important that the resident logging program does not interfere with the use of the word processor, and consequently, that it does not hinder the writer during his or her writing process in any way. Therefore logging and analyzing have been separated. Each part of Inputlog is programmed in different classes (see Figure 3). This allows us to easily update and debug the current analyses, and to further extend the functionality of the program. Inputlog runs on a PC with only one CPU and it's use of system resources is quite limited, contrary to, for example, Java.

For the logging of the writing process Inputlog uses two Windows Hooks: Journalrecord and Journalplayback. Journalrecord registers every keystroke and mouse operation (movement and click) together with the corresponding time stamps. The Journalplayback supports the playback module. For the parsing of the analyses we used the so-called Flex (Paxson, 1995) and Bison (Donnelly & Stallman, 1995) tools. Merging with Dragon Naturally Speaking was conducted with a Python script ("Python", 2007).

3.2 Structure of Inputlog

Figure 3 shows the structure of Inputlog, the so-called 'system topology'. Inputlog consists of four subsystems: central, playback, analysis and record. The central system consists of the classes session manager and data manager. The session manager manages a single writing session. The data manager maintains the logging data of a writing session.

The central system has three subordinated systems: playback, record and analysis. These three subsystems are all dependent on the central system:

- The playback system needs logged data to replay a writing session.
- The record system builds the logging data.
- The analysis system uses the logging data for further analysis.

The session manager can run any type of analysis by initializing the right class (e.g. PauseAnalyzer initializes the pause analysis). Per analysis the analyzer and the parser interact with each other.

⁷ Take into account that replaying requires a controlled test environment: the screen settings must be exactly the same between logging and replaying (cf. section 3.4.2).

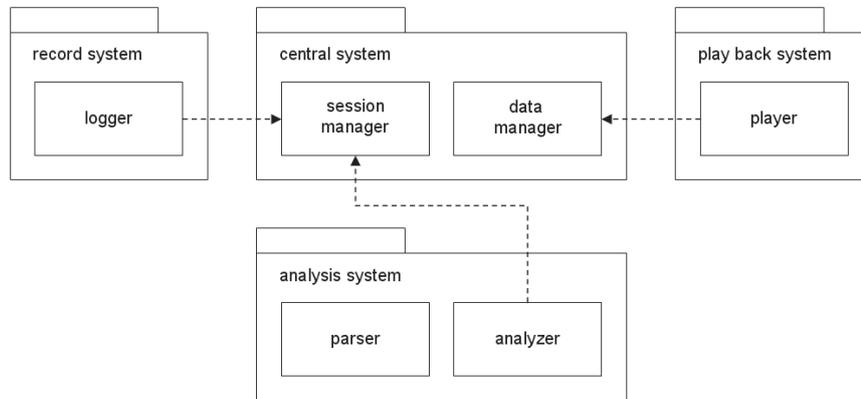


Figure 3. Inputlog system topology.

The classes' player and logger are so-called 'singletons', because only one of them can be active in the system. Consequently, Inputlog always needs to perform a consecutive analysis to verify that only one of these two subsystems is active: the logger cannot log while playing and vice versa. The three subsystems function independently. The GUI interacts with the central system to log, analyze or play a logging session.

3.3 Structure of IDF-files

Bytes	Meaning												
4	starting time of the log in milliseconds												
14 * 4	14 digits that indicate the size (viz. number of signs) of the session variables (6 values for the defined variables, 4 for the user defined variables and 4 for the undefined values)												
14 * #	14 session variables * number of the size that has already been read												
# * 20	serial number of keyboard and/or mouse event (type Eventmsg)												
	Each event that is logged by Inputlog creates a log event of 20 bytes. An Eventmsg is:												
	<table border="1"> <thead> <tr> <th>bytes</th> <th>meaning</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>message type (key in, key out, mousemovement, left mouse button in, left mouse button out, etc.)</td> </tr> <tr> <td>4</td> <td>paramL for keystrokes: virtual keycode for mouse events: x-value of location of the mouse</td> </tr> <tr> <td>4</td> <td>paramH for keystrokes: scancode for mouse events: y-value of location of the mouse</td> </tr> <tr> <td>4</td> <td>timestamp in milliseconds starting from the beginning of the log (time event – starting time of computer) * starting time of the log</td> </tr> <tr> <td>4</td> <td>handle to active window</td> </tr> </tbody> </table>	bytes	meaning	4	message type (key in, key out, mousemovement, left mouse button in, left mouse button out, etc.)	4	paramL for keystrokes: virtual keycode for mouse events: x-value of location of the mouse	4	paramH for keystrokes: scancode for mouse events: y-value of location of the mouse	4	timestamp in milliseconds starting from the beginning of the log (time event – starting time of computer) * starting time of the log	4	handle to active window
bytes	meaning												
4	message type (key in, key out, mousemovement, left mouse button in, left mouse button out, etc.)												
4	paramL for keystrokes: virtual keycode for mouse events: x-value of location of the mouse												
4	paramH for keystrokes: scancode for mouse events: y-value of location of the mouse												
4	timestamp in milliseconds starting from the beginning of the log (time event – starting time of computer) * starting time of the log												
4	handle to active window												

Figure 4. Structure of the IDF-file.

Each IDF-file contains the data of one logging session. In this file the logging data are combined with the session identification data (see section 4.2). The structure of the IDF-file is shown in Figure 4.

3.4 Program settings

Inputlog is dependent on some basic settings that are particular to the computer configuration that is used to log the writing session. We shortly discuss three configuration elements that are relevant when using Inputlog: system requirements, screen settings and keyboard lay-out.

3.4.1 System requirements

The computer must meet the following minimum requirements: Microsoft Word XP, Pentium III, 400 Mhz, 128mb RAM and 50MB additional disk space. In addition, Microsoft.Net Framework 2.0 must be installed, because the interface was developed in Visual Basic.Net⁸.

3.4.2 Screen settings

As stated before, Inputlog replicates the writing session exactly as it has been logged. In other words, it is not a video recording, but an actual playback of the writing session⁹. Therefore, it is important that the computer settings - both screen and program settings (e.g. toolbars, language settings, personal dictionaries etc.) - are exactly the same when replaying a writing session as when the writing session was recorded.

A short example is given to illustrate this issue. A writing session is recorded with only the standard toolbar active. However, between the logging session and the moment the observation session is replayed, the formatting and reviewing toolbar are added to the working environment, resulting in a different positioning of certain (graphic) elements on the screen in comparison to the previous screen outline. Consequently, because the replay uses the graphic xy-position of mouse clicks, the cursor might select a different icon or item than it did at the time the session was recorded.

Figure 5 shows, for instance, that during a replay with different screen settings, instead of changing the font into size 12, the paste-icon was selected. The result is that the continuity of the writing process reconstruction is severely disturbed. That is why we recommend researchers to keep the settings of the computer screen exactly the same between logging and replaying.

Note that this particular point of interest should only be taken into account when using the replay function of Inputlog. The generation of the different analyses is not disturbed by different settings.

⁸ These requirements are related to Microsoft Word. Researchers can download an additional tool from the Inputlog website to check if the research computer can run Inputlog.

⁹ We advice researchers who would like to use this replay as a resource for e.g. retrospective interviews to combine Inputlog with an on-line video registration tool (e.g. Camtasia by Techsmith). A combination of these tools does not lead to any conflict. Keep in mind that the system requirements of on-line video registration tools are quite high.

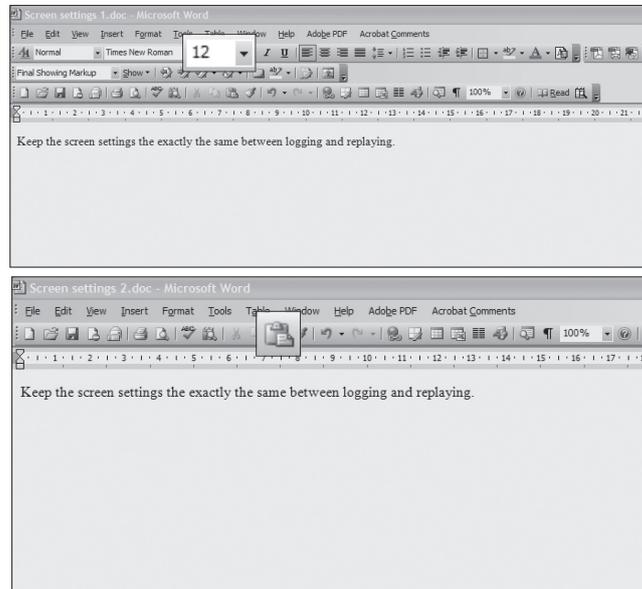


Figure 5. Illustration of playback with different screen settings.

3.4.3 Keyboard layout

A second configuration aspect that can influence the logging of a writing session is the keyboard layouts of the computer used during the logging session. As a result of the vast amount of different keyboard layouts, Inputlog has to detect the correct layout of the keyboard used. For instance, if a writing session is logged with QWERTY-settings and the actual layout of the keyboard is AZERTY, the following sentence will be represented incorrectly in the logging output:

<p>Typed sentence: This is a short writing session in Inputlog. Logged sentence: This is q short zriting session in Inputlog.</p>
--

To avoid this problem Inputlog is programmed to detect the correct hard-coded keyboard layouts for the lowercase characters and reads the connected Windows settings for the uppercase characters. At this moment, 33 different keyboard layouts are predefined in Inputlog 2.0 Beta (see the program's help-file for a detailed list).

4 Functional description

The previous section described the technical background of Inputlog. In this section we shortly 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 (see Figure 6). 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.

4.1 Entry screen

Inputlog starts with an entry screen that provides the user with a short overview of the program. The user can opt to (a) start a new logging session, (b) generate files from an existing logging file, (c) integrate various data, or (d) replay a logged session. The different tabs give the user direct access to the different functions. By using the main menu items at the top of the screen - File, Tools and Help - basic file operations can be activated, program settings can be changed and the help file can be accessed.

4.2 Record

By selecting the record tab, users can start a new logging session in the Microsoft Word environment, in I-pad or without any predefined program. The user can open Windows based software like the Internet, another Word processor such as Works or WordPerfect or programming software¹⁰.

Microsoft Word is the most common word processor at the moment and is therefore very familiar to most writers. I-pad on the other hand looks quite similar to Microsoft Notepad, but is more restricted. This 'in-house' developed limited word processor also provides x/y positions for every character and allows for more fine grained revision analyses in Inputlog. This analysis is still under development.

Next to the keyboard based word processing information from Microsoft Word or I-pad, researchers can integrate speech data from Dragon Naturally Speaking 8.1. To do so, the user has to select the required profile.

Before a new session is started the researcher can first specify the logging file and the identification data for a specific session. It can be identified by a maximum of ten variables (6 predefined and 4 user defined variables). These variables should enable the researcher to identify a writing session in detail. This information is included in the headers of the generated analytical files based on the session source file. In the file information, users can indicate where they want to save the logging session on their computer hard or network drive¹¹, and they can enter the unique filename for the source file, for example FirstnameLastname1 (= Firstname participant + Lastname participant + Number of session). The (source) file that will be generated in the recording session has the extension *.IDF, which is added automatically. This file will be used as input for the generate, integrate and play functions (cf. infra).

¹⁰ Researchers should keep in mind that for certain analyses (e.g. analyses on sentence or paragraph level) only the data logged in Microsoft Word can be used. For this kind of analysis, the basic data are interpreted with a set of algorithms based on Microsoft Word.

¹¹ It is recommended to create a unique folder per participant and per session. This will facilitate the administration of research projects.

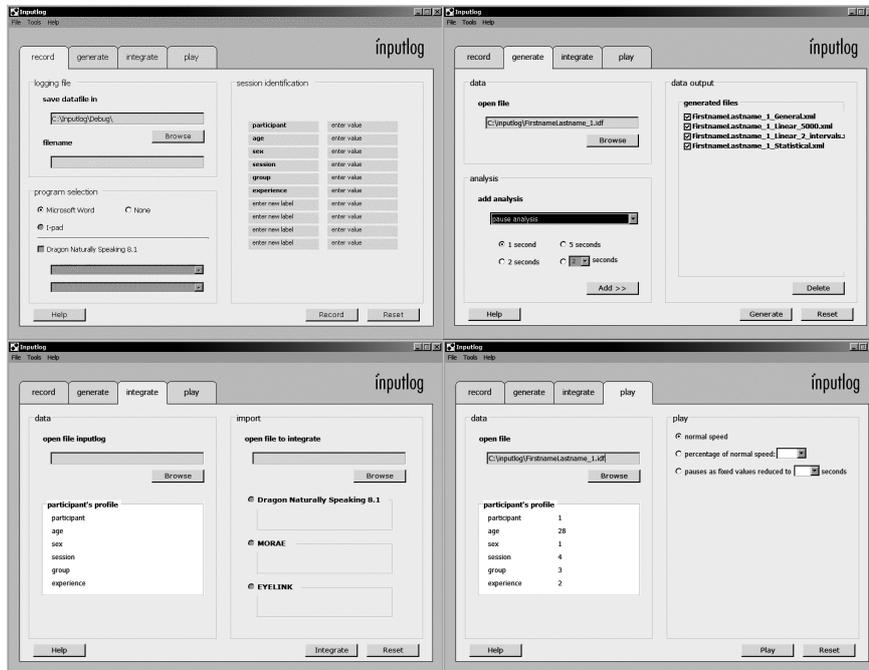


Figure 6. The four main tab pages of the Inputlog interface.

4.3 Generate

The generate tab opens a window showing the different analysis options. In this part of the program, analysis files can be generated on the basis of a source file that was recorded in a previous logging session. In other words, any IDF file can be opened at any time to generate data output files for specific analyses. The generate window consists of three sections. In the 'data section', users can specify the IDF file from which they would like to generate the analyses. In the analysis section the researcher can specify which output needs to be generated, by adding various types of analyses (e.g. pause analyses with 1, 2 and 5 seconds). In the section 'data output', a list of all the files to be generated is displayed. At this stage the researcher start generating or can delete an individual analysis or reset the whole generating procedure.

Inputlog 2.0 Beta offers 4 different data analyses:

1. General logging file: an XML file with the basic logging information of the writing session in which every line represents an input action (keyboard, mouse click or movement and - if present - speech); for every input action the session information is stored together with an identification of the input, the time stamp, the pause time that followed it, and – for a mouse operation - the xy-value of the

screen position (for an overview of the outputs of the analyses see Appendix 2)¹². The XML file can be converted to an Excel file or can be exported to SPSS for further analyses.

2. Linear text analysis: a plain linear text in XML-format with the complete linear production of the text (keyboard and speech) including mouse movements and pauses. The linear analysis is divided into two options: on the one hand researchers can generate 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) of their choice. In both options the threshold for the pause length can be adapted to meet the requirements for a particular study.
3. Statistical analysis: an XML file with basic statistical information of the writing session on a more aggregated level. Several process characteristics are shown, such as the number, mean and standard deviation of characters, words, sentences and paragraphs produced, pause times (based on the threshold entered in the interface) and the use of the different writing modes.
4. Pause analysis: an XML file with analyses of every non-scribal period. The threshold for the pauses can be set to 1, 2 or 5 seconds as a standard 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, pauses are summarized per word, sentence and paragraph location.

One other analysis is under construction at this moment:

5. Revision analysis: an XML file with a basic analysis of the number, the level and the kind of revision that has taken place during the writing session (see section 7 further developments).

Inputlog takes some time to generate the requested files; the progress is shown via the MS DOS screens that automatically appear. The different files are all placed in the same folder as the source file of the selected writing session (cf. 4.2 Record). This allows users to keep track of the recordings and analyses per writing session. The XML files that are generated can be read as such or can be imported into various statistical programs. As shown in Figure 1 the data structure of output files has a column structure. The general file can be used as a search file and a back-up file. The linear text files are represented in a more readable format. The researcher can easily switch between these two files (cf. section 4.3.2 Linking) to search for information. The statistical and pause analyses are structured in one column. This enables researchers to

¹² In this chapter we have opted to append examples of the various output files. For more detailed information on these outputs we refer to the Inputlog website and the program's help file.

transpose the data to a row structure and subsequently combine the data from various participants to a large data set. See Appendix 2 for further information on the XML files that are generated.

As stated before, researchers should take into account that keystroke logging provides a lot of data that need to be carefully analyzed (Spelman Miller & Sullivan, 2006). Researcher can adapt output files to provide the correct information to their research question, for instance, pauses that are recorded in keystroke logging serve different purposes, and therefore, describe different writing processes (Schilperoord, 2002). Also definitions on for example pause locations should be carefully kept in mind. The definitions that researchers have in mind might not be the same as the one the program uses.

In the sections below we provide more detailed information about how researchers can influence and use the data structure to get the best result for their analyses. The output files have some special features that we would like to elaborate on: selection of specific information by filtering the data output, linking between output files, error correction and generation via command line.

4.3.1 Filter data output

The XML structure of the data output enables researchers to adapt the data file to their needs by selecting the most important parts of a writing session, or in other words to filter out the parts that they do not need. This feature can be used to delete, for instance, the start of a writing session. In the beginning of a session participants might have some additional questions about a task while Inputlog is already recording, or the initial reading and planning time is informative as one measure, but not in the whole analysis. It can also be used to filter out data during a writing session, for instance, mouse movements that are related to search actions on the internet during a writing session. Again, the searching behavior of participants might be interesting in one measure, but not in the analyses as a whole. In such cases researchers can opt to filter the data output at a later stage. Consequently, the files will become more accurate if certain data are deleted.

The reduced basic XML file that is generated can easily be filtered via MS Notepad. Researchers only need to remove the irrelevant segments to be left with the adequate data for answering a research question (see the program's help file for more detailed information).

4.3.2 Linking

Inputlog 2.0 Beta offers the possibility to switch between output files (viz. general and linear, cf. *infra*) by linking time stamps (StartClock, cf. Figure 1). When using keystroke logging as a research tool, interpreting your data is an important step. Detailed case analysis may help to illustrate certain findings. If we take for example a writing study on press releases in which the researcher is interested in preformulation, he or she might be interested in a particular word, name or phrase that should be integrated

in the preformulation. It is easy to search for these kinds of segments in the linear text representation. The links on the time stamps related to these text segments make it possible to specifically locate the episodes in the general logging file, providing more detailed and fine-grained information about for example pause times preceding the preformulation. A reverse example might be that the researcher is interested in pauses that fall within a certain range. These pauses can easily be selected in an Excel representation of the general file (viz. by conditional formatting) and can then be seen in their writing context in the linear file, making a contextual interpretation easier.

4.3.3 Error correction

At present, more than a hundred researchers from all over the world are using Inputlog as a tool for writing research. As mentioned before, an important configuration aspect that can influence the logging of a writing session are the keyboard layouts of the computer used during the logging session. Although we have predefined different keyboard layouts some characters or symbols are not known by Inputlog. This may hinder the logging process. In that case researchers can search via MS Notepad in the reduced basic XML file and replace the unknown character by a known character (see help for more detailed information).

4.3.4 Generation via command line

Professional users can also use the command line of MS DOS to generate various analyses. This command line is related to the interface. An advantage of this command line is that researchers can generate an enormous amount of various relations in one command and which they can save in a Batch file. The commands that need to be used to create the command line can be found in the help file.

4.4 Integrate

The third tab of Inputlog is Integrate. This module has been developed on a conceptual level and the actual integration is being performed manually at the moment. This functionality will be available in the summer of 2007. This module will allow researchers to merge different XML output files of other logging and observation programs. At this point there are three main programs that we are planning to integrate with Inputlog: Dragon Naturally Speaking 8.1, Morae and Eyalink.

4.4.1 Dragon Naturally Speaking 8.1

As mentioned before Inputlog data can be combined with speech recognition output from Dragon Naturally Speaking (cf. Figure 1). 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 need to count with the same time measure. More specifically, Inputlog and Dragon Naturally Speaking need be based on an absolute

epoch time¹³, resulting in a new XML file that will be used as starting point for the analysis.

4.4.2 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. Morae is a macro oriented observation tool developed by Techsmith (www.techsmith.com) for usability research purposes. It also enables researchers to code a process on various levels while observing it online. It is our intention to complement the Inputlog data with Morae logging data. This program captures, for instances, changes between programs on a higher level and registers, for instance, the url-addresses of websites that are accessed during a writing session. In addition, researchers can flag important segments of writing sessions and code these segments leveling great detail. Just like Inputlog, Morae also logs very detailed timestamps which should enable us to integrate the additional data registered by Morae into the output of Inputlog. For the observation of writing processes during which the participants combine Microsoft Word with other programs especially, this integration opens new perspectives for further analyses. In the research project described in chapter 7 we combine Inputlog with data from Morae (and Dragon Naturally Speaking).

Figure 7 shows an example of a merger between the output files of Inputlog, Dragon Naturally Speaking and Morae. As seen in Figure 1 Dragon Naturally Speaking data are shown per dictated segment and the Inputlog data per event.

In this example four extra columns are added (cf. Figure 1: for the English translation of this excerpt see page 232). These extra columns are the clock time of Morae, the merge time (in milliseconds) between Morae and Inputlog calculated manually in MS Excel and SPSS, Morae's marker and corresponding code.

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 opts to switch into writing mode to delete the full stop with his keyboard and subsequently types an interspace. He proceeds with speech and dictates the final segment once again. It is striking though that 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

¹³ The start time of the epoch is January 1, 1970, 00:00 h GMT.

observation revealed that he prefers to continue with text production and to finish his paragraph.

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 7. Example of merged output file with data of Inputlog, DNS and Morae.

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 example to search through the information in a more structured manner and to combine more global analyses with very detailed analyses.

4.4.3 Eyelink

Finally, we are exploring the possibilities to integrate eye tracking data with Inputlog data. At the moment ScriptLog is the only logging tool that combines eye tracking and keystroke logging¹⁴ (Andersson & et al., 2006). Eyetracking is ideally suited for research on reading and writing processes. In chapters 5, 6 and 7, we study the effect of the text produced so far in more detail. In these studies pause time was one of the measures that we used to describe the cognitive load that the text produced so far imposes on a writer. Pause times may be indicative of monitoring the text produced so far. However, there is no direct indication of what writers are looking at and whether they, for example detect an error in the text produced so far. The recurring comment made in those chapters is the possibility to obtain more fine-grained data about the writers' monitoring activities during text production¹⁵. Eyetracking enables researchers to see what writers are looking at during a writing task.

Eyelink is one of the head mounted eye tracking systems of SR Research¹⁶. 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, in this situation the Inputlog data can provide more insight into the eye tracking data.

¹⁴ ScriptLog uses the eye tracking system iView X HED + HT (www.smi.de)

¹⁵ The study described in chapter 5 and 6 was replicated with the addition of Eyelink (Thomas Quinlan was the coordinator of this project).

¹⁶ More information on Eyelink can be found on www.eyelinkinfo.com

4.5 Play

The final tab of the Inputlog interface is the play function. A recorded writing session can be replayed using Inputlog. Again, the IDF file is used as a source file for the replay. To verify the participants' profile of the file that is selected for a play back session, all defined variables of the session identification appear in the dialog box on the left side of the screen. 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. However, we would like to restrict this option to a maximum of 120% of the original speed. Otherwise the program may have difficulties in processing and performing certain actions that require extensive memory access. 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.

4.6 Help file

So far, this chapter has described the possibilities of Inputlog, the most important technologies and main functionalities. For more information on certain features we refer to the help file. The help file contains a structured manual of the program. The main part of the help file consists of four chapters, each corresponding to the four basic functions of the software tool: record, generate, integrate and play. The help file on each of these functions is also directly available via the help button on the associated tabs. Furthermore, we integrated rules and definitions. Researchers should be able to have a clear idea of what they are measuring when using Inputlog as a research tool. This explains why we detail, for example, all the rules that are used to define pause locations and why we provide definitions of standard deviation.

5 Applications

To illustrate the research potential of Inputlog, we describe three possible applications in this section. In the first two applications, we briefly discuss writing processes taken from experiments in which we have used Inputlog as a research tool. The third application shows Inputlog as a research instrument in a broader sense.

Keystroke logging can be a suitable research instrument in a number of contexts (for an overview see Sullivan & Lindgren (2006)). Research areas that Inputlog can provide data for include: studies on cognitive writing processes in general, description of writing strategies in professional writing or creative writing, the writing development of children, of those with and without writing difficulties, 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. Next to strict writing research it can also be integrated in educational domains: second language learn-

ing, programming skills, typing skills and again subtitling¹⁷. For example, Lindgren (2003) used keystroke logging to facilitate reflection through peer-based intervention by replaying the writing session (in an educational context). Next, Sullivan and Lindgren (2006) discuss in their keystroke logging book how keystroke logging can also be used to investigate theories that do not primarily concern writing, but for which writing provides a window on the theory. Case in point is Galbraith's (1999) dual-process model. They state that:

To use key-stroke logging, coding criteria would need to be developed that would permit the researcher to decide when a new idea had been developed in the evolving text and to decide when an editing action can be defined as a revision as defined within the dual-process model.

Theory-building through the use of keystroke logging holds tremendous potential.

In this section we show two possible research designs - a case study versus an experimental study - in which keystroke logging can be used. The first example is taken from a case study of a writer producing a bad news letter; the second experiment features a more technical writing experiment focusing on the working memory requirements necessary for error detection in the 'text produced so far' (see also chapters 5 & 6).

5.1 Case study: pausing and revision behavior in writing bad news letters

In Figure 8 we show the linear text representation of a writer producing a bad news letter in which (s)he declines an offer to deliver a keynote address for an international conference. We were especially interested in the process characteristics of the writing episode that concerned the wording of the bad news itself. As we know from the literature on this issue, the strategic considerations to make 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).

The writer needed about ten minutes to finish the letter of about 130 words. The replay function of Inputlog was used as a stimulus for a retrospective think-aloud protocol. Figure 8 shows the sequential linear output (periods of 30 seconds) of a fragment that illustrates the writer's strategic considerations at the beginning of the second paragraph.

0:02:30	{8440} Presenting a {1330} paper to {2360} this group {3660} [CTRL+LEFT] {1020} quality {180} [END] {1030} deserves a {1530} thor
0:03:00	ough {1390} and {2420}time {1190} [BS 4] {1660} comprehensive effort {2130}. {3610} Of course, [BS 12]. {1020} Obviously {2900}
0:03:30	, such an effort {2030} requires {5550} a lot of [BS 8] {1050} considerable [BS 12] time. {4840}

¹⁷ In January 2007 we started a research project entitled 'Live subtitling using speech recognition: procedures for quality improvement' (in cooperation with the Flemish Radio and Television Broadcast station).

0:04:00	However, {1340} my schedule {2660} [CTRL+LEFT 4] {1280} [DEL 10] M [END] {2380} is fully comm{1770}itted to a
0:04:30	writing project {2300} {7170} [CTRL+LEFT 7] {2160}

Figure 8. Linear output of two fragments of a writing process in which a bad news letter was produced.

The fragment shows a very fragmented and staccato writing process that starts off with a long pause of more than 8 seconds (8440 milliseconds) in the beginning of the second paragraph (announcement of the refusal). In the production of about 30 words, there were 28 pauses longer than one second, which is substantially more than in the previous period when the introductory context of the letter was written. About half of the time in this fragment is used for pausing, showing the time attributed to careful and strategic formulation. An interesting example of such a strategic consideration is the revision that takes place after four minutes (0:04:00). The participant starts the sentence with 'However', but two words later he rereads the beginning of the sentence and realizes that this contrastive connective announces the bad news in a too early stage of the paragraph. Therefore, he decides to delete the connective to neutralize the context of this argumentative sentence.

Again, this example shows that process logging enables researchers to analyze writing processes from different perspectives enriching possible interpretation based on text analysis. Other observation methods, like recording (retrospective) thinking aloud protocols, might complement the acquired data.

5.2 Experimental study: the text produced so far and the use of working memory

An experiment was set up to assess the memory load while interacting with the 'text produced so far' during the text production phase. The design included the most frequently occurring error types found in a case study of professional writers that were using speech recognition for the first time to write business texts (Leijten, Ransdell, & Van Waes, submitted, also chapters 5 & 6). In the example at hand, we selected two text fragments in which a sentence with large speech recognition errors is corrected. The errors that occurred in the text produced so far were considered major errors because the number of characters that differed from the intended text was more than two. Besides, the selected errors could only occur in speech recognition, because they were misrecognitions that resulted in phonologically similar words. The task in the experiment consisted of a set of sentences that were presented to the participants as contextual information. In Figure 9 an example of a correct and an incorrect sentence is given (see sentence 2: translated from Dutch). After every context sentence (1) the participants had to click the 'ok' button, to indicate that they had finished reading the sentence. A subclause of the previous sentence was then presented as text produced so far (TPSF) in a subordinate causal structure (2), and the participants were prompted to complete the sentence (3) on the basis of the context provided earlier (1).

<p><i>Correct</i></p> <ol style="list-style-type: none"> 1. Because the height was not indicated, the pick-up truck drove by the underpass. 2. The pick-up truck drove by the underpass, 3. because the height was not indicated. <p><i>Incorrect</i></p> <ol style="list-style-type: none"> 1. Because the height was not indicated, the pick-up truck drove by the underpass. 2. The picot trug drove by the underparts. 3. because the height was not indicated. <p>Dutch “De heeft week reed onder de brief door want de hoogte was niet aangegeven.”</p>

Figure 9. Examples of correct and incorrect sentences in the TPSF experiment.

The participants could either choose to correct the errors first and then complete the sentence, or they could complete the sentence first and then correct the error. In casu, for this sentence 83% of the participants preferred to complete the sentence first and correct the error afterwards. Figure 10 shows the linear output of the writing session of two participants that used different writing strategies. Pauses longer than 1 second are included in the output; texts are in Dutch.

Writer 1	Writer 2
sentence completion > error correction (83%)	error correction > sentence completion (17%)
[MwC.910,701-391,348] [McLeft] de hoogte was niet aangegeven {2190} [MwC.491,352-145,346] [McLeft] [BS 9] eftruck {1270} [RIGHT 17] [DEL 4] ug <1170> [MwC.165,335-933,699] [McLeft]	{2660} [McLeft] [MwC.910,711-149,347] [McLeft] [MwC.149,347-375,355] [BS 9] ftruck {1600} [MwC.372,357- 269,340] [Mselect] [MwC.270,340-342,341] [BS 4] ug <1.81> [RIGHT 15] de hoogte stond niet aangegeven <1250> [MwC.348,338-884,695] [McLeft]

Figure 10. Example of linear output of a text fragment generated by Inputlog.
 Remark: MwC: mousemovement without click; McLeft: mouse click left;
 Mselect: selection of text by mouse; BS: backspace; right 17: arrow right 17.

In these examples of two short writing fragments, different writing strategies can be distinguished. The first writer prefers to continue completing the text first, before correcting the mistake in the TPSF. He positions his cursor after the first segment (TPSF; cf. xy-value of left mouse click) – without a significant previous pause – and then completes the sentence. After the sentence is completed, he pauses for 2.19 seconds. He then positions the mouse behind the incorrect word and deletes it by using the backspace key. Next, he navigates through the text by pressing the right arrows key and deletes the second error. Finally, he pauses briefly before moving on to the next sentence.

The log of the second participant's writing session reveals a different pattern. This writer pauses before he starts writing and then positions the cursor behind the error to

correct this first. After correcting the second error he completes the sentence. In the analysis of the research data presented here, we were mainly interested in a description of the interaction with the text produced so far. Differences in writing strategies can be related to both individual differences and to error characteristics in the TPSF. This short example illustrates how the detailed process information that is generated by Inputlog provides a basis for analyzing writing strategies that are no longer visible in the final written product¹⁸.

5.3 Applications on a broader level

Inputlog is a research tool that logs writing processes. In addition, 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 chapter 7, we recorded the retrospective interviews of the participants via Inputlog and Dragon Naturally Speaking. As a result, the retrospective interview was automatically transcribed.

This too 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 production of protocols.

6 Data analysis

As mentioned earlier, keystroke logging generates enormous amounts of data. Hence, we would like to describe three methods for analyzing Inputlog data. The first perspective is the use of factor analyses on the aggregated data of Inputlog. The second and third perspectives are related to developing text visualizations.

6.1 Factor analysis

Keystroke logging programs have the advantage that they collect data on a very low level. However, this advantage also has its drawbacks. It is sometimes very hard to interpret the large amount of data. Moreover, the main objective of logging these data is to indirectly measure some aspects of cognitive writing processes. Unfortunately, deciding which of the reported measures are driven by the same underlying variable can be difficult.

A statistical technique that might be helpful in overcoming these problems is *factor analysis*. In this section we shortly introduce the main procedure for using factor

¹⁸ Moreover, writing processes can be hard to register on-line, since pauses are related to various writing subprocesses (Schilperoord, 2002). A combination with thinking-aloud or retrospective interviews may help in these cases.

analysis as a way to explore data collected and analyzed with Inputlog¹⁹. In his introduction to factor analysis, Field (2005, p. 619) explains that this technique has three main uses:

1. to understand the structure of a set of variables;
2. to construct a questionnaire to measure an underlying or latent variable;
3. to reduce a data set to a more manageable size while retaining as much of the original information as possible.

In the perspective of writing research with logging programs, the third function is of great value. Factor analysis²⁰ can help to reduce a set of interrelated variables into a smaller set of so-called factors (or latent variables). In other words, the objective of this technique is to explain the maximum amount of variance by using the smallest number of explanatory concepts. Factor analysis reduces large amounts of measures taken to their underlying dimensions and cluster variables in a meaningful way. This is achieved by looking for variables that correlate highly with a group of variables, but which do not correlate with variables outside of that group.

Exploratory factor analyses on Inputlog data can be conducted by following this procedure:

1. Select variables: All the variables in the statistical and pause analysis of Inputlog can be selected (e.g. variables related to text length, duration of the writing session and pausing behavior). These general measures can be complemented with calculated ratios (e.g. words produced during the writing process divided by the number of words in final text, as a possible indication of the recursivity of the writing process).
2. Conduct the factor analysis: Run the factor analysis by selecting the variables you want to include in the exploratory analysis (e.g. by using SPSS).
3. Check the variables in a correlation matrix: The correlation matrix (R-matrix) gives an overview of the significance value of each correlation. Because factor analysis has to include variables that correlate fairly well, you can use this matrix to evaluate the pattern of relationships. It is recommended to exclude strongly correlating variables (e.g. a correlation coefficient higher than .9) and variables that hardly or do not correlate. If necessary, rerun the factor analysis (step 2)²¹.

¹⁹ We would like to thank Gert Rijlaarsdam (University of Amsterdam) who presented this method during a workshop in Antwerp (November 2006).

²⁰ There are many different techniques or methods for conducting a factor analysis (for example, principal axis factoring, maximum likelihood, unweighted least squares, generalized least squares) and also many different types of rotations that can be done after the initial extraction of factors. Because this is only a short introduction to factor analysis, we will mainly base our description on a simple principle component analysis in combination with a varimax rotation.

²¹ SPSS also offers the possibility to report a determinant value (which should be different from 0), the Kaiser-Meyer-Olkin Measure (minimum value > .6) and the Bartlett's test of sphericity (to test whether the correlation is an identity matrix). In sum, these tests provide a minimum standard for conducting factor analyses.

Table 1. Example of an SPSS output showing the Total Variance Explained in a factor analysis

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	4,949	32,992	32,992	4,949	32,992	32,992	4,201	28,007	28,007
2	3,018	20,120	53,112	3,018	20,120	53,112	3,018	20,121	48,128
3	1,727	11,510	64,622	1,727	11,510	64,622	2,430	16,200	64,328
4	1,136	7,576	72,198	1,136	7,576	72,198	1,181	7,871	72,198
5	,977	6,515	78,713						
6	,854	5,693	84,406						
7	,605	4,033	88,438						
8	,514	3,427	91,866						
9	,426	2,838	94,704						
10	,371	2,474	97,178						
11	,185	1,232	98,410						
12	,093	,619	99,029						
13	,080	,533	99,562						
14	,038	,250	99,812						
15	,028	,188	100,000						

Extraction Method: Principal Component Analysis.

- Determine the factors: In the principal component analysis the components are ordered on the basis of the initial eigenvalues as well as a cumulative percentage of the total variance. By default, SPSS uses a cut-off point of the eigenvalue that is larger than 1 and uses all the factors above this threshold to optimize the factor structure in a Rotation Sums of Squared Loadings. In the example we present in Table 1 fifteen variables were used in the factor analysis. Using the above mentioned criterion (eigenvalue > 1) we can account for roughly 72 % of the total variance by using only four components or factors, which is a radical reduction of the total number of variables.
- Interpret the factors: To make the factors more accessible you can use a Rotated Component Matrix to see which variables correlate with which factor. Because factor analysis is an exploratory technique it is now up to the researcher to make sense of the factors. However, the loading values reported in the matrix are very helpful in making sense of the contribution of each of the variables to a specific factor. Also, comparing individual cases on the basis of the selected factors can be useful for interpreting the factors. In our example, for instance, the first factor was mainly characterized by a low number of produced words, a large pause time with especially fairly long pauses in the first half of the writing process and a high ratio of number of words produced vs. words in the final text (little revision). The second factor on the other hand was characterized by a much lower ratio, a high number of (short) pauses and a relatively long total production time. As this illustration shows, these factor descriptions create an interesting basis for relating them to writing profiles.

6. Conduct a multi and/or univariate analysis: In the last step of the procedure, the factors can be used as variables, for instance, to test the effect of experimental conditions.

In sum, factor analysis is a possible way for the researcher to reduce of the set of measures Inputlog datafiles produce. The short description presented above details a possible procedure for exploring the data by using this statistical technique and exploring the underlying dimensions of clusters of variables in a meaningful way.

6.2 Progression analysis and GIS

The development of the text is currently represented in Inputlog in the linear text analysis. To visualize this textual development we would like to extend Inputlog with two graphical representations of the text progression, a basic and a more extensive one. In the basic progression analysis we would like to visually represent the number of characters that are produced at each moment during the writing process taking into account the characters that are deleted at that stage. This basic progression analysis is based on Perrin's (2003) writing strategy research.

Because this basic analysis is a static reproduction of the writing process, we would also like to develop a more interactive representation inspired by Lindgren (2002; 2007). She uses a Geographical Information System (GIS) to visualize and summarize the writing process. GIS enables researchers to analyze different subprocesses of the writing process by selecting representative variables. The graphical representations are not static. Instead they allow a researcher to interact with the data at different levels and to move back and forth between the data and their representation. Consequently, as well as being a tool for visualization and data mining, this technique can support a dynamic analysis of the cognitive processes during writing due to the interactive nature of the data mining approach on which GIS is based. Figure 11 shows an example of a GIS graph that is based on a manually adapted dataset of Inputlog²². To generate these kinds of progression analyses automatically, we first need to further optimize the revision analysis because these analyses components are based on the revision output.

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 line indicates the total character production including deleted characters at each point in time; the bottom line indicates the characters retained after deletions at each point in time. The dotted line shows all the points in time at which the writer is working on the text, representing both pauses and deletions. The size of the circles refers to the length of the pause. When the line drops, a number of characters are deleted.

²² We would like to thank Eva Lindgren, Umeå University (Sweden) who generated this graph for us with ArcGIS (ESRI).

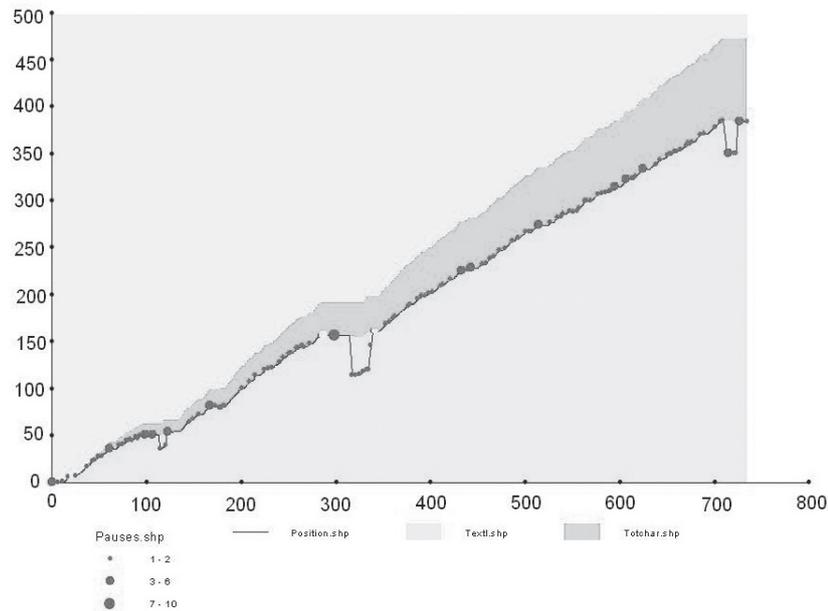


Figure 11. Text progression graphic as logged in Inputlog and visualised in a GIS application (source: Lindgren).

7 Further development

We have identified four important niches that may increase the applicability of Inputlog, especially in the domain of writing process research. In the near future we would like to further develop the following (in order of priority).

7.1 Revision analysis

Work on the revision analysis component for Inputlog has already been started, but because of its complexity it is very time consuming to further stabilize and extend the analysis module. In the revision analysis we would like to produce an output analysis in which different characteristics of in-process revisions are described, for example the number of revisions, type of revisions, level of revisions, number of words and characters involved in the revision operation, as well as the location of the revisions in relation to the point of utterance. To define revisions we have developed an algorithm and a set of rules. The revision analysis first of all defines critical events in the writing process that can be linked to a 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 to revise or the positioning of the cursor, the (possible) deletion of the

text and the end of the revision. In Figure 12 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.]¹ {evolution.}

The first operation is a very basic one: the writer simply uses the backspace key at the point of utterance to delete the full stop and the word ‘progression’ and then types the new word ‘evolution’ (see rule 1). 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: 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 (see rule 2).

	begin (movement)	selection/positioning	deletion	end (movement)
1	-	-	backspace	-
2	mouse movement left	left click	delete	arrow right
<i>n</i>

Figure 12. Examples of rules defining revisions.

At the moment we have predefined about 50 sets of rules to test the algorithm for deletions and substitutions. However, after the testing phase, the rules will have to be extended and further tested to cover a more complete range of revisions.

7.2 Subtitling via speech recognition

Speech recognition software is now commonly used during simultaneous subtitling of (live) television broadcasts via ‘respeaking’. Running commentary and live interviews during the live broadcasts are dictated with speech recognition software to a subtitling program. The speed of text production in speech recognition increases the possibility to subtitle more and more programs. Broadcasting stations that use this technique include: the BBC in England, the VRT in Belgium and the NOS in the Netherlands.

Inputlog can be used to describe the writing process during subtitling in order to contribute to the improvement of this ‘writing’ method. Live subtitling always takes place under high time pressure. The translation of spoken text to written text makes it an extra complicated task. From a methodological point of view, these writing processes are very interesting since they allow us to observe human-computer-interaction in a detailed manner. In this research area we would like to focus on three perspectives: speech recognition (how can the quality of speech recognition be improved?), subtitling (what is the best method - block versus scrolling - to use for live subtitling?) and cognitive writing processes (how can cognitive difficulties be

prevented?). The main goal of this research project is to increase the speed and accuracy of subtitling via a procedure for quality improvement.

7.3 Typing tests

The influence of a writing instrument on the writing process probably interacts with the degree of automation of the writing process at hand (Caporossi, Alamargot, & Chesnet, 2004). When writers start using the keyboard to write their texts, we see that it is influenced by motor skill and by linguistic elements (cf. developmental writing). Therefore, researchers might also be interested in measurements that evaluate and measure writers' typing skills and use it as an assessment and coaching tool in typing courses.

An interesting measure on typing skill that Inputlog can provide is the 'inter key intervals' (Nottbusch, Weingarten, & Sahel, 2007; Weingarten, Nottbusch, & Will, 2004) or the transition time (Grabowski, 1996; Wengelin, 2006). When elaborating parsing rules in Inputlog, a possible solution is a self-regulatory add-on that recognizes typing errors in copying tasks and shows statistics on a typing test (e.g. speed, accuracy, type of revisions). To this end, the revision analysis of Inputlog needs further improvement. The results of such a typing test could be used to define writer specific pausing thresholds. Typing performance can also be used as a covariate in analyses in which, for instance, process time is important.

Furthermore, detailed results of typing tests and key transitions also enable teachers to gain insight in the difficulties of a writer acquiring typing skills (e.g. left hand-right hand coordination during typing) and to teach adequate and very specifically diagnosed strategies. A related topic that we would like to address is the analysis of writing processes of dyslectic writers.

7.4 Searchable output files

The linear representation of writing sessions logged with Inputlog is helpful when searching for important information. In section 4.3.2 we described the possibility to switch between the general output and the linear text file to facilitate searching in the files. However, we would like to develop this feature in a broader context. At present, researchers can search 'manually' in the files and they can search for correct representations of words, for example a preformulation in a press release must contain the word 'bank stock'. This method is not flawless, since writers make typing errors that disrupt the linear representation of words. Parsing rules should make it possible to find strings with typing errors. In the interface researchers should also be able to define the view of the linear representation, for example each 'searched word' is represented on a separate row with the corresponding action and pause times.

This opens several new research perspectives for example for studies on text coherence. Researchers can easily filter causal relations that are linguistically marked and analyze the production and pause times.

In addition to the developments in these particular areas, we will pay special attention to the further development and optimization of the existing modules. In addition, the flexibility of adapting the output files of Inputlog to the specific needs of researchers is high on the agenda. Furthermore, the compatibility with different versions of the Windows operating systems and the Office environment will require constant attention.

8 Conclusion

Inputlog is a registration tool that enables researchers to register computer based writing processes in Windows environments and subsequently analyze and interpret the writing behavior of writers. In addition to other keystroke logging programs Inputlog also logs texts dictated by the speech recognition software Dragon Naturally Speaking 8.1.

Keystroke logging is an unobtrusive way to observe writing processes (indirect), and as such, it does not influence the writing processes of the writers. They might be aware that their writing process is logged, but other than that they are not confronted with them being a research object. This differs from thinking-aloud protocols (direct observation of cognitive activities) in which writers are very aware of the research situation (Janssen, Van Waes, & Van den Bergh, 1996; Van Someren, Barnard, & Sandberg, 1994).

We see various possibilities to integrate Inputlog as a research tool in writing process research. On the one hand as a research tool that can be used autonomously, to answer research questions on linguistic, textual and cognitive studies of writing, but also for broader applications concerning the development of language learning, literacy and language pedagogy (Spelman Miller & Sullivan, 2006, p. 1). Secondly, it can be used in combination with other research methods, for instance eyetracking, usability testing and thinking aloud protocols (to create triangulation, as described by Van der Geest, Leijten & Van Waes (2006)). Finally, it can be called in as an instrument to analyze other research methods, for instance, to measure pause durations within thinking aloud protocols that are automatically transcribed via speech recognition.

Notes

To facilitate a broad usage of Inputlog, the program is put at the disposal of the research community free of charge (www.inputlog.net), provided that reference is made to: Leijten, M., & Van Waes, L. (2006). *Inputlog: New perspectives on the logging of online writing processes in a windows environment*. In K. Sullivan & E. Lindgren (Vol. Eds.) & G. Rijlaarsdam (Series Ed.), *Studies in Writing: Vol. 18. Computer Key-Stroke logging and Writing. Methods and Applications* (pp. 73-94). Oxford: Elsevier.

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

marielle.leijten@ua.ac.be or luuk.vanwaes@ua.ac.be.

Acknowledgements

We would like to thank Wesley Cabus, Ahmed Essahli, Wim Claessens, Mathia Van de Poel and Bart Van de Velde for their excellent work in programming Inputlog (BOF 2005-2007). We would also like to thank the KdG-Polytechnic Antwerp for providing internships and a collective project on Inputlog (PWO 2005-2008; dr. Nico Verlinden). 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. Finally, we would like to thank Tom Van Hout for proofreading this chapter.

Appendix 1

Glossary

ArrayList	Implements the IList interface using an array whose size is dynamically increased as required.
BAT	A batch file is a text file containing a series of commands intended to be executed by the command interpreter. When a batch file is run, the shell program (cmd) reads the file and executes its commands, normally line-by-line. DOS batch files have the filename extension .BAT.
CPU	Central Processing Unit Reads software instructions and tells your computer what to do.
DLL	Dynamic Link Library A Windows library that can be shared by multiple applications
event	Every action on the computer
EVENTMSG	The EVENTMSG structure contains information about a hardware message sent to the system message queue. (This structure is used to store message information for the JournalPlaybackProc callback function.)
GIS	A Geographic Information System is a system for creating, storing, analyzing and managing spatial data and associated attributes. GIS is a tool that allows users to create interactive queries, analyze, and edit data.
GUI	Graphical User Interface
IDF	Inputlog Data File
IXF	Inputlog XML file (intermediate)
journalplayback	Public static final Hook.Descriptor JOURNALPLAYBACK Posts messages previously recorded by a JOURNALRECORD hook procedure.
journalrecord	Public static final Hook.Descriptor JOURNALRECORD Records input messages posted to the system message queue. (http://www.jniwrapper.com/docs/javadoc/winpack/com/jniwrapper/win32/hook/Hook.html)
scancodes	The data from a keyboard comes mainly in the form of scancodes, produced by key presses or used in the protocol with the computer. The PC keyboard interface is designed so the system software has maximum flexibility in defining certain keyboard operations. This is accomplished by having the keyboard return scancodes rather than ASCII codes. Each key generates a 'make' scancode when pressed and a 'break' scancode when released. The computer system interprets the scancodes to determine what operation it is to perform. (http://www.barcodeman.com/altek/mule/scandoc.php)
Virtual Keycode	Unique number that identifies one key
Windows Hook	A hook is a point in the system message-handling mechanism where an application can install a subroutine to monitor the message traffic in the system and process certain types of messages before they reach the target window procedure.
XML	Extensible Markup Language: which allows you to define the tags (markup) that you need to identify the data and text in XML documents.
XSL	XSL is a way of applying transformations and formatting to XML documents.

Appendix 2

Examples of outputs

label	action
.	interspace
BS	backspace (* number of keystrokes)
DEL	delete
SHIFT	shift on/off: this is visible in the output
Caps Lock	caps lock on/off: this is visible in the output
CTRL	control + key
ALT	alt + key
UP 5	up * number of lines up
DOWN 3	down * number of lines
LEFT 8	left * number of characters
RIGHT 8	right * number of characters
TAB	tab * number
ENTER	enter * number
HOME	home
END	end
PU	page up * number
PD	page down * number
INS	insert + character
Left Button	mouse click left
Movement	mouse movement (+ the start and end value of the xy-axis)
MouseWheel	scroll with the mouse scroll wheel & the Windows scrollbar (+ the start and end value of the xy-axis)
Right Button	mouse click right
Writing Mode	Input mode (1=keyboard, 2=Mouse, 3= Speech)
Output	Text that appears on the screen
Start Time	Time of key in: in milliseconds
End Time	Time of key up: in milliseconds
Action Time	Time between key in and key up: in milliseconds
Pause Time	Time between key in and key in: in milliseconds
Pause Location	Time of key up: hours:minutes.seconds,100th of a second
x value	Location of the mouse on x-axis
y value	Location of the mouse on y-axis

* The last two columns contain the values of the mouse. The location of the keystrokes is not detailed.

General Logging File

Session Identification	
Filename	FirstnameLastname_1.idf
Recording Time	28/01/2007 14:39:26
Participant	1
Session	4
Pause Threshold	500ms

WritingMode	Output	StartTime	EndTime	ActionTime	PauseTime	Pause Location	X	Y
2	Movement	15	32078	32063	15		426	313
2	Left Button	32172	32312	140	94		426	313
1	T	33687	33812	125	1515	7		
1	h	34093	34218	125	406			
1	e	34281	34422	141	188			
1	SPACE	34468	34593	125	187			
1	s	34843	35000	157	375			
1	c	35234	35468	234	391			
1	i	35422	35515	93	188			
1	e	35828	36031	203	406			
1	n	35984	36109	125	156			
1	t	36312	36468	156	328			
1	i	36500	36593	93	188			
1	s	36687	36843	156	187			
1	t	36906	37062	156	219			
1	ENTER	38015	38140	125	1109	3		
1	C	39218	39375	157	1203	3		
1	o	39531	39625	94	313			
1	l	39797	39890	93	266			
1	d	39968	40125	157	171			
1	p	40687	40797	110	719	1		
1	l	40922	41140	218	235			
1	a	41187	41328	141	265			
1	y	41281	41406	125	94			
1	SPACE	41672	41812	140	391			
1	(42312	42406	94	640	2		
1	NUM 2	42828	42953	125	516	2		
1	NUM 0	43125	43218	93	297			

WritingMode	Output	StartTime	EndTime	ActionTime	PauseTime	Pause Location	X	Y
1	NUM 0	43281	43390	109	156			
1	NUM 2	43500	43593	93	219			
1)	44531	44625	94	1031	2		
1	SPACE	45890	46031	141	1359	2		
1	LEFT	46750	46843	93	860	2		
1	UP	47953	48093	140	1203	7		
1	LEFT	48500	48609	109	547	7		
1	LEFT	48687	48781	94	187			
1	LEFT	48875	48953	78	188			
1	LEFT	49015	49109	94	140			
1	LEFT	49172	49297	125	157			
1	LEFT	49343	49422	79	171			
1	LEFT	49500	49593	93	157			
1	LEFT	49687	49765	78	187			
1	LEFT	49828	49937	109	141			
1	LEFT	50000	50109	109	172			
1	LEFT	50172	50297	125	172			
1	LEFT	50343	50422	79	171			
1	LEFT	50500	50609	109	157			
1	CTRL + B	52031	52140	109	1531	7		
1	RIGHT	53031	53172	141	1000	7		
1	RIGHT	53750	53843	93	719	7		
1	ENTER	54593	54703	110	843	4		
1	ENTER	55172	55281	109	579	4		
2	Movement	58047	63390	5343	2875	4	418	396
1	T	65265	65406	141	1875	4		
1	e	66156	66297	141	891	1		
1	l	67125	67218	93	969	1		
1	l	67312	67422	110	187			
1	SPACE	67547	67672	125	235			
1	m	67828	68000	172	281			
1	e	68047	68250	203	219			
1	SPACE	68187	68359	172	140			
1	y	68797	68906	109	610	2		
1	o	68968	69140	172	171			
1	u	69078	69218	140	110			
1	r	69812	69984	172	734	1		
1	SPACE	70093	70250	157	281			

WritingMode	Output	StartTime	EndTime	ActionTime	PauseTime	Pause Location	X	Y
1	s	70593	70718	125	500	2		
1	e	70859	71062	203	266			
1	c	71203	71359	156	344			
1	r	71468	71672	204	265			
1	e	71625	71812	187	157			
1	t	72062	72250	188	437			
1	s	72468	72625	157	406			
1	SPACE	72922	73062	140	454			
1	a	73250	73390	140	328			
1	n	73437	73625	188	187			
1	d	73578	73734	156	141			
1	SPACE	73687	73797	110	109			
1	a	75406	75562	156	1719	2		
1	s	75672	75843	171	266			
1	k	75797	75953	156	125			
1	SPACE	76718	76875	157	921	2		
1	m	77047	77234	187	329			
1	e	77172	77359	187	125			
1	SPACE	77312	77437	125	140			
1	y	77562	77672	110	250			
1	o	77734	77890	156	172			
1	u	77828	77968	140	94			
1	r	78281	78468	187	453			
1	SPACE	78422	78578	156	141			
1	q	79156	79281	125	734	2		
1	u	79406	79531	125	250			
1	e	79578	79656	78	172			
1	s	79718	79906	188	140			
1	t	79859	80031	172	141			
1	i	79984	80156	172	125			
1	o	80093	80265	172	109			
1	n	80203	80343	140	110			
1	s	81109	81265	156	906	1		
1	.	82047	82156	109	938	3		
1	SPACE	82312	82422	110	265			
1	O	82703	82828	125	391			
1	h	83172	83297	125	469			
1	SPACE	83390	83515	125	218			

Linear Logging File | periods of 60 seconds

Session Identification	
Filename	FirstnameLastname_1.idf
Recording Time	28/01/2007 14:39:26
Participant	1
Session	4
Pause Threshold	2000ms

Interval	Output
0:00:00	[Movement][LeftButton]The scientist[ENTER]Coldplay ·{(NUM2)[NUM0][NUM0][NUM2]}{LEFT}[UP][LEFT13][CTRL+B][RIGHT2][ENTER2]{2875}[Movement]
0:01:00	Tell me your secrets and ask me your questions. Oh lets go back to the start. Running in circles, coming up tails. Heads on a silence apart. {Movement}
0:02:00	[LeftButton][ENTER2][UP2]{2187}Come up to meet you, tell you I'm sorry. ·You don't know how lovely you are. {3110}I had to find you, tell you I need you. {3860}Tell you I set you apart
0:03:00	[BS]{2234}! {11219}[Movement][LeftButton][Movement][LeftButton][Movement][LeftButton][Movement]{4906}[LeftButton]{5906}
0:04:00	[LeftButton][Movement][ENTER2]{2329}Noboda[BS]y said it was easy, it4s [BS3]'s such a sho[BS]ame for us to part. {2890}Nobody said it was easy, no one ever said it would be this hard{4125}. {5016}Oh take me back to the start
0:05:00	. {4344}[ENTER2]{2719}[Movement][LeftButton][Movement][LeftButton][Movement]{10734}[LeftButton]{8578}[LeftButton][Movement][LeftButton][Movement][LeftButton]
0:06:00	[Movement][LeftButton]! {4594}was just gu{5515}essing at numbers and gi[BS2]figures. {5813}Pullu[BS]ing your puzzles apart. Questions of science, science and progra[BS]ess. {13578}
0:07:00	[Movement][LeftButton]{2828}[DEL]{3235}[LEFT4][DEL] Pulling your puzzles apart. ·Qeston[BS2]ions of science, sciences[BS] and progra[BS]ess.{11547} Do not p[BS]speak as I
0:08:00	oud as my heart{10797};[BS].{6281}[ENTER2]{18781}Tell me you love I[BS]me, come back and haunt me, Oh and I rush to the start
0:09:00	. Running in circles, chasing our tails. Coming back as we are. {14313}[ENTER2]{2000} Nobody said it was a[BS]eays, ·Of[BS]h it's such a shame for us to part{2093}. {2297}
0:10:00	Noby[BS]ody said it was easy, no ov[BS]ne ever said it would be w[BS]so hard. {4375}[Movement][ENTER2]I'm going back to the start.{3297}[Movement][RightButton][Movement][LeftButton]

Statistical Logging File

Session Identification	
Filename	FirstnameLastname_1.idf
Recording Time	28/01/2007 14:39:26
Participant	1
Session	4
Pause Treshold	500ms
Age	28
Sex	1
Experience	2
Group	3
task	2

Process Information	
Words	
Total Characters	761
Total Words	195
Average Word Length (WL)	3.815
Standard Deviation (WL)	1.844
Sentences	
Total Sentences	23
Average Characters/Sentence (C/S)	33.087
Standard Deviation (C/S)	17.056
Average Words/Sentence (W/S)	8.478
Standard Deviation (W/S)	5.133
Paragraphs	
Total Paragraphs	6
Average Characters/Paragraph (C/P)	126.833
Standard Deviation (C/P)	115.390
Average Words/Paragraph (W/P)	32.500
Standard Deviation (W/P)	27.970
Average Sentences/Paragraph (S/P)	3.833
Standard Deviation (S/P)	3.371

Process Time	
General	0:10:56
Number of Segments	193
Total Process Time (in seconds)	656.703
Average Process Time	3.403
Standard Deviation (Process Time)	5.453
Total Pause Time	0:05:32
Number of Pauses	192

Total Pause Time (in seconds)	332.582
Average Pause Time	1.732
Standard Deviation (Pause Time)	2.559
Active Writing Time	0:05:24
Total Writing Time (in seconds)	324.121
Average Writing Time (WT)	1.679
Standard Deviation (WT)	4.278

Writing Mode	
Keyboard	0:07:18
Total Time Keyboard (in seconds)	438.343
Number of Clusters Keyboard (CK)	7
Average Time (CK)	62.620
Standard Deviation (CK)	58.180
Number of Segments Keyboard (SK)	180
Average Time (SK)	2.435
Standard Deviation (SK)	2.488
Switches Keyboard to Mouse	7
Mouse	0:03:38
Total Time Mouse (in seconds)	218.360
Number of Clusters Mouse (CM)	8
Average Time (CM)	27.295
Standard Deviation (CM)	20.694
Number of Segments Mouse (SM)	13
Average Time (SM)	16.797
Standard Deviation (SM)	13.227
Switches Mouse to Keyboard	7

Pause Logging File

Session Identification	
Filename	FirstnameLastname_1.idf
Recording Time	28/01/2007 14:39:26
Participant	1
Session	4
Pause Treshold	1000ms

General Information	
Total Process Time	0:10:56
Interval Length	0:01:05
Interval Length (in seconds)	65.671
Total Number Of Pauses	84
Total Pause Time	0:04:17
Total Pause Time (in seconds)	257.625
Total Mean Time (MT)	3.067
Standard deviation (MT)	3.441

Pause Location	
Within Words (1)	
Number of Pauses (WW)	6
Mean Pause Time (WW)	1.945
Standard Deviation (WW)	1.766
Between Words (2)	
Number of Pauses (BW)	19
Mean Pause Time (BW)	1.961
Standard Deviation (BW)	2.282
Between Sentences (3)	
Number of Pauses (BS)	31
Mean Pause Time (BS)	3.001
Standard Deviation (BS)	3.280
Between Paragraphs (4)	
Number of Pauses (BP)	14
Mean Pause Time (BP)	4.679
Standard Deviation (BP)	5.302
Initial Pauses (5)	
Number of Pauses (IP)	0

Mean Pause Time (IP)	
Standard Deviation (IP)	
End Pauses (6)	
Number of Pauses (EP)	1
Mean Pause Time (EP)	3.297
Standard Deviation (EP)	0.000
Undefined Pauses (7)	
Number of Pauses (UP)	13
Mean Pause Time (UP)	3.604
Standard Deviation (UP)	3.089

Summary per Interval	
Interval 1	
Number of Pauses (I1)	10
Mean Pause Time (I1)	1.470
Standard Deviation (I1)	0.562
Interval 2	
Number of Pauses (I2)	7
Mean Pause Time (I2)	1.449
Standard Deviation (I2)	0.404
Interval 3	
Number of Pauses (I3)	12
Mean Pause Time (I3)	2.490
Standard Deviation (I3)	2.898
Interval 4	
Number of Pauses (I4)	4
Mean Pause Time (I4)	3.621
Standard Deviation (I4)	2.139
Interval 5	
Number of Pauses (I5)	8
Mean Pause Time (I5)	4.090
Standard Deviation (I5)	2.994
Interval 6	
Number of Pauses (I6)	6
Mean Pause Time (I6)	4.490
Standard Deviation (I6)	2.864
Interval 7	
Number of Pauses (I7)	12

Mean Pause Time (I7)	2.745
Standard Deviation (I7)	3.474
Interval 8	
Number of Pauses (I8)	7
Mean Pause Time (I8)	7.279
Standard Deviation (I8)	6.775
Interval 9	
Number of Pauses (I9)	11
Mean Pause Time (I9)	2.642
Standard Deviation (I9)	3.884
Interval 10	
Number of Pauses (I10)	7
Mean Pause Time (I10)	2.259
Standard Deviation (I10)	1.200

9 | Summary

In this thesis we describe the organization of speech recognition based writing processes. In the field of writing research, speech recognition is a new writing instrument that may cause a shift in writing process research because the underlying processes are changing. In addition to this, we take advantage of one of the weak points of speech recognition, namely the misrecognitions in the text. As it is, speech recognition visualizes how writers deal with errors in the text produced so far. In addition, speech recognition may cause a different view on writing processes because the processes are easier to reveal.

This thesis consists of 7 articles. These 7 articles are related to four main research projects. Each project is described in a section. The first section is about the influence of speech recognition on the adaptation processes and writing processes of novice speech recognition users (chapters 2, 3, and 4). In section 2 we describe the research project on error correction strategies in isolated sentences (chapters 5 and 6). The third section describes a research project on error correction strategies of professional speech recognition users (chapter 7). Finally in section 4, we wind up this thesis with an article on the logging program Inputlog (chapter 8).

Section I

Adaptation and writing processes of novice speech recognition users in their professional working environments

This section describes the adaptation and learning processes of professional writers who have started using speech recognition systems to write their professional business texts (10 lawyers and 10 academics). The data we discuss in this study are taken from a larger set of observations we collected in the context of a research project. In this research project we observed the writing processes of twenty participants five times during a period of about one month. During these observation sessions the different tasks which the participants performed were job-related and part of their normal writing activities, for example letters, e-mails or reports. This resulted in 100 observation sessions of about 20 minutes.

The method of the research project can be described as follows. Before the participants started using speech recognition for the first time they watched an introductory video about the use of the speech technology program. This video was provided by the software company. The participants were asked to use the speech recognition system during their day-to-day work for at least three hours a week. They could decide for themselves how to use the software and they were not restricted to the exclusive use of speech input. Keyboard & mouse could also be used as complementary input devices. In total we observed the participants five times while they were writing in their own environments, respectively after 1, 3, 6, 9 and 12 hours of working with speech recognition.

The process data were collected using an online camera (Camtasia™) and a sound recorder (Quickrecord™). Because of the combination of the different input modes (keyboard, mouse and speech) we were not able to use existing logging programs. We observed the participants during each writing session and took notes about specific writing circumstances that could not be registered in any other way. These recordings and notes enabled us to reconstruct the writing process in detail.

In this first section we approach the collected data from three different perspectives: adaptation strategy, learning style and previous (classical) dictating experience. Chapter 2 focuses mainly on the adaptation strategies and learning styles of writers with the same background (lawyers that already had previous dictating experience). Chapter 3 extensively takes the difference between previous dictating experiences into account and focuses on the adaptation processes and writing processes. Chapter 4 focuses on the error correction strategies of novice speech recognition users.

Summary of chapter 2

How do writers adapt to speech recognition software? The influence of learning styles on writing processes in speech technology environments

This chapter describes a case study in which we analyzed the learning processes of two writers with similar writing experience. We selected two lawyers who had the same experience level in classical dictating. However, both writers self-reported a different learning style: accommodator versus diverger style (based on a taxonomy designed by Kolb). So, both participants only had to adapt to the speech software because they were experienced in working with computers and dictating devices.

According to Kolb, accommodators score high on active experimentation and concrete experience. His model states that people with an accommodative orientation tend to solve problems in a trial-and-error manner. Divergers score high on concrete experience and reflective observation. In this learning style the emphasis is on adaptation by observation rather than by action.

The lawyers that participated in the case study differed mainly on (a) the amount of time they spent in the speech recognition mode, and (b) the mode they used to solve 'technical problems' caused by the speech recognition software. The results show that both participants adapt differently to the new writing mode, The adaptation process develops differently and seems to be driven by the learning styles of the participants. The accommodator explored the new writing mode actively and then - after trial-and-error - opted for a selective use of the speech recognizer (mainly formulation, as in traditional dictating). The diverger systematically explored the possibilities the speech recognition mode and continued adapting to the writing mode.

Summary of chapter 3

Writing with speech recognition: The adaptation processes of professional writers with and without dictating experience

In this chapter we describe the data from a quantitative perspective (10 participants) and in a case study (2 participants).

We chose to observe two different groups of writers: lawyers and academics. Most of the lawyers that participated in the study were used to dictating their texts with classical dictating devices. The academics on the other hand were not familiar with dictating. This difference in writing preferences enabled us to take into account previous dictating experience in the description of the adaptation processes.

In order to describe different aspects of the adaptation and writing processes of the participants we developed a categorization model. The model takes the complexity of the hybrid speech recognition writing mode into account and makes the enormous amount of process data accessible for further research. We paid attention to the multimodal aspect of the writing process, this is the use of different writing modes.

Next to this we have taken the problem-solving strategies of the participants using speech recognition into account (we use the term 'repair' to refer to technical problems caused by the speech recognizer and revisions which are content related). In addition we developed a transcription model to represent the multimodal writing process in a multi-layered linear representation.

The quantitative analysis of the use of the writing mode shows that those participants who had no previous dictating experiences, tend to use the voice input more extensively, both for formulating and reviewing. The group without dictating experience develops a pattern in which they steadily switch more between writing modes, both in the first and the second writing half. The group with dictating experience, on the other hand, switches less in the second writing half and also makes less use of the speech mode in that stage of the writing process.

This result is explored and further confirmed in the more detailed case analysis. The other analyses in the case study – i.e. repair, revision, and pause analysis – refine the differences in the organization of the writing process between the writers and show that the speech recognition mode seems to create a writing environment that is open for different writing profiles. The writer with previous dictating experience maintained the writing habits he had developed by using traditional dictating devices, while the writer without previous dictating experience stuck to his word processing writing style and relied heavily on the visual feedback of his dictation that appeared as typed text on the screen. In other words, the speech recognition mode itself does not seem to trigger writers into adapting a specific writing style, as opposed to what happens when writers have to adapt to writing their texts in either the dictating or the word processor mode.

Summary of chapter 4

Repair strategies in writing with speech recognition: The effect of experience with classical dictating

This chapter describes the same research project as chapter 2 and 3, but again from a different perspective. In this chapter we focus on repair strategies of writers who have started using speech recognition systems to write business texts. The writers differed in their previous writing experience. They either had classical dictating experience or they were used to writing their texts with a word processor.

The study confirms the potential hybrid character of speech recognition as a writing mode. One of the main differences between classical dictating and dictating with speech recognition is that the writer gets the dictated text displayed on the screen almost immediately, making it directly accessible in a word processor. As previous research has shown, the interaction with 'the text produced so far' is a crucial aspect in the organization of the writing process. The case study presented here also lends support to this idea. The extent to which the developing text influences the writing process may depend upon the extent to which writers make use of it while dictating.

The writers' interaction with the text produced so far is also influenced by errors that occur in the representation of the dictated text, e.g. due to (technical) misrecognition by the software.

In the case study described in this chapter, we observed that the interaction with the imperfect text on the screen can either lead to a highly recursive writing process in which every error is repaired almost immediately, or it can lead to a less recursive writing process in which repairs are made at the end of a section or when revising the first draft. One of the main differences in the writing processes of both writers seems to be related to the mode monitoring in combination with the switching behavior. Although the amount of switches is comparable for both writers, the strategies to control the writing mode seem to be different. On the one hand, we can conclude that there is no direct effect of repairs on the monitoring of writing modes, because repairs do not directly lead to mode switches. On the other hand, we can conclude that both participants have developed other error correction strategies: the experienced dictator *ignores* errors that appear in the text produced so far and *postpones* repairing errors to a later stage; the writer without previous dictating experience sometimes *anticipates* on errors and switches writing modes to avoid errors. Both writers also differ in the amount and type of errors they solve immediately. Speech recognition seems to create a writing environment that is open to different writing styles. The interaction with the imperfect text produced so far seems a decisive characteristic of speech recognition based writing processes.



Section II Error correction strategies in isolated contexts

In section 2 we describe the research project on error correction strategies in isolated sentences. Error analysis involves detecting and correcting discrepancies between the text produced so far (TPSF) and one's mental representation of what the text should be. In the study presented in chapter 5, we hypothesized that error correction with speech – like while dictating with speech recognition software – differs from keyboard as a delivery system for two reasons. It takes place auditory but it also produces different kinds of typical errors. The study measured effects of (1) presentation mode, auditory or visual-tactile, (2) error span, whether the error spans more or less than two characters, (3) mode of writing, whether text has been generated by speech recognition or keyboard, and (4) lexicality, whether the text error comprises an existing word. The analyses performed in chapter 5 are based on aggregated data. In chapter 6 we describe a more detailed re-analysis of the data via multilevel analysis.

Summary of chapter 5

The effect of errors in the text produced so far: Strategy decisions based on error span, input mode, and lexicality

The purpose of the research in this chapter is to isolate the effects of writing mode (keyboard based word processing vs. speech recognition) from various error types. That is, texts were offered only visually or also via an auditory channel. Via this approach we want to extend the work on error correction by describing the cognitive effort and error correction strategies related to different types of naturally-occurring online errors. Error analysis involves detecting, diagnosing and correcting discrepancies between the text produced so far (TPSF) and one's mental representation of what the text should be. We assume that working memory makes a substantial contribution to the primary task of error detection and text completion and that it competes for resources with the secondary task of responding to an auditory probe.

By comparing error correction strategies we can then determine the relative contributions of cognitive effort from several sources, error span, input mode, and lexicality. We assume that the cognitive effort will differ for error types. By comparing error types we can determine the effect of these error types on the working memory. For this purpose we used a series of online measures: preparation time (the time it takes to position the cursor to continue either with production or correction), immediate or delayed error correction, production time, interference reaction time, and accuracy.

The experimental condition, in which the TPSF was either offered only visually, or also through read-aloud before the visual prompt, influences the writer's strategies during error analysis. In the speech condition writers delay error correction more often and start writing sooner than in the non-speech condition. In other words, writers more often opt to prioritize text production when speech is present. In general the addition of speech reduces the cognitive demands of writing: shorter preparation and reaction times.

Error span has a rather consistent effect on strategy choice. The effect is especially powerful when interacting with the speech condition. On the one hand, large errors lead to longer preparation and production times as well as slower interference reaction times, indicating that they consume more working memory resources. On the other hand, they produce a higher rate of error analysis success than small errors. The pausing behavior of writers provides extra evidence for the complexity of large errors. A text that is not preceded by speech and that contains large errors leads to the highest accuracy in error correction. The pausing times were longer before large errors, indicating that it took writers more effort to correct them, however when the error correction of the large errors was delayed the writers could easily return to the error, probably because the errors were so obvious to find. The correction was more difficult in this situation, perhaps because retrieving the correct information about the context is a highly demanding activity.

Summary of chapter 6

Isolating the effects of writing mode from input mode, error span, and lexicality when correcting text production errors: A multilevel analysis

In this chapter we reanalyze the data from the TPSF experiment from a hierarchical perspective. A multilevel approach increases the statistical power in comparison with the unilevel statistical analyses we presented in chapter 5. It provides a sound basis to verify whether certain participants' and sentence characteristics might have disturbed our interpretation of the effect of condition and error type on the writer's interaction with the TPSF. A multilevel analysis was conducted to take into account the hierarchical nature of the data. For each variable (interference reaction time, preparation time, production time, immediacy of error correction, and accuracy of error correction), multilevel regression models are presented. By doing this, we take into account possible disturbing person characteristics while testing the effect of the different conditions and error types at the sentence level.

When comparing the resulting errors as manipulated variables, one finds that it is the auditory property of the speech, the input mode itself, that frees resources for the primary task of writing, both when completing correct representations of TPSF clauses and non-correct clauses. Large errors require more cognitive effort, and they are corrected with a higher accuracy than small errors. The latter also holds for small errors that result in non-existing words.



Section III

Error correction strategies of professional speech recognition users writing business texts

Summary of chapter 7

The text produced so far in business texts: Error correction strategies of professional speech recognition users

One of the challenges in writing research in general is to explain the structural variation in writing processes within and between subjects. More or less recursion has been attributed to writing experience, proficiency, task characteristics and the writing mode or medium. In this study, we focus on writers who use speech recognition as their primary tool for text production. Furthermore, we concentrate on one significant subprocess, namely revision, more specifically: error correction as one of the key

factors in determining the structural characteristics of writing processes.

We observed 10 professional writers who are experienced speech recognition users. The writers are observed while writing a business report. The focus of the study is on error correction strategies. We provide 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.

In this study we have combined converging research methods. The observation methods that we used are: (1) product analysis (the final product - a report - is analyzed on various levels), (2) process analysis (the data of the writing process are logged by the logging program Inputlog, the speech recognizer Dragon Naturally Speaking 8.1, and the usability program Morae), and (3) protocol analysis (the participants were asked questions during a stimulated retrospective interview).

The results are described on three levels: overall level, subgroup level (writer groups) and individual level (case study). The opposition between immediate and delayed error correction is quite decisive for the way in which writers structure their writing processes. Next to this, the distinction between technical problems and revisions plays also an important role. Most writers prefer solving technical problems immediately. The same does not necessarily hold for revisions. However, the strategy to postpone errors is not equivalent to postponing revisions. The overall results show three distinct patterns of error correction. First, there are writers who prefer writing a first time final draft and solve technical problems immediately as well as revising the text produced so far immediately (handle profile). Second, writers who solve more than half of the deficiencies in the text produced so far immediately, but who also delay or postpone various technical problems and revisions (postpone revisions profile). Finally, writers who prefer delaying error correction and who delay technical problems to a 2nd draft (postpone technical problems profile). These error correction strategies are illustrated and further elaborated upon in the case study.

IV | Section IV Logging writing processes in a Windows environment

Summary of chapter 8

Inputlog: A research tool to observe and analyze multimodal writing processes in a Windows environment

The use of computers as writing instruments has not only had a profound effect on the writing practice and the attitudes towards writing, it has also created new possibilities

for writing research. In the field of cognitive writing research especially, keystroke logging programs have become very popular. In this chapter we describe a logging program, called Inputlog. Inputlog is a logging tool that enables researchers to log the input of a writing session in a Microsoft Windows™ environment.

Inputlog 2.0 Beta 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 'integrate' module that allows merging with other process data; (4) a 'play' module that enables researchers to review the writing session. In this chapter we describe the technical and functional characteristics of Inputlog 2.0 Beta and give advice on applications of Inputlog as a research tool. We also describe methods to analyse the logged data from various perspectives (e.g. factor analysis and progression analysis).

The most distinguishing characteristics of Inputlog to date are its word processor independent functionality (it operates in a Windows environment), the parsing technology (input and process components are separated to provide a modular – high-speed – system), the standard XML structure of the output (enables merging between XML structured data of various programs) and the logging of speech recognition (Dragon Naturally Speaking 8.1).

We conclude the chapter with a preview of the plans for further developments in four niches: revision analysis, subtitling via speech recognition, evaluating typing tests and searchable output files.

References

- Adams, J. W., Hitch, G., & Hutton, U. (1997). Working memory and children's mental addition. *Journal of Experimental Child Psychology*, 67, 21-38.
- Aho, A., Sethi, R., & Ullman, J. (1986). *Compilers: Principles, Techniques and Tools*. Reading, Massachusetts: Addison-Wesley
- Alamargot, D., Dansac, C., Ros, C., & Chuy, M. (2005). Rédiger un texte procédural à partir de sources: Relations entre l'empan de production écrite et l'activité oculaire du scripteur. In D. Alamargot, P. Terrier & J. M. Cellier (Eds.), *Production, compréhension et usage des écrits techniques au travail* (pp. 51-68). Toulouse: Octarès.
- 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.
- Baddeley, A. D. (1986). *Working memory*. Oxford: Oxford University Press.
- Baddeley, A. D., & Hitch, G. (1974). Working memory. In G. A. Bower (Ed.), *Recent advances in learning and motivation* (Vol. 8, pp. 47-90). New York: Academic Press.
- Bangert-Drowns, R. L. (1993). The word processor as an instructional tool: A meta-analysis of word processing in writing instruction. *Research in the Teaching of English*, 63(1), 69-93.
- Bereiter, C., & Scardamalia, M. (1987). *The psychology of written composition*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Bernstein, L. (1990). *Developing an adequately specified model of state level student achievement with multilevel data*. Paper presented at the American Educational Association
- Blau, S. (1983). Invisible writing: Investigating cognitive processes in writing. *College, Composition and Communication*, 34, 297-312.
- Bourdin, B., & Fayol, M. (1994). Is written language production more difficult than oral language production? A working memory approach. *International Journal of Psychology*, 29, 591-620.
- 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.
- Caporossi, G., Alamargot, D., & Chesnet, D. (2004). Using the computer to study the dynamics of handwriting processes. *Lecture Notes in Computer Science*, 3245(242-254).
- Chenoweth, N. A., & Hayes, J. R. (2003). The Inner Voice in Writing. *Written communication*, 20(1), 99-118.
- Crossman, W. (2004). *VIVO. [Voice-In/Voice-Out] The coming age of talking computers*. Oakland: Regent Press.
- Daiute, C. A. (1986). Physical and cognitive factors in revising: Insight from studies with computers. *Research in the Teaching of English*, 20, 141-159.
- Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behaviour*, 19, 450-466.
- De Maeyer, S., & Rymenans, R. (2004). *Onderzoek naar kenmerken van effectieve scholen: Kritische factoren in een onderzoek naar schooleffectiviteit in het technisch en beroepssecundair onderwijs in Vlaanderen. [Study of the characteristics of effective schools: Critical factors in school effectivity of secondary education in Flanders]* University of Antwerp, Antwerp.

- Donnelly, C., & Stallman, R. (1995). *Bison: The YACC-compatible parser generator. Version 1.25*.
- Ericsson, K. A., & Kintsch, W. (1995). Long-Term Working Memory. *Psychological Review*, *102*(2), 211-245.
- Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, *87*, 215-251.
- Faigley, L., & Witte, S. (1981). Analyzing revision. *College Composition and Communication*, *32*, 400-414.
- Feng, J., Karat, C., & Sears, A. (2005). How productivity improves in hands-free continuous dictation tasks: Lessons learned from a longitudinal study. *Interacting with computers*, *17*, 265-289.
- Field, A. (2005). *Discovering statistics using SPSS*. London/Thousand Oaks/New Dehli Sage Publications.
- Fisk, A. D., Derrick, W. L., & Schneider, W. (1986-87). A methodological assessment and evaluation of dual-task paradigms. *Current Psychology Research & Reviews*, *5*, 315-327.
- 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. (1981). A cognitive process theory of writing. *College Composition and Communication*, *32*, 365-387.
- Flower, L., & Hayes, J. R. (1985). 'Talking about protocols'. *College Composition and Communication*, *37*, 16-54.
- Flower, L., Hayes, J. R., Carey, L., Schriver, K., & Stratman, J. (1986). Detection, diagnosis and the strategies of revision. *College, Composition and Communication*, *37*, 16-55.
- Galbraith, D. (1996). Self-monitoring, discovery through writing and individual differences in drafting strategy. In G. Rijlaarsdam, H. Van den Bergh & M. Couzijn (Eds.), *Theories, Models and Methodology in Writing Research* (Vol. 1, pp. 121-141). Amsterdam: Amsterdam University Press.
- Galbraith, D. (1999). Writing as a knowledge constituting process. In M. Torrance & D. Galbraith (Eds.), *Knowing what to Write: Conceptual Processes in text Production* (pp. 139-160). Amsterdam: Amsterdam University Press.
- Galbraith, D., & Torrance, M. (2004). Revision in the context of different drafting strategies. In L. Allal, L. Chanquoy & P. Largy (Eds.), *Revision: Cognitive and Instructional Processes* (pp. 63-86). Dordrecht: Kluwer Academic Publishers.
- Galbraith, D., Torrance, M., & Hallam, J. (in preparation). Effects of writing on conceptual coherence. University of Staffordshire.
- Goldberg, A., Russell, M., & Cook, A. (2003). The effect of computers on student writing: A meta-analysis of studies from 1992 to 2002. *Journal of Technology, Learning, and Assessment*, *2*(1), 1-24.
- Goldstein, H. (1995). *Multilevel statistical analysis*. London: Edward Arnold.
- Gould, J. D. (1978). How experts dictate. *Journal of Experimental Psychology: Human Perception and Performance*, *4*(4), 648-661.
- Gould, J. D. (1981). Composing letters with computer-based text editors. *Human Factors*, *23*(5), 593-606.
- Gould, J. D., & Alfaro, L. (1984). Revising documents with text editors, hand-writing recognition systems and speech-recognition systems. *Human Factors*, *26*(4), 91-406.
- Grabowski, J. (1996). Writing and speaking: Common grounds and differences towards a regulation theory of written language production. In C. M. Levy & S. E. Ransdell (Eds.), *The science of writing: Theories, methods, individual differences, and applications* (pp. 73-92). Mahwah, NJ: Lawrence Erlbaum Associates.
- 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.
- Haas, C. (1989a). Does the medium make the difference? Two studies of writing with pen and paper and with computers. *Human-Computer Interaction*, *10*, 149-169.
- Haas, C. (1989b). How the writing medium shapes the writing process: Effects of word processing on planning. *Research in the Teaching of English*, *23*, 181-207.
- Haas, C. (1989c). 'Seeing it on the screen isn't really seeing it': Computer writers' reading problems. In G. E. Hawisher & C. L. Selfe (Eds.), *Critical perspectives on computers* (pp. 16-29). New York: Teachers College Press.
- Haas, C. (1996). *Writing Technology: Studies on the materiality of literacy*. Mahwah, NJ: Lawrence Erlbaum.
- Hacker, D. J. (1994). Comprehension monitoring as a writing process. *Advances in Cognition and Educational Practice*, *6*, 143-172.
- Hacker, D. J. (1997). Comprehension monitoring of written discourse across early-to-middle adolescence. *Reading and Writing*, *9*(3), 207-240.
- Hacker, D. J., Plumb, C. S., Butterfield, E. C., Quathamer, D., & Heineken, E. (1994). Text revision:

- Detection and correction of errors. *Journal of Educational Psychology*, 86(1), 65-78.
- Halverson, A., Horn, D. B., Karat, C., & Karat, J. (1999). *The beauty of errors: Patterns of error correction in desktop speech systems*. Paper presented at the Human-Computer Interaction — INTERACT '99, Edinburgh.
- Hartley, J. (2007). Longitudinal studies of the effects of new technologies on writing: Two case-studies. In M. Torrance, L. Van Waes & D. Galbraith (Eds.), *Writing and Cognition: Methods and Applications* (Vol. 20, pp. 293-306). Oxford: Elsevier.
- Hartley, J., Howe, M., & McKeachie, M. (2001). Writing through time: Longitudinal studies of the effects of new technology on writing. *British Journal of Educational Technology*, 32(2), 141-151.
- Hartley, J., Sotto, E., & Pennebaker, J. (2003). Speaking versus typing: A case-study of the effects of using voice-recognition software on academic correspondence. *British Journal of Educational Technology*, 34(1), 5-16.
- Hayes, J. R. (1996). A new framework for understanding cognition and affect in writing. In C. M. Levy & S. E. Ransdell (Eds.), *The science of writing: Theories, methods, individual differences, and applications* (pp. 1-27). Mahwah: New Jersey: Lawrence Erlbaum Associates.
- Hayes, J. R., & Chenoweth, N. (2006). Is Working Memory Involved in the Transcribing and Editing of Texts? *Written Communication*, 23(2), 135.
- Hayes, J. R., & Flower, L. (1986). Writing research and the writer. *American Psychologist*, 41, 1106-1113.
- Hayes, J. R., Flower, L., Schriver, K., Statman, J., & Carey, L. (1987). Cognitive processes in revision. In S. Rosenberg (Ed.), *Reading, writing, and language possessing* (Vol. 2, pp. 176-240). Cambridge: Cambridge University Press.
- Hayes, J. R., & Hayes, 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.
- Holmqvist, K., Johansson, V., Strömquist, S., & Wengelin, Å. (2002). Studying reading and writing online. In S. Strömquist (Ed.), *The diversity of languages and language learning* (pp. 103-123). Lund: Centre for Languages and Literature, Lund University.
- Honeycutt, L. (2003). Researching the use of voice recognition writing software. *Computers and Composition*, 20, 77-95.
- Honeycutt, L. (2004). Literacy and the writing voice: The intersection of culture and technology in dictation. *Journal of Business and Technical Communication*, 18(3), 294-327.
- Hoskyn, M., & Swanson, H. (2003). The relationship between working memory and writing in younger and older adults. *Reading and Writing*, 16, 759-784.
- Hox, J. (2002). *Multilevel analysis: Techniques and applications*. Mahwah, New Jersey/London: Lawrence Erlbaum Associates Publishers.
- 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.
- Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge, MA: Harvard University Press.
- Joram, E., Woodruff, E., Lindsay, P., & Bryson, M. (1990). Students' editing skills and attitudes toward word processing. *Computers and Composition*, 7, 55-72.
- Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in Working Memory. *Psychological Review*, 99(1), 122-149.
- Karat, C., Halverson, C., Horn, D., & Karat, J. (1999). *Patterns of entry and correction in large vocabulary continuous speech recognition systems*. Paper presented at the CHI 99, Pittsburgh.
- Karat, J., Horn, D. B., Halverson, C. A., & Karat, C. (2000). *Overcoming Unusability: Developing strategies in speech recognition systems*. Paper presented at the Conference on Human Factors in Computing Systems, CHI 2000, Den Haag.
- Kellogg, R. T. (1988). Attentional overload and writing performance: Effects of rough draft and outline strategies. *Journal of Experimental Psychology: Learning, memory and cognition*, 14(2), 355-365.
- Kellogg, R. T. (1994). *The Psychology of Writing*. New York: Oxford University Press.
- Kellogg, R. T. (1996). A model of working memory in writing. In C. M. Levy & S. E. Ransdell (Eds.), *The Science of Writing: Theories, methods, individual differences and applications* (pp. 57-71). Hillsdale, NJ: Lawrence Erlbaum.
- Kellogg, R. T. (1999). Components of Working Memory in Text Production. In M. Torrance & G. Jeffery (Eds.), *The cognitive demands of writing: processing capacity and Working Memory effects in text production* (Vol. 3, pp. 43-61). Amsterdam: Amsterdam University Press.
- Kellogg, R. T. (2001). Competition for working memory among writing processes. *American Journal of*

- Psychology*, 114, 175-191.
- Kellogg, R. T. (2004). Working memory components in written sentence generation. *American Journal of Psychology*, 117, 341-361.
- Kieft, M., Rijlaarsdam, G., Galbraith, D., & Van den Bergh, H. (in preparation). The effects of students individual characteristics and writing instruction on writing performance. University of Amsterdam.
- Kieft, M., Rijlaarsdam, G., Galbraith, D., & Van den Bergh, H. (to appear). The effects of adapting a writing course to students' writing strategies. *British Journal of Educational Psychology*.
- 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*. Unpublished Lic. Thesis, Stockholm University, Stockholm.
- Larigauderie, P., Gaonac'h, D., & Lacroix, N. (1998). Working memory and error detection in texts: What are the roles of the central executive and the phonological loop? *Applied Cognitive Psychology*, 12, 505-527.
- Lee, Y. J. (2002). A comparison of composing processes and written products in timed-essay tests across paper-and-pencil and computer modes. *Assessing Writing*, 8(2), 135-157.
- Leijten, M. (2007). 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., De Maeyer, S., Ransdell, S., & Van Waes, L. (in preparation). Isolating the effects of writing mode from error span, input type, and lexicality when correcting text production errors: a multilevel analysis. University of Antwerp.
- Leijten, M., Janssen, D., & Van Waes, L. (in preparation). The text produced so far in business texts: error correction strategies by professional speech recognition users. University of Antwerp.
- Leijten, M., Ransdell, S., & Van Waes, L. (submitted). Isolating the effects of writing mode from error span, input type, and lexicality when correcting text production errors. University of Antwerp.
- Leijten, M., & Van Waes, L. (2003a). Schrijven zoals je spreekt, spreken zoals je schrijft: De invloed van spraakherkenning op het schrijfproces van dicteerders met verschillende leerstijlen [Writing as you talk, talking as you write: The influence of speech recognition on the writing process of dictators with different learning styles]. *Tijdschrift voor Taalbeheersing*, 25(4), 325-341.
- Leijten, M., & Van Waes, L. (2003b). *The writing processes and learning strategies of initial users of speech recognition: a case study on the adaption process of two professional writers* (Research Report No. 2003022). Antwerp: University of Antwerp.
- Leijten, M., & Van Waes, L. (2005a). *Inputlog: A logging tool for the research of writing processes* (Research Papers, Faculty of Applied Economics). Antwerp: University of Antwerp.
- Leijten, M., & Van Waes, L. (2005b). Writing with speech recognition: The adaptation process of professional writers with and without dictating experience. *Interacting with Computers*, 17(6), 736-772.
- Leijten, M., & Van Waes, L. (2006a). 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. (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.
- Levitt, W. J. M. (1983). Monitoring and self-repair in speech. *Cognition*, 14(1), 41-104.
- Levine, J. R., Mason, T., & Brown, D. (1992). *Lex & Yacc* (2nd ed.).
- Levy, C. M., & Marek, P. (1999). Testing components of Kellogg's multicomponent models of Working Memory in writing: The role of the phonological loop. In M. Torrance & G. Jeffery (Eds.), *The cognitive demands of writing. Processing capacity and Working Memory effects in text production*. (Vol. 3, pp. 25-41). Amsterdam: Amsterdam University Press.
- Levy, C. M., & Ransdell, S. E. (1996). Writing signatures. In C. M. Levy & S. E. Ransdell (Eds.), *The Science of Writing: Theories, Methods, Individual Differences, and Applications* (pp. 149-162). Mahwah, NJ: Lawrence Erlbaum.
- Levy, C. M., & Ransdell, S. E. (2001). Writing with concurrent memory loads. In T. Olive & C. M. Levy (Eds.), *Contemporary Tools and Techniques for Studying Writing* (pp. 9-29). Dordrecht: Kluwer Academic Publishers.
- Lindgren, E. (2002). The LS graph: A methodology for visualising writing revision. *Language Learning*, 52(3), 565-595.
- Lindgren, E. (2005). *Writing and Revising: Didactic and Methodological Implications of Keystroke Logging*. (Skrifter från moderna språk, No. 18). Umeå, Sweden: Umeå University, Department of Modern Languages.

- 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. 83-96). Oxford: Elsevier.
- Lindgren, E., & Sullivan, K. P. H. (2003). Stimulated recall as a trigger for increasing noticing and language awareness in the L2 writing classroom: A case study of two young female writers. *Language Awareness, 12*, 172-186.
- Lindgren, E., & Sullivan, K. P. H. (2006a). Analysing on-line revision. In K. P. H. Sullivan & E. Lindgren (Eds.), *Computer Keystroke Logging: Methods and Applications* (Vol. 18, pp. 157-188). Oxford: Elsevier.
- Lindgren, E., & Sullivan, K. P. H. (2006b). 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.
- 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. (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.
- Matsuhashi, A. (1987). Revising the plan and altering the text. In A. Matsuhashi (Ed.), *Writing in real time: Modelling production processes* (pp. 197-223). New York: Academic Press.
- Mazeland, H. (2003). *Inleiding in de conversatieanalyse [Introduction to conversation analysis]*. Bussum: Coutinho.
- McCutchen, D. (1996). A capacity theory of writing: Working memory in composition. *Educational Psychology Review, 8*(3), 299-325.
- 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. (2004). Memory in writing: Empirical evidences from the dual-task technique working. *European Psychologist, 9*(1), 32-42
- Olive, T., & Kellogg, R. T. (2002). Concurrent activation of high- and low-level production processes in written composition. *Memory and Cognition, 30*, 594-600.
- Olive, T., & Piolat, A. (2002). Suppressing visual feedback in written composition : Effects on processing demands and coordination of the writing processes. *International Journal of Psychology, 37*(4), 209-218.
- Paxson, V. (1995). *Flex: A fast scanner generator, edition 2.5*.
- Perrin, D. (2003). Progression analysis: Investigating writing strategies at the workplace. *Journal of Pragmatics, 35*(6), 907-921.
- Pilotti, M., Chodorow, M., & Thornton, K. C. (2004). Error detection in text: Do feedback and familiarity help? *The Journal of General Psychology, 131*(4), 242-266.
- Piolat, A., Roussey, J. Y., Olive, T., & Amada, M. (2004). Processing time and cognitive effort in revision: effects of error type and of working memory capacity. In L. Allal, L. Chanquoy, P. Largy & Y. Rouiller (Eds.), *Revision: Cognitive and Instructional Processes* (pp. 21-38). Dordrecht: Kluwer Academic Publishers.
- Python, (2007). Retrieved February 12, from <http://www.python.org/>
- Quené, H., & Van den Bergh, H. (2004). On Multi-Level Modeling of data from repeated measures designs: A tutorial. *Speech Communication, 43*(1-2), 103-121.
- Quinlan, T. (2002). *Speech recognition technology and the writing processes of students with writing difficulties*. Paper presented at the Earli Sig Writing 2002, Stafford, England.
- Quinlan, T. (2004). Speech recognition technology and students with writing difficulties: Improving fluency. *Journal of Educational Psychology, 96*, 337-346.
- 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.
- Quinlan, T., Loncke, M., Leijten, M., & Van Waes, L. (in preparation). Writers juggling error-correcting with sentence generation. Educational Testing Service, University of Ghent, University of Antwerp.
- Rabbitt, P. (1978). Detection of errors by skilled typists. *Ergonomics, 21*, 945-958.
- Rabbitt, P., Cummings, P., & Vyas, S. (1978). Some errors of perceptual analysis in visual search can be detected and corrected. *Quarterly Journal of Experimental Psychology, 30*, 417-427.
- Ransdell, S. E., & Hecht, S. A. (2003). Time and resource limits on working memory: Cross-age consistency in counting span performance. *Journal of Experimental Child Psychology, 86*, 303-313.
- Ransdell, S. E., & Levy, C. M. (1996). Working memory constraints on writing quality and fluency. In C. M. Levy & S. E. Ransdell (Eds.), *The science of writing: Theories, Methods, Individual Differences,*

- and Applications (pp. 93-105). Mahwah, NJ: Lawrence Erlbaum Associates.
- Ransdell, S. E., & Levy, C. M. (1999). Writing reading and speaking memory spans and the importance of resource flexibility. In M. Torrance & G. Jeffery (Eds.), *The cognitive demands of writing: processing capacity and working memory effects in text production*. Amsterdam: Amsterdam University Press.
- Ransdell, S. E., Levy, C. M., & Kellogg, R. T. (2002). The structure of writing processes as revealed by secondary task demand. *L1-Educational Studies in Language and Literature* 2(2), 141-163.
- Rashbash, J., Steele, F., Browne, W., & Prosser, B. (2004). A user's guide to MLwiN Version 2.0. Retrieved February 6 2007, 2007
- Reece, J., & Cumming, G. (1996). Evaluating speech-based composition methods: Planning, dictation, and the listening word processor. In C. M. Levy & S. E. Ransdell (Eds.), *The science of writing: Theories, Methods, Individual Differences, and Applications* (pp. 361-380). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Rodman, R. D. (1999). *Computer speech technology*. London: Artech House Publishers.
- Sadowski, M., Kealy, W. A., Goetz, E. T., & Paivio, A. (1997). Concreteness and imagery effects in the written composition of definitions. *Journal of Experimental Psychology*, 89, 518-526.
- Schegloff, E., Jefferson, G., & Sacks, H. (1977). The Preference for Self-Correction in the Organization of Repair in Conversation. *Language*, 53, 361-382.
- Schilperoord, J. (1996). *It's about time: Temporal aspects of cognitive processes in text production*. Amsterdam/Atlanta: Rodopi.
- Schilperoord, J. (2002). On the cognitive status of pauses in discourse production. In T. Olive & C. M. Levy (Eds.), *Contemporary tools and techniques for studying writing* (Vol. 10, pp. 61-88). Dordrecht/Boston/London: Kluwer Academic Publishers.
- Selzer, J. (1983). The composing process of an engineer. *College Composition and Communication*, 35, 178-188.
- Selzer, J. (1984). Exploring options in composing. *College Composition and Communication*, 35, 276-284.
- 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* (IPLab-59). 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.), *Models and Methodology in writing research* (pp. 526-541). Amsterdam: 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.), *Theories, Models and Methodology in writing research* (pp. 526-541). 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.
- Shah, P., & Miyake, A. (1996). The separability of working memory resources for spatial thinking and language processing: An individual differences approach. *Journal of Experimental Psychology: General*, 125, 4-27.
- Sharples, M. (1999). *How We Write: Writing as Creative Design*. London: Routledge.
- Sharples, M., & Pemberton, L. (1992). Representing writing: External representations and the writing process. In P. O'Brian Holt & N. Williams (Eds.), *Computers and Writing: State of the art* (pp. 319-336). Dordrecht: Kluwer Academic Publishers.
- Smagorinsky, P. (1989). The reliability and validity of protocol analysis. *Written communication*, 6, 463-479.
- Snijders, T. A. B., & Bosker, R. J. (1999). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. London/Thousand Oaks/New Dehli: Sage Publications.
- Snyder, M. (1974). Self-monitoring of expressive behavior. *Journal of Personality and Social Psychology*, 30(4), 526-537.
- 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.
- Sternberg, S. (1969). The discovery of processing stages: Extensions of Donders's method. *Acta Psychologica*, 30, 276-235.
- Stifelman, L. J. (1993). *User repairs of speech recognition errors: An intonational analysis*. Massachusetts: MIT Media Laboratory Technical Report.

- Strömquist, 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.
- Strömquist, S., & Karlsson, H. (2001). *ScriptLog for Windows: Users manual*. Lund: University of Lund.
- Strömquist, S., & Malmsten, L. (1997). *ScriptLog Pro User's manual*. Göteborg: Göteborg University, Dept of Linguistics.
- Sullivan, K. P. H., & Lindgren, E. (2006). *Computer Key-Stroke Logging and Writing*. Oxford: Elsevier Science.
- Terrel, S. R. (2002). The effect of learning style on doctoral course completion in a Web-based learning environment. *Internet and Higher Education*, 5, 345-352.
- Torrance, M., & Galbraith, D. (2006). The processing demands of writing. In C. A. MacArthur, S. Graham & J. Fitzgerald (Eds.), *Handbook of writing research* (pp. 468). New York: Guilford Publications.
- Towse, J., Hitch, G., & Hutton, U. (1998). A reevaluation of working memory capacity in children. *Journal of Memory and Language*, 39, 195-217.
- Van den Bergh, H., & Rijlaarsdam, G. (1996). The dynamics of composing: Modelling writing process data. In C. M. Levy & S. E. Ransdell (Eds.), *The science of writing: Theories, methods, individual differences, and applications* (pp. 207-232). Mahwah, NJ: Lawrence Erlbaum Associates.
- 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 den Haak, M. J., De Jong, M. D. T., & Schellens, P. J. (2004). Employing think-aloud protocols and constructive interaction to test the usability of online library catalogues: A methodological comparison. *Interacting with Computers*, 16, 1153-1170.
- Van den Haak, M. J., De Jong, M. D. T., & Schellens, P. J. (2006). Hardopdenprotocollen en gebruikersonderzoek: Volledigheid en reactiviteit van de synchrone hardopdenmethode. *Tijdschrift voor Taalbeheersing*, 28(3), 185-197.
- Van der Geest, T., Leijten, M., & Van Waes, L. (2006). Taalproductie en -verwerking onderzoeken met de computer [Researching language production and processing with the computer]. *Tijdschrift voor Taalbeheersing*, 28(3), 181-185.
- Van Someren, M. W., Barnard, Y. F., & Sandberg, J. A. C. (1994). *The think aloud method: A practical guide to modelling cognitive processes*. London: Academic Press.
- Van Waes, L. (1991). *De computer en het schrijfproces: De invloed van de tekstverwerker op het pauze- en revisiegedrag van schrijvers [The computer and the writing process: The influence of the word processor on the pausing and revision behavior of writers]*. Enschede: VMW (PhD thesis).
- 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., & 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.
- 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.
- Williams, N., Hartley, P., & Pittard, V. (2004). Talking to write: Investigating the practical impact and theoretical implications of speech recognition (SR) software on real writing tasks. In L. Van Waes, D. Galbraith & M. Torrance (Eds.), *Recent developments in writing process research* (Vol. 2): Kluwer.
- Wood, E., Willoughby, T., Specht, J., & Porter, L. (2002). An examination of how a cross-section of academics use computer technology when writing academic papers. *Computers & Education*, 38, 287-301.

NL | Samenvatting

Zouden we niet allemaal zo vlot willen schrijven als die ene columnist die we zo bewonderen, of die schrijfster bij wie we al op voorhand zitten te wachten tot haar volgende boek verschijnt? Het hoeven niet eens pareltjes van best-selling proza te zijn, voor een moeiteloos geschreven zakelijke tekst zouden we ook met alle plezier tekenen. De realiteit is: schrijven is best moeilijk. Of om het met de woorden van Kees van Kooten te zeggen: “schrijven is blijven zitten tot het er staat: schoonschrift.” (p.120)¹

Al sinds mensenheugenis proberen we schrijven zo gemakkelijk mogelijk te maken. Hiervoor worden enerzijds producten ontwikkeld die het mechanisme van schrijven vergemakkelijken, maar anderzijds wordt ook aandacht besteed aan producten die de onderliggende processen ondersteunen. Denk hierbij aan het gemak waarmee kinderen tegenwoordig met computers als schrijfinstrument omgaan. Door allerlei hulpmiddelen zoals spellingcheckers leren ze al snel spelfouten te herkennen omdat een kronkellijntje hen hierop attendeert. Het is een schril contrast met de instrumenten waarover onze voorouders konden beschikken: deze waren zeer beperkt en erg arbeidsintensief. Als we een duik in de geschiedenis nemen dan zien we dat voordat het alfabet ontstond, boodschappen in rotstekeningen werden uitgebeeld met figuren en symbolen. Het gemak waarmee we tegenwoordig aanpassingen in een tekst maken, zou ondenkbaar zijn als we nog steeds over dezelfde schrijfinstrumenten zouden beschikken. Als ik deze tekst met pen en papier zou moeten schrijven, zou ik veel beter nadenken over de precieze formulering van iedere volgende zin, terwijl nu mijn vingers regelmatig even op het toetsenbord blijven rusten, formuleringen aftasten, bedenken, bijsturen, terwijl mijn ogen gericht teruglezen om op die manier de rest van de tekst te kunnen vormen. Om het maar niet

¹ Fragment van: Kooten, K. & Spieker, E. (2003). *Letterlust*. Uitgeverij Manteau/De Harmonie Antwerpen/Amsterdam, p.120.

eens te hebben over de strategie die ik zou gebruiken, mocht ik deze tekst in een rotswand moeten beitelen. Kortom, efficiëntie en schrijfgemak zijn twee belangrijke pijlers in de evolutie van schriftelijke communicatie.

De zoektocht naar uiterste efficiëntie heeft de laatste jaren onder meer tot de ontwikkeling van spraakherkenning geleid. Spreken wordt veelal gezien als de meest natuurlijke taalmodus. In tegenstelling tot spreken, is schrijven veel moeilijker en nog steeds zijn er heel wat mensen in deze wereld die de vaardigheid nooit hebben aangeleerd. Met andere woorden: er kunnen meer mensen goed spreken, maar minder mensen goed schrijven. Historisch gezien, is geschreven taal echter noodzakelijk geworden om boodschappen te bewaren en te verspreiden. De basis is echter gesproken taal, zoals Crossman beweert:

From a Darwinian perspective, written language is a 6,000-to-10,000-year-old bridge that humanity has been using to walk from our First Golden Age of oral culture to our Second. We undertook this journey to survive as species. Six thousand to ten thousand years ago, lacking the ability to store and retrieve by memory the growing sum of survival information, our species faced two options: develop new storage-retrieval technology or self-destruct. That's when and why we created the written-language bridge. (Crossman, 2004, p. 21)

Schrijven kan dus gezien worden als een visuele representatie van gesproken taal: een combinatie die spraakherkenning ten volle benut. Bij spraakherkenning dicteren schrijvers hun teksten aan de computer en die zet deze gesproken tekst om naar geschreven tekst die vervolgens weer op het beeldscherm verschijnt. Ideaal zou je zeggen. Uitgaande van een ideale wereld waarin we met 100% accurate spraakherkenningssoftware zouden schrijven, is dit inderdaad het geval. In dit proefschrift zullen we echter zien dat schrijven met spraakherkenning nog niet altijd probleemloos verloopt en soms een nieuw soort schrijffout oplevert. De software is namelijk nog niet geavanceerd genoeg en herkent (nog) niet alle woorden die de gebruiker dicteert. Als de spraakherkenner een woord niet herkent, kiest hij een vergelijkbaar woord of vergelijkbare woorden uit het lexicon met de hoogste slaagkans binnen de geschreven context. Een voorbeeld van een typische spraakherkenningsfout is dat bijvoorbeeld het commando 'ongedaan maken' als 'ogen dan maken' op het scherm verschijnt. Het verschil met de bekende typefouten is dat de schrijver het niet (altijd) zelf 'in de hand' heeft dat iets verkeerd begrepen wordt door de computer. Dit maakt het herlezen van een tekst geschreven met spraakherkenning een uitdagende taak. Ook het kenmerk dat tekstproductie eigenlijk veel sneller gaat met spraakherkenning zorgt voor een andere focus op het corrigeren van fouten.

In dit proefschrift beschrijven we in de eerste plaats een veranderend schrijfproces. Daarnaast willen we ook een nadeel van spraakherkenning, namelijk de ongewenste fouten in de tekst, vertalen naar een voordeel voor schrijfprocesonderzoek. Spraakherkenning zorgt er immers voor dat het veel beter zichtbaar wordt hoe schrijvers met fouten omgaan.

We kunnen stellen dat spraakherkenning een nieuw medium is dat een verandering in schrijfprocesonderzoek teweegbrengt: de onderliggende cognitieve processen

van tekstproductie en -correctie veranderen. Bovendien kan spraakherkenning een ander licht werpen op schrijfprocessen, omdat ze gemakkelijker te bestuderen zijn. Met andere woorden, spraakherkenning legt 'het vel' van schrijfprocessen bloot.

Dit proefschrift bestaat uit zeven artikelen, gerelateerd aan vier onderzoeksprojecten. Ieder project wordt beschreven in een sectie. De eerste sectie gaat over de invloed van spraakherkenning op adaptatie- en schrijfprocessen van 'nieuwe' spraakherkenningsgebruikers (hoofdstuk 2, 3 en 4). In de tweede sectie beschrijven we een onderzoeksproject naar correctiestrategieën in geïsoleerde zinnen (hoofdstuk 5 en 6). De derde sectie gaat over correctiestrategieën van professionele spraakherkenningsgebruikers (hoofdstuk 7). De laatste sectie behandelt het (toets)registratieprogramma Inputlog (hoofdstuk 8).

I Sectie I

Adaptatie- en schrijfprocessen van onervaren spraakherkenningsgebruikers in hun professionele schrijfomgeving

De eerste sectie van dit proefschrift beschrijft de adaptatie- en leerprocessen van professionele schrijvers die spraakherkenning leren gebruiken om hun zakelijke teksten te schrijven. De data die we in de verschillende hoofdstukken bespreken, zijn afkomstig uit een grotere dataset. In een overkoepelend onderzoeksproject hebben we de schrijfprocessen van 20 participanten gedurende een periode van een maand op 5 verschillende momenten geobserveerd (10 advocaten en 10 academici). Tijdens deze observatiesessies voerden de participanten verschillende schrijftaken uit die deel uitmaakten van hun dagelijkse taken (o.a. e-mails, rapporten, brieven).

Voordat de participanten voor de eerste keer met de spraakherkenningssoftware werkten, kregen ze een inleidende video te zien over het gebruik van spraakherkenning. Alle deelnemers werd gevraagd om het spraakherkenningsprogramma te gebruiken tijdens hun dagelijkse schrijfactiviteiten, en dit gedurende minstens 3 uur per week. Er werden hun geen verdere beperkingen over het gebruik opgelegd; ze konden dus spraakherkenning gebruiken in combinatie met toetsenbord en muis. In totaal werden alle deelnemers vijfmaal geobserveerd in hun eigen werkomgeving terwijl ze aan het schrijven waren, respectievelijk nadat ze ongeveer 1, 3, 6, 9 en 12 uren geschreven hadden met spraaktechnologie. Elke observatie nam ongeveer 20 tot 30 minuten in beslag.

De data werden verzameld door gebruik te maken van een onlinecamera (Camtasia™) en een geluidsrecorder (Quickrecord™). Omdat we hier te maken hadden met een combinatie van verschillende inputmodi (toetsenbord, muis en spraak) was het niet mogelijk om bestaande loggingprogramma's te gebruiken. Om het gebruikersge-

drag te registreren dat niet via de online-camera en de geluidsrecorder vastgelegd kon worden, maakten we handmatig aanvullende notities. Die combinatie van registratiemethodes stelde ons in staat het schrijfproces gedetailleerd te reconstrueren.

In deze eerste sectie benaderen we de data vanuit drie perspectieven: adaptatiestrategie, leerstijl en ervaring met klassieke dicteerapparatuur. Hoofdstuk 2 is gericht op de adaptatie- en leerstrategieën van schrijvers met dezelfde achtergrond (advocaten met ervaring in klassiek dicteren). Hoofdstuk 3 neemt het verschil in dicteerveraring als vertrekpunt en is gericht op het adaptatie- en schrijfproces. Hoofdstuk 4 gaat dieper in op de correctiestrategieën van onervaren spraakherkenningsgebruikers.

Samenvatting hoofdstuk 2

Hoe passen schrijvers zich aan spraakherkenningssoftware aan? De invloed van leerstijlen op schrijfprocessen met spraakherkenning

In hoofdstuk 2 concentreren we ons op een casestudy waarin we het schrijfproces van twee ervaren dicteerders beschrijven. We hebben deze schrijvers gedurende enkele weken geobserveerd om na te gaan hoe ze zich in die initiële leerperiode aanpassen aan de nieuwe technologie. Beide participanten zijn advocaat en ze waren bij de start van het onderzoek al bekend met het dicteren van teksten in een traditionele omgeving (dictafoon en secretaresse). Dit betekent dat ze beiden vertrouwd waren in het sprekend schrijven en alleen met het spraakprogramma moesten leren werken. De schrijvers verschillen qua leerprofiel (gebaseerd op een vragenlijst van Kolb): *accommodator* (doener) en *divergeerder* (dromer).

De sterkste vaardigheden van een accommodator zijn actief experimenteren en leren van concrete ervaringen. Dit betekent dat iemand met deze leerstijl veel intuïtief zal proberen en aan het resultaat zal merken of de methode werkt of niet. Daar zal hij zijn gedrag dan weer op afstemmen. Een ander gedragskenmerk van mensen met deze leerstijl is dat ze ongeduldig zijn in het proberen van nieuwe dingen. Divergeerders daarentegen leggen bij het leren de nadruk op concrete ervaringen en reflectieve observatie. Ze passen zich eerder aan een nieuwe situatie aan door deze eerst te observeren dan door meteen actie te ondernemen. Divergeerders bedenken graag alternatieve oplossingen. Daarnaast zijn mensen met deze leerstijl vaak fantasierijk (Kolb, 1984).

De advocaten in dit onderzoek verschilden vooral in (a) de tijd dat ze effectief schrijven met spraakherkenning, en (b) de modus die ze gebruiken om 'technische problemen' veroorzaakt door de spraakherkenningssoftware te corrigeren. Uit de case-study blijkt dat precies die verschillen doorslaggevend zijn in het adaptatieproces. De 'accommodator' handhaaft in hoge mate zijn vertrouwde schrijfstrategie en exploreert slechts beperkt de mogelijkheden van de nieuwe modus; de 'diver-

geerder' daarentegen exploreert veel explicieter het hybride karakter van de nieuwe dicteermodus en past gaandeweg zijn initiële schrijfgedrag aan.

Samenvatting hoofdstuk 3

Schrijven met spraakherkenning: De invloed van spraakherkenning op het schrijfproces van schrijvers met en zonder dicteerervaring

In dit hoofdstuk beschrijven we de data vanuit een kwantitatief perspectief (10 participanten) en in een case studie (2 participanten).

We hebben ervoor gekozen om twee verschillende schrijversgroepen te observeren: advocaten en wetenschappers. De meeste advocaten waren gewend om hun teksten te dicteren op de klassieke manier. Voor de wetenschappers was het gebruik van spraakherkenning de eerste kennismaking met dicteren. Dit verschil in ervaring hebben we als basis genomen om de adaptatieprocessen te beschrijven.

Om de verschillende aspecten van het adaptatie- en schrijfproces te beschrijven hebben we een analysemodel opgesteld. Het model houdt rekening met de complexiteit van de hybride schrijfmodus en maakt de enorme hoeveelheid procesdata toegankelijk. We hebben rekening gehouden met de verschillende schrijfmodi die tijdens het schrijven gebruikt worden. Het model is voornamelijk opgesteld om het 'repair'-proces van de participanten te beschrijven. 'Repairs' verwijzen hier naar 'problemen' en 'revisies'. In dit onderzoek verstaan we onder problemen: technische problemen die gerelateerd zijn aan de spraakherkenner, met name die gevallen waarbij de spraakinput niet leidt tot datgene wat de schrijver voor ogen had. Revisies zijn in deze studie met name gedefinieerd als veranderingen in een tekst na evaluatie. Die veranderingen zijn niet veroorzaakt door de spraakherkenner, maar zijn bedoeld om de inhoud, de formulering of de lay-out van een tekst aan te passen. Om het multimodale schrijfproces in een lineaire representatie weer te kunnen geven, hebben we een transcriptiemodel opgesteld.

Een kwantitatieve analyse van het gebruik van de schrijfmodi toont dat de participanten zonder ervaring in klassiek dicteren veel meer spraak gebruiken, zowel voor het formuleren van de tekst als voor het reviseren ervan. De groep zonder dicteerervaring ontwikkelt een patroon waarbij ze in de eerste en de tweede helft van het schrijfproces veel wisselt tussen schrijfmodi. De groep met dicteerervaring daarentegen wisselt in de tweede helft van het schrijfproces veel minder tussen verschillende modi en ze maakt ook veel minder gebruik van spraak in dit stadium van het schrijfproces.

Dit resultaat is gedetailleerder bekeken in een case studie. De aanvullende analyses in de case studie, zoals een repair- en pauze-analyse, verfijnen de verschillen in de organisatie van schrijfprocessen tussen de verschillende schrijvers. Concluderend kunnen we zeggen dat de resultaten van deze case studie suggereren dat spraak-

herkenning een open schrijffomgeving creëert voor verschillende schrijfstijlen. De schrijver met dicteerervaring hield vast aan de gewoontes die hij had ontwikkeld bij het schrijven met de klassieke transcriptie-apparatuur. De andere schrijver, zonder dicteerervaring, hield vast aan de gewoontes van de typische tekstverwerkergebruiker: veel recursieve bewegingen in de tekst en een grote afhankelijkheid van de visuele feedback op het scherm tijdens zijn schrijfproces. Met andere woorden, de spraakherkenningmodus zelf lijkt geen specifieke schrijfstijl op te leggen in tegenstelling tot wat er gebeurt als schrijvers die gewend zijn met pen en papier te schrijven zich moeten aanpassen aan andere schrijfmodi, zoals klassiek dicteren of schrijven met een tekstverwerker.

Samenvatting hoofdstuk 4

Correctiestrategieën in schrijfprocessen met spraakherkenning: Het effect van ervaring met klassiek dicteren

Hoofdstuk 4 beschrijft de resultaten van hetzelfde onderzoeksproject als hoofdstuk 2 en 3, maar vanuit een ander perspectief. In dit hoofdstuk richten we ons op 'repair'-strategieën van schrijvers die voor het eerst zakelijke teksten schrijven met spraakherkenningsoftware. De schrijvers verschilden in dicteerervaring: de ene groep had wel dicteerervaring en de andere groep had alleen ervaring met schrijven met de tekstverwerker.

Het onderzoek bevestigt het potentieel hybride karakter van spraakherkenning. Eén van de grote verschillen tussen klassiek dicteren en dicteren met spraakherkenning is dat in het laatste geval de gedicteerde tekst in een tekstverwerkingsprogramma op het scherm verschijnt, waardoor de tekst direct toegankelijk wordt voor de schrijver. Eerder onderzoek heeft aangetoond dat de 'reeds geproduceerde tekst' van cruciaal belang is voor de organisatie van het schrijfproces. De case studie bevestigt de invloed van de tekst op het scherm. De mate waarin de tekst op het scherm het schrijfproces beïnvloedt, is afhankelijk van de mate waarin de schrijvers er gebruik van maken tijdens het dicteren. De interactie met de tekst wordt ook bepaald door de fouten die voorkomen in de tekstrepresentatie, bijvoorbeeld door een technische herkenningfout van de software.

De case studie in dit hoofdstuk beschrijft de interactie van de schrijvers met de incorrecte tekst op het scherm. Dit kan enerzijds leiden tot een zeer recursief schrijfproces, waarbij bijna elke fout onmiddellijk opgelost wordt. Of het kan leiden tot een minder recursief proces, waarbij de fouten aan het einde van een alinea hersteld worden of na het schrijven van een eerste versie. Een van de grote verschillen in het schrijfproces van beide schrijvers blijkt gerelateerd te zijn aan de manier waarop ze met de schrijfmodi omgaan in combinatie met hun voorkeur om verschillende modi voor verschillende subprocessen te gebruiken. Hoewel het aantal moduswisselingen vergelijkbaar is voor de twee schrijvers, zijn de strategieën van beide schrij-

vers zeer uiteenlopend. Aan de ene kant zien we geen direct effect van repairs op de keuze van de schrijfmodus; repairs leiden niet noodzakelijk tot een moduswisseling. Aan de andere kant kunnen we concluderen dat beide participanten andere correctiestrategieën ontwikkelen. De ervaren dicteerder 'negeert' fouten in de reeds geproduceerde tekst en stelt correcties uit tot een latere fase; de schrijver zonder dicteerervaring anticipeert soms op fouten in de tekst en wisselt vaker van schrijfmodus om fouten te voorkomen. Beide schrijvers verschillen in het aantal en type fouten dat ze onmiddellijk oplossen. Spraakherkenning blijkt dus een schrijfomgeving te creëren die openstaat voor verschillende schrijfstijlen. De interactie met de (incorrecte) tekst op het scherm lijkt een bepalende factor in schrijfprocessen met spraakherkenning.

II Sectie II Correctiestrategieën in geïsoleerde contexten

Schrijvers moeten tijdens hun schrijfproces steeds tekortkomingen in de 'reeds geproduceerde tekst' (text produced so far – TPSF) opsporen. Hiervoor vergelijken ze de tekst op het scherm met de mentale representatie die ze van de tekst hebben gemaakt. Als de fout is waargenomen, dient de schrijver nog twee stappen te zetten: hij moet een goede diagnose maken van de fout en ze vervolgens ook nog herstellen. In sectie 2 beschrijven we een onderzoeksproject waarbij we dergelijke correctiestrategieën hebben bestudeerd in een experimenteel gecontroleerde context. Daarbij onderzochten we hoe fouten gecorrigeerd worden in geïsoleerde zinnen. We hebben met name gekeken naar de effecten van verschillende fouttypen op de cognitieve belasting van een schrijftaak en op het revisiegedrag van de schrijvers. We vermoeden dat de belasting van het werkgeheugen van schrijvers bepalend is in de correctiestrategie en bijvoorbeeld mee de keuze bepaalt om de revisietaak dan wel de schrijftaak voorrang te geven.

In het onderzoek dat we in hoofdstuk 5 en 6 beschrijven gaan we ervan uit dat tekstcorrectie in schrijfcontexten met spraak – zoals bij dicteren met spraakherkenning – verschilt van traditionele tekstverwerking met toetsenbord en muis. Enerzijds verschilt de wijze waarop de TPSF aangeboden wordt en anderzijds verschilt het soort fouten. Spraakherkenningsprogramma's maken het mogelijk om een tekst te dicteren aan de computer. De software zet immers de steminput om in tekst op het scherm, net zoals een toetsaanslag die resulteert in een teken op het scherm. Spraakherkenning produceert echter andere fouten dan de traditionele typefouten, de woorden op het scherm zijn namelijk altijd bestaande woorden, maar niet altijd de bedoelde woorden. De vraag die in deze sectie centraal staat, is of het gebruik van spraaktechnologie van invloed is op de cognitieve processen van schrijvers.

De kern van het experiment bestond erin dat de proefpersonen deelzinnen (TPSF) moesten afmaken die hen werden voorgelegd. In

sommige van die deelzinnen kwamen fouten voor uit een van de gedefinieerde foutcategorieën. Om ervoor te zorgen dat de deelzinnen een TPSF-karakter kregen, werd eerst een context gecreëerd. Daarbij werden telkens twee deelzinnen gepresenteerd die causaal verbonden waren. De proefpersonen konden ze rustig lezen en dan beslissen om de context weg te klikken. Door de zinnen op die manier te presenteren, werd de context in het kortetermijngeheugen opgeslagen als basis voor de productietaak. Tijdens het herlezen van de TPSF moesten de schrijvers regelmatig een tweede taak uitvoeren (drukken op een knop).

Het onderzoek meet effecten van de manier waarop de TPSF aangeboden wordt: enerzijds geschreven en anderzijds zowel gesproken (voorgelezen) als geschreven. Daarnaast gaan we in op verschillende fouttypen: (1) foutgrootte: kleine fouten (één of twee letters) versus grote fouten, (2) 'input'-modus: tekst geschreven met spraakherkenning of via toetsenbord en (3) het effect van de lexicaliteit: de foutmanipulatie resulteerde enerzijds in onbestaande woorden, anderzijds in (andere) bestaande woorden.

Om de effecten van de verschillende fouttypen op het werkgeheugen bepalen hebben we een aantal online metingen verricht: voorbereidingstijd (de tijd die schrijvers nodig hebben om hun cursor te positioneren om enerzijds te vervolgen met tekstproductie of anderzijds eerst de tekst te corrigeren), productietijd en reactietijd. Bij de twee andere metingen ging het om onmiddellijk of uitgestelde correctie en ten slotte om accuratesse van de correctie.

De analyses uit hoofdstuk 5 zijn gebaseerd op geaggregeerde data. In hoofdstuk 6 beschrijven we een meer gedetailleerde heranalyse van de data via multilevel-analyse.

Samenvatting hoofdstuk 5

Het effect van fouten in de reeds geproduceerde tekst: Strategische beslissingen gebaseerd op foutgrootte, input modus en lexicaliteit

In hoofdstuk 5 beschrijven we uitgebreid het theoretisch kader van het (semi-)experimentele onderzoek, lichten we de onderzoeksmethode toe en beschrijven we de resultaten op een geaggregeerd niveau.

Het onderzoek toonde aan dat de experimentele conditie waarbij de TPSF alleen geschreven of ook gesproken aangeboden werd, de schrijfstrategie beïnvloedde. In de gesproken conditie gaven de schrijvers er vaker de voorkeur aan om de correctie van fouten uit te stellen. Dit betekent dat ze er eerder voor kozen om de tekst af te maken in de gesproken conditie dan in de geschreven conditie. Met andere woorden, schrijvers geven er de voorkeur aan om door te gaan met tekstproductie als de tekst ook gesproken wordt aangeboden. In het algemeen zorgt de gesproken tekst voor een afname in cognitieve belasting tijdens het schrijfproces: er is een kortere voorbereidingstijd en de reactietijden zijn sneller.

De schrijvers slagen er beter in om grote fouten in de TPSF te corrigeren dan kleine

fouten. Kleine fouten worden sneller over het hoofd gezien, vooral in de spraakconditie waarbij de TPSF ook voorgelezen wordt. In dergelijke gevallen ontstaat er immers bij het beluisteren van de tekst al een mentale representatie in het geheugen die bij een snelle confrontatie met de TPSF op het scherm niet leidt tot een detectie van het kleine verschil.

Kleine fouten worden dus niet altijd gedetecteerd, maar als ze gedetecteerd worden, creëren ze een lagere belasting voor het werkgeheugen. Uit de analyse van de reactietijd blijkt namelijk dat de schrijvers over het algemeen meer tijd nodig hebben om op de pieptoon te reageren bij grote fouten dan bij kleine fouten. Grote fouten vergen dus iets meer initiële verwerkingstijd dan kleinere fouten, vooral als ze voorafgaand niet worden voorgelezen. Er is een interactie-effect (modus*foutgrootte), dat erop wijst dat het aanbieden van een gesproken TPSF, de voorkeur om eerst de zin af te maken nog versterkt.

De analyse van de gedetailleerde pauzetijden uit de logging van Inputlog stellen ons in staat dit beeld nog te verfijnen. De resultaten wijzen erop dat er langer gepauzeerd wordt voor een grotere fout dan voor een kleinere fout. Dit wijst erop dat de grote fouten meer cognitief belastend zijn tijdens de schrijftaak. In de gevallen waarin schrijvers correctie van grote fouten uitstelden, konden ze de fouten daarna snel terugvinden, maar de correctie vergde wel weer meer tijd. Omdat de fouten zo duidelijk waren, konden de schrijvers ze snel detecteren, maar het oplossen ervan was tijdsintensiever. Het werkgeheugen werd in die gevallen meer belast met het ophalen van de contextinformatie die in het korte termijn geheugen opgeslagen is.

Samenvatting hoofdstuk 6

Het isoleren van de effecten van schrijfmodus ten opzichte van inputmodus, foutgrootte en lexicaliteit tijdens de correctie van tekstproductiefouten: Een multilevel-analyse

In dit hoofdstuk hebben we de data van het TPSF-experiment vanuit een hiërarchisch perspectief beschreven. In vergelijking met de unilevel statistische analyses, zoals beschreven in hoofdstuk 6, zijn multilevel-analyses statistisch krachtiger. Multilevel-analyses vormen een goede basis om te controleren of de persoons- en zinskenmerken de data-interpretatie van het effect van de experimentele conditie of de fouttypen en de interactie van de schrijvers met de TPSF verstoord hebben. Hiervoor hebben we multilevel-analyses uitgevoerd die rekening houden met de hiërarchische structuur van de data. Voor iedere variabele (reactietijd, voorbereidingstijd, productietijd, onmiddellijke of uitgestelde correctie en correctheid) presenteren we multilevel regressiemodellen. Hiermee nemen we de mogelijk verstorende persoonskenmerken in acht, terwijl we de verschillende condities en fouttypen op zinsniveau analyseren.

De analyses bevestigen in hoge mate de resultaten uit hoofdstuk 5. Ze tonen aan dat het voornamelijk de schrijfmodus (gesproken tekst en/of geschreven tekst) is, die

het werkgeheugen vrijmaakt voor de primaire schrijftaak, zowel tijdens het vervolledigen van correcte als van incorrecte zinnen. Grote fouten kosten meer inspanning, maar ze worden ook beter opgelost dan kleine fouten. Dit laatste geldt ook voor kleine fouten die resulteren in niet-bestaande woorden.



Sectie III Correctiestrategieën van professionele spraakherkenningsgebruikers tijdens het schrijven van zakelijke teksten

Samenvatting hoofdstuk 7

De 'text produced so far' in zakelijke teksten: correctiestrategieën van professionele spraakherkenningsgebruikers

Een van de uitdagingen in schrijfprocesonderzoek is om de structurele variatie tussen en binnen schrijvers te verklaren. Veel of weinig recursiviteit wordt toegewezen aan schrijfervaring, vakkundigheid, taakeigenschappen en de schrijfmodus die gebruikt wordt. In dit onderzoek richten we ons op schrijvers die spraakherkenning als hun belangrijkste schrijfinstrument gebruiken. Bovendien concentreren we ons op een belangrijk subprocess, namelijk reviseren en dan meer bepaald de correctie van fouten als bepalende factor om structurele karakteristieken van het schrijfproces te bepalen.

We hebben 10 professionele schrijvers - tevens ervaren spraakherkenningsgebruikers - geobserveerd. Deze schrijvers werden geobserveerd terwijl ze een zakelijk rapport schreven. De nadruk van het onderzoek lag op het ontdekken van strategieën om fouten op te lossen. We beschrijven de fouten waar professionele spraakherkenningsgebruikers mee te maken krijgen, hoe ze deze fouten herstellen en waarom ze voor verschillende strategieën kiezen om fouten op te lossen.

In dit onderzoek hebben we onderzoeksmethoden gebruikt die elkaar aanvullen. De methoden die we gebruiken zijn: (1) productanalyse (eindtekst), (2) procesanalyse (de data van het schrijfproces zijn geregistreerd met het logging programma Inputlog, de spraakherkenner Dragon Naturally Speaking 8.1 en het usability-programma Morae), en (3) protocolanalyse (de participanten beantwoordden vragen tijdens een retrospectief interview waarbij ze fragmenten uit hun schrijfproces te zien kregen).

De resultaten worden op drie niveaus beschreven: algemeen, subgroepen (drie schrijversgroepen) en individueel (drie participanten). De tegenstelling om fouten onmiddellijk te corrigeren of de correctie uit te stellen is bepalend voor de manier waarop schrijvers hun schrijfproces organiseren. Bovendien speelt ook het onder-

scheid tussen technische problemen en revisies een belangrijke rol. De meeste schrijvers geven er de voorkeur aan om technische problemen onmiddellijk te corrigeren. Maar dat geldt niet noodzakelijk voor revisies. De strategie om foutcorrectie uit te stellen, is niet dezelfde als de strategie om revisies uit te stellen. De algemene resultaten tonen drie verschillende patronen van foutcorrectie. Allereerst zijn er schrijvers die een eerste tekstversie zo correct mogelijk proberen te schrijven. Zij geven er de voorkeur aan om zowel technische problemen als revisies onmiddellijk op te lossen ('handle' profiel). Ten tweede zijn er schrijvers die meer dan de helft van alle fouten onmiddellijk oplossen, maar die ook een groot deel van de correcties (zowel van technische problemen als van revisies) uitstellen tot een later moment ('postpone revisions' profiel). Ten slotte is er een groep schrijvers die bij voorkeur het merendeel van de correcties uitstelt. Die groep stelt vooral de correctie van technische problemen uit naar een tweede revisieronde ('postpone technical problems' profiel). Deze uiteenlopende strategieën worden meer gedetailleerd toegelicht in de individuele case studies.

IV | Sectie IV Schrijfprocessen registreren in een Windows-omgeving

Samenvatting hoofdstuk 8

Inputlog: een onderzoeksinstrument om schrijfprocessen in verschillende schrijfmodi te observeren en analyseren

Dankzij de ontwikkeling van toetsregistratieprogramma's is het mogelijk geworden om digitale schrijfprocessen nauwkeurig te registreren en te analyseren. Dit hoofdstuk beschrijft Inputlog, een registratieprogramma dat onderzoekers de mogelijkheid biedt om schrijfprocessen te registreren in Windowstoepassingen om die daarna vanuit een aantal invalshoeken te analyseren.

De belangrijkste functies van Inputlog 2.0 Beta zijn:

- opslaan van data van schrijfsessies in een Windowsomgeving;
- genereren van databestanden als input voor verdere (statistische) analyse van tekst-, proces-, pauze-, revisie- en moduskenmerken van de schrijfsessie(s);
- integreren van loggingdata uit andere programma's;
- afspelen van geregistreerde schrijfprocessen met verschillende snelheden.

In dit hoofdstuk beschrijven we de technische en functionele kenmerken van dit onderzoeksinstrument en illustreren we een aantal toepassingsmogelijkheden aan de

hand van data uit een experimenteel onderzoek. We formuleren een aantal adviezen over hoe Inputlog gebruikt kan worden bij onderzoekstoepassingen. Daarnaast beschrijven we een aantal methoden om de geregistreerde data vanuit verschillende invalshoeken te bestuderen (bijvoorbeeld factoranalyse en progressie-analyse).

De meest kenmerkende eigenschappen van Inputlog zijn de onafhankelijkheid van een tekstverwerker (Inputlog functioneert binnen alle Windows-toepassingen), de parsingtechniek (input en proces componenten zijn gescheiden om een modulair – zeer efficiënt – systeem te garanderen), de gestandaardiseerde XML-structuur van de data die onderzoekers in staat stelt om output van verschillende programma's te integreren en ten slotte de logging van spraakherkenning (Dragon Naturally Speaking 8.1 en later).

Het hoofdstuk sluit af met een vooruitblik op nieuwe toepassingen van Inputlog die nog in ontwikkeling zijn. Hierbij onderscheiden we vier verschillende niches: revisie-analyse, ondertiteling via spraakherkenning, evalueren van typetests en de zoekvriendelijkheid van databestanden.

Curriculum Vitae

Mariëlle Leijten (1975) studied Design and Communication at the high school of Rotterdam and Text and Communication at the Faculty of Arts at Tilburg University. She wrote a thesis on 'The impact of text structure and linguistic markers on the text comprehension of elderly people' (1999). In 2000 she started a research project on the influence of speech recognition on writing processes at the University of Antwerp.

As a researcher, Mariëlle Leijten worked in a small research group on 'Writing and Digital Media' at the University of Antwerp. Next to her research projects on writing processes, she teaches various courses on business communication at bachelor, master and post-master level.

Mariëlle Leijten is active in various organizations like the special interest group on Writing (EARLI), the Dutch organization on Applied Linguistics (VIOT) and organized workshops, symposia and conferences in the field of Writing.

At the moment, she finishes her mandate at the University of Antwerp by conducting further research on Inputlog (revision module), error correction strategies and speech recognition.

www.ua.ac.be/marielle.leijten

