An Evaluation of Procedural Instructional Text

Nathalie Colineau, Cécile Paris
CSIRO, Mathematical and Information Sciences,
Locked Bag 17, North Ryde, NSW 1670, AU
{nathalie.colineau,
cecile.paris}@csiro.au

Keith Vander Linden
Dept. of Computer Science, Calvin College,
Grand Rapids, MI 49546, USA
kvlinden@calvin.edu

Abstract

This paper presents an evaluation of the instructional text generated by Isolde, an authoring tool for technical writers that automates the production of procedural on-line help. The evaluation compares the effectiveness of the instructional text produced by Isolde with that of professionally authored instructions, such as MS Word Help. The results suggest that the documentation produced by Isolde is of comparable quality to similar texts found in commercial manuals.

1 Introduction

Instructional text is a useful and relatively constrained sub-language and has thus been a popular target for research-oriented natural language generation (NLG) systems. Much work has been done in this area, e.g., (Rösner & Stede, 1992; Kosseim & Lapalme, 1994; Paris & Vander Linden, 1996; Power *et al.*, 1998), demonstrating that existing technology is adequate for generating draft instructions. However, only a few of these projects have been formally evaluated, e.g., (Hartley *et al.*, 2000), and the evaluations performed have focussed on the fluency and grammaticality of the output text rather than on its effectiveness. This tends to be the case, in fact, for evaluations of NLG systems in general. People are asked to rate the acceptability of the generated texts or to compare them to human-authored texts - e.g., (Lester & Porter, 1997; Callaway & Lester, 2001), without measuring the actual impact of the texts on their intended users.

The evaluation of the STOP system (Reiter *et al.*, 2001) is a notable exception to this trend. STOP produced texts tailored to individual smokers intended to convince them to stop smoking. As an evaluation, the researchers performed a large-scale study of how effective the generated texts were at achieving this goal. Rather than checking the output text for errors, or comparing its fluency with that of hand-written text, they compared how often readers of STOP's individually tailored texts actually stopped smoking as compared to readers of generic, generated texts. Thus, the evaluation assessed the relative effectiveness of tailored and generic texts at achieving their intended goals.

In our evaluation, we also sought to perform an evaluation of the effectiveness of the instructional texts produced by Isolde. Isolde[1] is an authoring tool for technical writers that automatically generates parts of a system's on-line help (Paris *et al.*, 1998b). In this domain, the writer's goal is to help the users achieve their goals. It is thus crucial to assess the effectiveness of the instructional texts in a real task. Unlike the STOP evaluation, however, we compared the effectiveness of our generated texts with that of human-authored texts. In this paper, we first introduce the type of texts that Isolde generates and give an overview of Isolde. We then present the evaluation we conducted and draw some conclusions.

2 Procedural Help

Documentation for interactive devices typically includes conceptual help, business help, and procedural help (Paris *et al.*, 1998a). Conceptual help defines the concepts used in an application, business help indicates how the application is embed-

ded in its context of use, and procedural help enumerates the series of steps required to perform a goal. Isolde focuses on procedural help. Procedural help, which can be seen as answering the question "How to?", is an attractive target text for NLG systems because it is:

Constrained: Procedural help describes a system's functions in terms of users' actions on the interface. It is highly structured and heavily based on the system behavior. Its automatic production thus seems realistic.

Significant within the help system: Procedural help is a key element of "minimalist instructions" (Carroll 1990), whose philosophy is based on the argument that learning software is more effective when the software documentation is short, simple and directed towards real work activities. This notion of brevity can also be linked to the Grice's maxim of quantity in the discourse theory (Grice, 1979). Young (1999), in fact, uses this principle to select the content of plan descriptions and compute concise instructions.

Routine and time consuming to produce: Technical writers consider procedural help as the easiest but also the most routine and tedious part of the documentation process because they must systematically perform all possible functions, recording their actions step by step. Writing procedural help can be the most time-consuming activity of the technical writing process (Paris *et al.*, 1998a). It is thus desirable to automate it, or at least support its production (cf. Power *et al.*, 1994).

Based on these motivations, Isolde aims at automating the production of procedural on-line help. We believe this addresses an issue that is both important and relevant in documentation production.

3 System Overview

Isolde provides an interactive on-line help drafting tool aimed at supporting technical writers. From an analysis of technical writers at work, we noted that they typically begin by building a representation of the application's functionalities by executing tasks step by step, recording all the steps and the interface feedback (Power *et al.*, 1994; Ozkan *et al.*, 1998). This representation ranges from textual to semi-formal, e.g., flowcharts. Then, from this representation, they write the text proper.

Isolde offers facilities to support and formalize the first step, i.e., building a representation of the application's functionalities that can be reused later in the documentation maintenance process. Isolde then also automates the second step, i.e., the generation of the text. Isolde aims at being integrated into both the writers' environment and the software design process.

The representation chosen for the application's functionalities is a task model, a semi-formal notation. It is graphically represented in the Diane+ notation (Tarby & Barthet, 1996) and was tested with technical writers for usability (Ozkan *et al.*, 1998). Figure 1 shows the Isolde architecture. The task model can be manually entered using a dedicated graphical editor Tamot. Alternatively, this model can be acquired automatically or semi-automatically from available sources of information through a variety of tools - e.g., a text-to-task extraction tool, an interaction recorder or a tool to construct automatically a draft task model from the

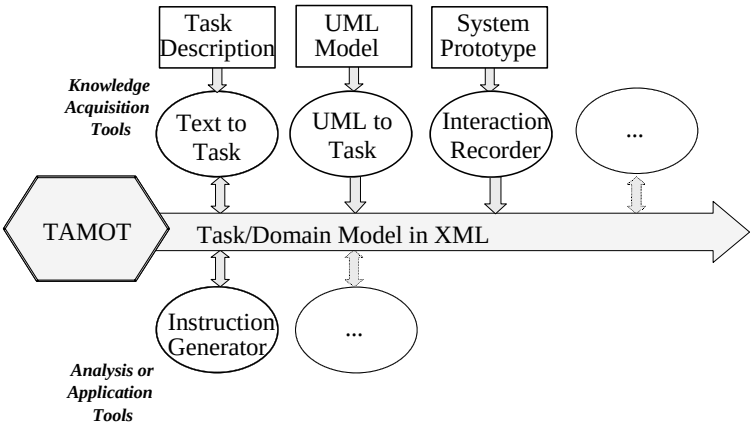


Figure 1: Isolde Architecture

output of a CASE tool[2] (Lu *et al.*, 1998; Vander Linden *et al.*, 2000). Note that the task model for the application may have already been built by HCI specialists as part of their involvement in software team - e.g., (Balbo & Lindley, 1997; Diaper & Stanton, in press) and can then be re-used or augmented by technical writers to ensure their suitability for on-line help generation.

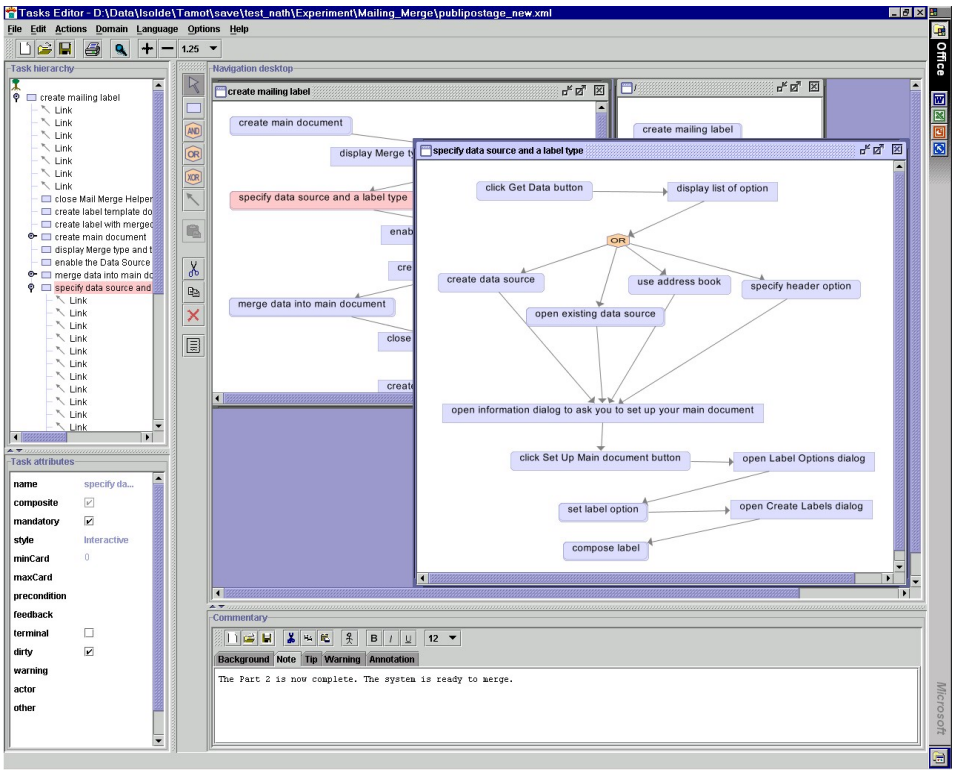


Figure 2: Task model design to create mailing labels with MS Word

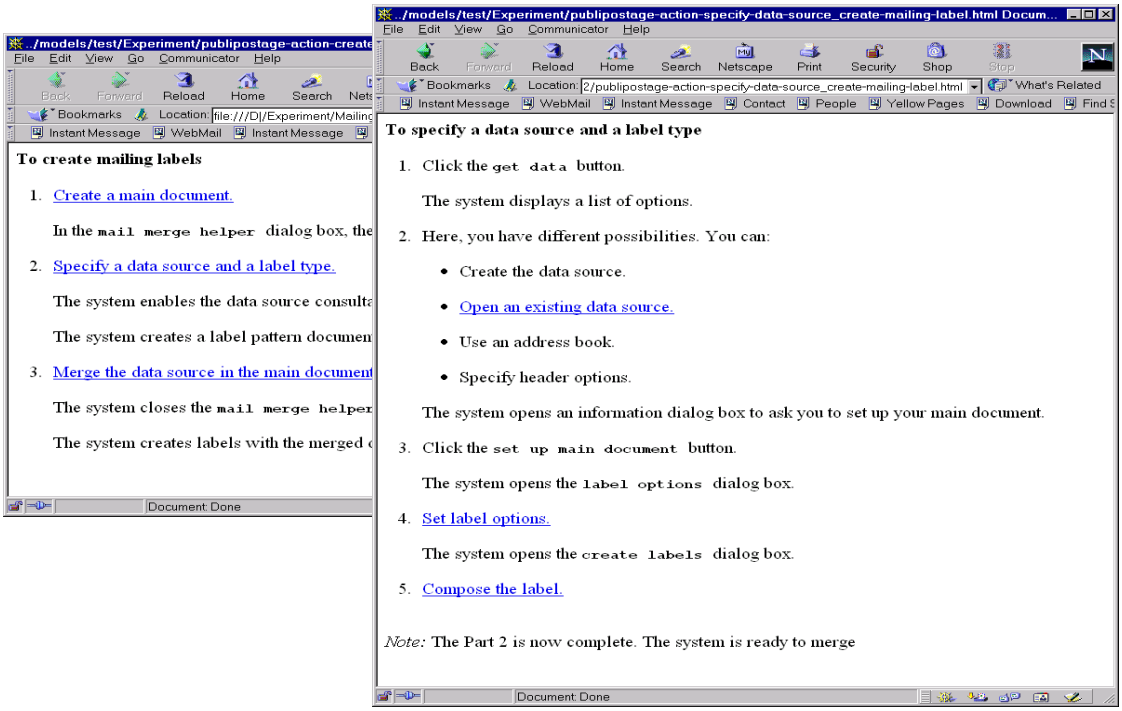


Figure 3: Hypertext output displayed in a Netscape browser - the mailing labels task

When satisfied with the task model, the technical writer exports the model to the generator for the on-line help to be generated. In Isolde, the technical writer refines the input and controls the generation of instructions through Tamot. A task model example, shown in Tamot, is presented in Figure 2. This window includes a tree-structured representation of the task hierarchy on the left, and a graphical representation of the tasks on the right.

The Instruction generator uses: (1) the Moore & Paris text planner (Moore & Paris, 1993); (2) a sentence planner implemented as extensions to the text planner; and (3) the KPML development environment and lexico-grammatical resources (Bateman, 1997). This system is implemented as a LISP server. It plans the instructions using discourse plans that handle any task model configuration, including sequences, compositions and Boolean connectors. It plans hypertext links and can integrate canned text with generated text. Style parameters have also been implemented, giving the technical writers some control at the discourse or sentence level. For example, writers can decide to produce a concise text by aggregating simple tasks and suppressing levels of decomposition, or they can choose to produce step-by-step instructions, in which each task decomposition is presented in a separate frame. Figure 3 shows the instructions generated for the task model in Figure 2.

4 Evaluation of the Generated Instructional Text

Our experiment attempted to assess the effectiveness of the help generated by Isolde as compared with manually authored help. For both types of texts, we:

- Measured the user's performance in accomplishing a specific task (i.e., task achievement and time needed); and
- Asked the users to rate the usefulness of the texts, the adequacy of the content and the coherence of the organization.

We did not evaluate grammatical correctness as in AGILE (Hartley *et al.*, 2000). Given the aim of documentation, we need to ensure that the generated instructions allow the users to achieve or learn about their task, whatever the quality or the complexity of the text generated.

4.1 Experimental Design

Our methodology involved four steps: (1) choosing three tasks in Word; (2) designing the three corresponding task models; (3) producing the on-line help with Isolde for these models, and (4) evaluating the help on two user groups: one group received the Word help, the other the automatically generated help. Participants to the experiment were not aware of which help they were using.

Task Selection: To compare the effectiveness of Isolde help with that of manually authored help, we asked users to perform a real task, using the on-line help as a resource. We decided to work with Word, as it provides both a task environment and on-line instructions. Thus, our aim was to select 3 tasks, 2 simple and 1 complex[3] that would be new for the subjects, to help prevent introducing a bias based on prior knowledge and encourage users to read the help. We also chose tasks such that their Word help text was "self-contained" (i.e., without extensive reference to other parts of the help) and the text generated by Isolde for the task would be of similar reading complexity. Our final constraint was that the 2 simple tasks had the same number of elementary actions. We had no prior assumption as to what task would be easier to model or document than any other task. With these constraints, we chose:

- Task 1: create a document template and save it in a specific directory;
- Task 2: create index entries;
- Task 3: create mailing labels by merging a label template with an address list, and save the label pattern.

Task Design: To generate the on-line help for to these tasks, we first had to design the task models. We did so using Tamot. As typically done by technical writers, we executed the tasks step by step, recording all the steps. Feedback expressions (e.g., display of windows, confirmation messages) were also included, either as system actions, or as notes or warnings with canned text. The aim was to be as close as possible to the system behavior.

Hypertext Generation: When the task model was completed, we generated the corresponding on-line help. We then used the Flesch (1974) score[4] to

	Task 1		Task 2		Task 3	
Readability scores	Word Help	Isolde Help	Word Help	Isolde Help	Word Help	Isolde Help
Average Sentence Length (ASL)	16.241	7.821	11.882	9.791	15.548	7.609
Flesh Reading Ease Score	62.821	70.644	68.712	75.671	77.668	71.951

Table 1: Comparison of Word and Isolde readability score

compare the readability of the generated texts with the Word help (see Table 1). This was to ensure both texts would be of similar reading complexity and would thus involve the same amount of time to consult. For the experiment, only one of the help texts was accessible to the user, displayed in a Netscape browser to preserve anonymity.

Formation of Subject Groups: A total of 35 subjects did the experiment (3 tasks per subject), split into 2 groups. Subjects were randomly assigned to a group, but we ensured an even number of men and women in each group. The subjects were not expert in Word, but they knew how to use the software.

4.2 Scenario of Experiment

The experiment consisted of asking subjects to perform the 3 tasks described above with Word. For each task, they were given some directions as to what was expected of them (e.g., create a template with the CSIRO logo, and save it in a specific directory). The directions did not include explanations on how to achieve the task. These were to be found in the on-line help provided. Subjects could consult the help at any time (i.e., before or while performing the task). Subjects were told to read the directions and ask for clarification if required before starting on the task. After each task, they filled out a questionnaire asking them to rate the help they used.

The questionnaire aimed at evaluating the usefulness of the help, the quality of its content (i.e., its quantity and relevance) and the coherence of its organization. The questions asked and the factors of acceptability that they rate are shown in Table 2. Each question was answered using a six-point scale, assigning letter grades A (high) through F (low). These letters were later converted into digits from 6 to 1 for the statistical analysis. The questionnaire also included questions that checked the users' previous level of familiarity with the task.

During the experiment, we recorded the time to measure the user's performance. We limited the allowable time (10 minutes for the simple tasks and 15 minutes for the complex task) to encourage the users to consult the help instead of exploring the application by themselves[5]. We observed whether subjects consulted the help or not, and the number of times they did so. Finally, to evaluate the success rate on each task, we recorded the errors made. The marking scale was set as follows:

- For Task 1, the subject lost one point if the document was saved in the wrong directory and two points if it was not saved in the template format.
- For Task 2, the subject lost one point for each index entry that they marked incorrectly.
- For Task 3, the subject lost one point if the mailing labels were not created, and another point if the label pattern was not saved.

Group 1 was assigned Word help for Tasks 1 and 3, and Isolde help for Task 2, while Group 2 was assigned Isolde help for Task 1 and 3, and Word for Task 2.

Factors	Questions
Usefulness	How would you evaluate the usefulness of the help?
Adequacy of Content	Did the help provide you with enough information to perform the task?
	Was the information provided in the help relevant for your task?
Coherence	Did the help give you a clear picture of the steps required to accomplish the task?
	How well was the help organised?

Table 2: Grading factors presented to subjects

4.3 The Results

Because we wanted to assess the effectiveness of the help in aiding a user to accomplish a task, we first screened out users who knew the tasks before hand (based on the questionnaire)[6], and those who did not consult the help at all (based on our observations). As a result, we were left with 12 subjects out of 35 for Task 1, 34 for Task 2, and 29 for Task 3. We analyzed the data for task performance in terms of the time it took to finish the task and the number of errors made. In all cases, we ran an Anova single factor, and, when results are significant, we report them for a 0.05 level of confidence.

Results on Task Performance: With respect to errors, there was no evidence that either help was more effective than the other. The small differences observed were not statistically significant. This is shown in Figure 4.

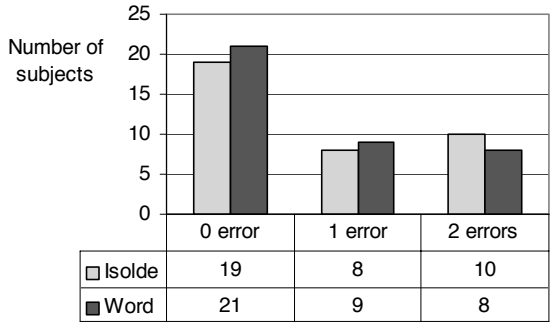


Figure 4: Task performance comparison - all tasks

With respect to time, we observed interesting differences that were contrary to our expectations. Table 3 presents the results obtained by running an Anova. The times are reported in seconds.

	Tasks 1 & 2	Task 2	Task 3
Isolde Help	388.86	428.58	398.83
Word Help	278.91	296.70	553.63
Difference	*109.94*	*131.88*	**154.80**
Anova (F-test)	5.77	6.33	7.82
Level of Confidence	0.02	0.01	0.01

Table 3: Time Performance (in seconds)

The first column combines Tasks 1 and 2; Task 1 alone did not allow separate computation due to the small number of subjects. The difference in time performance was in favor of Word for the simple tasks (indicated in italics in the table), while it was in favor of Isolde for the complex task (indicated in bold).

Results on acceptability of the Help: Figures 5, 6 and 7 show the ratings of the different help texts for the different tasks, based on the responses on the questionnaire. We computed means for both Isolde and Word, using the six-point scale, for each task and for each of the help dimension we would like to observe: (1) usefulness regarding the task, (2) quality of content provided, and (3) coherence and clarity of the help organization.

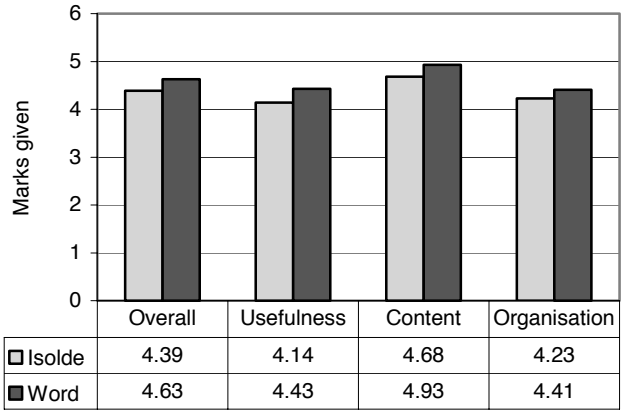


Figure 5: Rate for the simple tasks (Task 1 and 2)

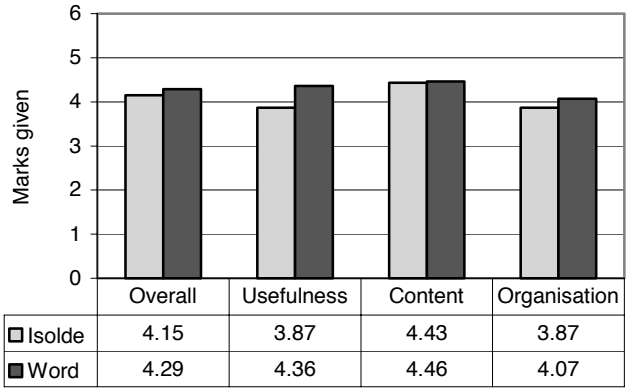
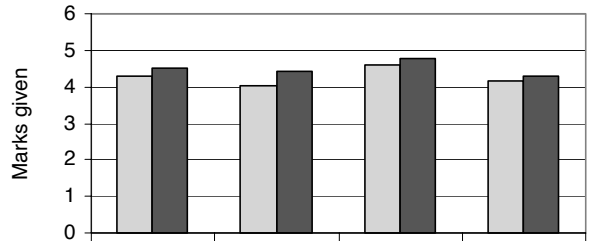


Figure 6: Rate for the complex task (Task 3)

The Overall column summarizes these different values. As shown in Figures 5, 6 and 7, Isolde scored closely to Word, within approximately 1/4 of a point for the content and the organization, and 1/3 of a point for the usefulness. In all cases, both Isolde and Word were positively evaluated on average.



	Overall	Usefulness	Content	Organisation
Isolde	4.29	4.03	4.58	4.14
Word	4.50	4.41	4.76	4.28

Figure 7: Rate combining all the tasks

We checked whether the differences were significant or not by running an Anova on each task for each dimension. The results did not show any significant differences between the two help texts.

	Overall	Usefulness	Content	Organisation
Anova (F-test)	2.18	1.61	0.67	0.00
Level of Confidence	0.14	0.20	0.41	0.94

Table 4: Anova result combining all the tasks

Table 4 reports the results obtained when the scores are aggregated over all the tasks, though we also performed the test for each separate task. Our results indicate that, in terms of acceptability and usefulness of the help, Isolde's performance approaches that of the manually authored texts.

5 Discussion

People are usually reluctant to consult help and typically consider on-line help to be unuseful. Our work aims at providing help texts such that users can quickly find the information they need. Our challenge then was to provide enough relevant information, without extraneous details, within the constraints imposed by the knowledge available to generate text automatically. As shown in Table 1, the help generated by Isolde and manually-authored texts are similar in terms of their readability score. Isolde, however, generates shorter sentences, with strictly the information required to achieve a task. While our generated texts thus often contain less information than manually-authored texts (because of the knowledge available to produce them), they constitute less text to browse (i.e., each instruction is shorter) and may thus make it easier to access important information. From the experiment, it seems that providing this type of text has a significant impact on complex tasks (where the amount of consultation and the time spent understanding the tasks are greater), but no impact on simple tasks (where users seem more comfortable reading the manually-authored texts).

6 Conclusion

This paper has discussed an evaluation of the procedural instructions generated by the Isolde authoring tool. The evaluation compared the effectiveness of the instructional texts generated by Isolde with those written by technical writers in the context of a real task. The results showed: (1) no significant differences with respect to the number of errors made while performing the tasks; (2) some significant differences in time performance, in favour of Isolde for complex task and in favor of the manually-authored texts for simple tasks; and finally, (3) no significant differences with respect to the acceptability of the help. These results are encouraging because they show that the effectiveness of Isolde's automatically generated texts is comparable with that of manually-authored texts, even though they often contain less information.

Acknowledgements

We wish to thank Lu, S., past members of the Isolde team, the people who participated in the experiment and Bétrancourt, M. for her advice during the experiment. We acknowledge the support of ONR (grant N00014-99-0906), CSIRO and Calvin College.

References

Balbo, S. & Lindley, C. (1997). Adaptation of a task analysis methodology. In *Proc. of Interact'97*. Sydney: Australia, Chapman and Hall. 355-361.

Bateman, J. (1997). Enabling technology for multilingual natural language generation: the KPML development environment. In *Natural Language Engineering*, 3(1):15-55.

Callaway, C. & Lester, J. (2001). Evaluating the effects of natural language generation techniques on reader satisfaction. In *Proc. of CogSci'01*, Edinburgh, UK.

Carroll, J. (1990). *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. MIT Press, Cambridge, Massachusetts.

Diaper, D. & Stanton, N. (Eds)(in press). *The Handbook of Task Analysis for Human-Computer Interaction*. Lawrence Erlbaum Associates.

Flesch, R. (1974). *The art of readable writing*. Harpercollins.

Grice, H.P. (1979). Logique et Conversations. In *Communications* nº30, 57-72.

Hartley, T., Scott, D., Kruijff-Korbayová, I., Sharoff, S., Sokolova, L., Dochev, D., Staykova, K., Cmejrek, M., Hana, J. & Teich E. (2000). Evaluation of the final prototype, deliverable of the AGILE project. URL: http://www.itri.brighton.ac.uk/projects/agile

Kosseim L. & Lapalme G. (1994). Content and rhetorical status selection in instructional text. In *Proc. of INLG*, Kennebunkport, ME. 53-60.

Lester, J. & Porter, B. (1997). Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. In *Computational Linguistics*, 23(1):65-101.

Lu, S., Paris, C. & Vander Linden, K. (1998). Towards the Automatic Construction of Task Models from Object-Oriented Diagrams. In Chatty, S., and Dewan, P., (Eds.) *Engineering for Human-Computer Interaction*, Kluwer Academic Publishers, 169-189.

Moore, J. & Paris, C. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics* 19(4):651-694.

Ozkan, N., Paris, C. & Balbo, S. (1998). Understanding a Task Model: An Experiment. In *Proc. of HCI'98*, H. Johnson. K. Nigay and C. Roast (Eds), Springer 123-138.

Paris, C. & Vander Linden, K. (1996). Drafter: An Interactive Support Tool for Writing Multilingual Manuals, *IEEE Computer*, 29(7):49-56.

Paris, C., Ozkan, N. & Bonifacio, F. (1998a). The Design of New Technology for Writing On-Line Help. In *Proc. HCI'98*, H. Johnson. K. Nigay and C. Roast (Eds), Springer, 189-206.

Paris, C., Ozkan, N. & Bonifacio, F. (1998b). Novel Help for On-Line Help. In *ACM SIGDOC'98*, Quebec City, Canada, September, 70-79.

Paris, C., Tarby, J. & Vander Linden, K., (2001). A Flexible Environment for Building Task Models. In *Proc. of the HCI'01*, Lille, France.

Power, R., Pemberton, L., Hartley, A. & Gorman, L. (1994) Drafter: User requirements analysis, WP2 Deliverable, Drafter Project IED4/1/5827.

Power, R., Scott, D. & Evans, R. (1998). What You See is What You Meant: direct knowledge editing with natural language feedback, In *Proc. of ECAI'98*, Brighton, UK, August.

Reiter, E., Robertson, R., Lennox A. S. & Osman, L. (2001). Using a randomised controlled clinical trial to evaluate an NLG system. In *Proc. of ACL'01*, Toulouse, France, 434-441.

Rösner, D. & Stede, M. (1992). TECHDOC: A system for the automatic production of multilingual technical documents. In: G. Görz (Eds): *KONVENS 92 - Proc. of the German conference on natural language processing*. Springer, Berlin/Heidelberg.

Tarby, J.C. & Barthet, M.F. (1996). The DIANE+ Method. In *Proc. of CADUI'96*. Namur, June 5-7, J.Vanderdonckt (ed). 95-119.

Vander Linden, K., Paris, C. & Lu, S. (2000). "Where Do Instructions Come From?" Knowledge Acquisition and Specification for Instructional Text. In *IMPACTS in Natural Language Generation: NLG Between Technology and Applications*, Schloss Dagstuhl, Germany. Becker, T. & Busemann, S. (Eds). DFKI report D-00-01, 1-10.

Young, R.M. (1999). Using Grice's maxim of quantity to select the content of plan descriptions. *Artificial Intelligence* (115), 215-256.

---

[1] Integrated Software and On-Line Documentation Environment.

[2] Computer Aided Software Engineering Tool.

[3] The difference between a simple task and a complex one is the number of elementary actions required to perform the task (18 *vs.* 40, in our experiment). It is important to note that the higher the number of elementary steps, the deeper the decomposition will be.

[4] A standard document has a Flesh Reading score of approximately 60 to 70. The higher the score, the easiest a document is considered to be.

[5] This is not meant to go against the minimalist approach, which intends to encourage an exploration of the system in a learning environment. This is a means to control our experimental situation and observe the effectiveness of the help in a task situation.

[6] The subjects who knew the tasks were not filtered ahead of time to allow us to observe task performance and help usage differences between “novices” and “experts”. No major differences were found (due to the lack of subjects), but it was still interesting to have qualitative input on these issues to inform future experiments.