The Paradox of the Assisted User: Guidance can be Counterproductive

Christof van Nimwegen[1] **Daniel Burgos[2]** **Herre van Oostendorp[1]** **Hermina Schijf[1]**

[1] Utrecht University, Center for Content and Knowledge Engineering

Padualaan 14, 3584CH, Utrecht, The Netherlands. {christof, herre, h.schijf}@cs.uu.nl

[2] Open University of the Netherlands, Educational Technology Expertise Centre

Valkenburgerweg 177, 6401DL, Heerlen, The Netherlands. daniel.burgos@ou.nl

Abstract

This paper investigates the influence of interface styles on problem solving performance. It is often assumed that performance on problem solving tasks improves when users are assisted by *externalizing* task-related information on the interface. Although externalization requires less recall and relieves working memory, it does not instigate planning, understanding and knowledge acquisition. Without this assistance, task-information must be *internalized,* stored in the user's memory, leading to more planning and thinking and perhaps to better performance and knowledge. Another variable that can influence behavior is "Need for Cognition" (NFC), the tendency to engage in effortful cognitive tasks. We investigated the effects of interface style and cognitive style on performance using a conference planning application. Interface style influenced behavior and performance, but NFC did not. The internalization interface led to more planful behavior and smarter solutions. When planning and learning are the aim, designers should thus beware of giving a user (too) much assistance. Understanding how people react to interface information can be crucial in designing effective software, especially important in the areas of education and learning.

Author Keywords

Interface design, planning, problem solving, plan-based, display-based, externalization, Need For Cognition

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: Interaction, Styles, Graphical user interfaces (GUI)

INTRODUCTION

Software and web applications have seen a tremendous development during the last decades, becoming more advanced, and demanding fast learning from users. Nowadays, the importance of *usability* in software development is mostly acknowledged, at least to some extent. The interfaces of today are by no means comparable with command-line interfaces of the past. A recurring issue in usability guidelines is the importance of minimizing "user memory load" [12], also referred to as *computational offloading* [17]. This means that the working memory of a user is relieved so that a maximum of cognitive resources can be devoted to the task at hand. The externalization of certain information could be of assistance to the problem solver. The need to plan is reduced, and externalization can limit the possibilities in a given situation, thus reducing the problem space that has to be searched [24].

A way to implement some degree of externalization is to make parts of the interface context-sensitive, e.g. by hiding or disabling functions that are not applicable at the moment. By doing so, the user is "taken by the hand" by limiting choices and providing feedback [23]. Examples are wizards, help-options and graying-out menu-items that don't permit using them, thus offering a context-sensitive interface with only "possible" actions. For example, in Word you cannot select "paste" from the "edit"-tab in the menu when nothing is copied first. "Paste" shown in gray color indicates *both* its presence *and* its unavailability. "Graying out" *externalizes* the applicability of an operator on a certain moment, and makes remembering this information unnecessary. In the opposite situation, when no such features are provided, a user has to *internalize* the information himself, and store this information in his/her memory. Concerning the graying out of menu items in this manner, there have been some famous remarks in the past from within the HCI community [19]. Mostly, irritation was expressed stemming from the fact that the graying out in itself raised new questions. In complex software, users often did not know *why* something was grayed out, possibly causing more frustration than not graying out items at all, leave alone the chance of graying out being of extra help.

Besides the fact that externalizing relevant information is almost default in direct manipulation interfaces, there is also research advocating the use of externalization. An experiment by Zhang and Norman [24] showed that relieving working memory by externalizing information can be useful for cognitive tasks; the more is externalized, the easier it is to solve the problem.

CHI 2006, April 22–28, 2006, Montréal, Québec, Canada.

Despite such results showing that supporting the user can make tasks easier, it could also be argued that externalization may have negative consequences for task performance and knowledge acquisition. In general, there is the notion that learning is more effective when people experiment and discover for themselves. Exploratory learning as a whole has been a subject of research in many domains. Carroll [4] for example, already more than a decade ago propagated minimalism in design and instructions. An example is to leave instructions intentionally incomplete to promote intrinsic interest through a problem challenge. Perhaps knowledge gained using an externalization-based interface is more volatile and difficult to transfer to other situations. In the context of this presumed volatility there is a famous study investigating what users remember of the detailed content of the Mac Write interface [11]. It showed that even experienced users recall little of menu contents, even though they use them all the time and achieve successful performance. Although the needed information for the task was found and applied, it was not remembered later on. This is not always a disadvantage (e.g. with a trivial task) but can be undesirable when learning or gaining insight *itself* is exactly the aim of the task, or when making errors comes with a cost.

Payne, Howes and Reader [16] state that internalization vs. externalization effects can be framed as just one example of the cost/benefit analysis of environmental action. When information is externalized on the screen, users simply use it (cheap) and don't need to incur the cost of storing relevant information in their memory, as the information is reliably always available. This in turn allows users to quickly being able to perform a task. However, when information is not externalized, planning and learning occurs, making it possible to optimize the performance. O'Hara & Payne [13, 14] distinguished between plan-based and display-based problem solving. In plan-based problem solving one uses detailed problem strategies from long-term memory. Display-based problem solving on the other hand makes little use of acquired knowledge but relies on interface information. Plan-based activity eventually leads to a shorter solution route, because steps are planned, and no unnecessary steps are taken, while a display-based strategy involves more steps because of more searching. O'Hara and Payne tried to provoke plan-based problem solving behavior by varying the operator cost in an interface. A puzzle was solved in a low-cost interface and a high-cost interface. The high-cost interface imposed a much longer lockout time on subjects after performing an action than the low-cost interface. Results showed that the high-cost interface indeed induced a more plan-based approach, which in turn led to better task performance (fewer moves). Also, the task knowledge acquired with the high-cost interface could more easily be transferred to problems within the same domain. They warned, however, that just freeing up cognitive resources is not enough. The design of the problem-solving environment has to encourage the use of these resources for planning as well. In short, strong reliance on the visual display may result in less transfer of skills and less planning. Research using two interfaces to the Tower of Hanoi [18] showed converging results. Subjects used a command line interface or a direct manipulation interface to the problem. It showed that subjects performed better (in the long run) with the command line interface (although direct manipulation interfaces are generally being perceived as "easier"). Thus, externalizing information may not be beneficial when the ultimate goal is to achieve better, transferable task performance and knowledge. Besides better transfer to other situations and problems, one could imagine that possessing better internal knowledge-elements, leads to better performance even when one has worked with the application some longer time ago.

In other research by the authors [20] an abstract version of the Missionaries and Cannibals Puzzle was used. We investigated the effects of internalization vs. externalization on task performance, knowledge, *and transfer* over time to different problems in the same domain. This problem, called Balls & Boxes (B&B) had to be solved using an application that existed in two interface styles: externalization vs. internalization. Externalization was realized by graying out inapplicable (momentarily unavailable) buttons that were needed to perform certain operations. In the internalization condition this support was absent. The experiment consisted of two sessions with 8 months between them. Surprisingly, not ever, on any task aspect did subjects in the externalization condition perform better. It also showed that subjects in the internalization condition had better *knowledge* afterwards. Even after 8 months, the internalization subjects still proved to have better knowledge and now also better performance. In addition, these subjects performed better at transferring the acquired rule-knowledge to a different but similar task.

This finding is similar to results from Palmiter and Elkerton [15], although they did not focus on interface styles per se. In an experiment users were presented with animated demonstrations, textual instructions or a combination of both to learn a program. It showed that demonstrations (regarded as an easy way to learn) may not provide useful knowledge for retention and transfer. The demonstration group was faster and more accurate when learning the tasks, but several days later the textual group was faster and as accurate performing a similar task. The results confirmed the observation that users who "worked harder" had advantage later on, and in a transfer situation. Perhaps the textual procedures were learned better due to differences in processing effort. The text learners may have expended more effort to read, integrate and understand the information, resulting in improved long-term memory. Users who watched the demonstrations may have passively observed them without processing the information solidly.

In a follow up study [22] we investigated whether possible differences in attitude to problem solving between subjects were of influence. Besides varying interface style (internalization vs. externalization) we tried to influence task conception or attitude by giving instructions encouraging plan-based behavior in one group, and encouraging display-based behavior in the other group. This was done using specific instructions before the start of the tasks, meant to change the affective state the subjects. We provided either a low-planning instruction ("try to solve the problem as fast as possible, mistakes are not a problem") or a high-planning instruction ("try to solve the problem as economically as possible, it will pay off"). Again, in general, internalization subjects performed better and had better knowledge. The instructions did not have any effect on *externalization* subjects. It seemed as if upon being confronted with our type of externalization interface, subjects ignored, or even forgot the planning instruction altogether. However, the instruction *did* have effect on internalization subjects. When receiving a high-planning instruction, they attempted only half as many illegal moves as compared to those given a low-planning instruction. Also the number of unnecessary moves was lower with the high-planning instruction.

Puzzles like B&B [20, 21, 22], TOH [18] and the 8-puzzle [13] that were used in the mentioned experiments, have been extensively studied in the problem-solving literature [18], and user behavior can be precisely traced with them. They can be solved both by planning and by trial-and-error. Although being excellent tasks from a scientific point of view, a drawback is that findings from these experiments may not be relevant in less abstract tasks. O'Hara and Payne [14] tried to examine whether their findings with an abstract problem (the 8-puzzle) could be generalized to an administrative task. In their experiment, subjects had to prepare letters that had to be sent to visitors of a conference, by copying the information and pasting it into the appropriate letter. Results showed that insights gained from experiments using an abstract problem, were also applicable to a less abstract administrative task.

THE CURRENT RESEARCH PROBLEM

We argue that in problem-solving situations where people need to learn the underlying rules of a system, make as little mistakes as possible, or find a solution as economically as possible, the use of a plan-based approach is preferred. If it is preferable that people use a plan-based approach during interaction, one should find out how people can be persuaded to use that approach. We will investigate our assumptions, with a more realistic life-like task of which the functioning and appearance are more towards everyday-realism. We see our current task, involving constraint-scheduling, as a step towards realism and regard the shift from classic puzzles to our current task as an intermediary one. The difference is that not so much the "catch" to a math-like puzzle has to be found or that continuous action sequences [13] have to be performed. We mean to extend prior work in the sense that users are presented with tasks that have a "real life" goal, appearance and controls. Not so much the specific functions or controls of the application have to be "learned" (they are quite straightforward) but users rather had to attempt to uncover heuristics of problem solving behavior provoked by different interface styles.

Interface Style

We will again investigate the influence of interface style on a problem solving performance task. We define the externalization of the underlying rules as visualizing the result of *applying a set of constraints* onto the interface. Externalization shows which actions are allowed with the object at hand, so the user is able to see the *outcome* of the *application* of the rules. This is comparable with the graying out of menu items in for example Microsoft Word, to show that the current situation does not permit those functions to be used, i.e. the constraints are not met (e.g. selecting the paste function before something has been cut or copied).

Cognitive Style

In [20] we investigated the influence of externally induced motivation. Here we included possible differences in task conception or problem solving attitude, focusing on more stable personal qualities of people that could influence behavior. We measured "need for cognition" (NFC), a construct that measures the tendency of individuals to engage in and enjoy effortful cognitive tasks [2]. It has consistently explained variations in individuals' predisposition to engage in varying levels of cognitive activity, making it reasonable to presume an effect on planfulness. Both persons high and low on NFC need to make sense of their world (e.g. in a computing environment), but do this differently. Persons with a high NFC love to seek, reflect on and reason about information, whereas someone with low NFC only thinks as hard as (s)he has to and is inclined to rely on others. The fact that high-NFC persons generally recall more information than low-NFC persons could have implications for HCI. This is interesting for research on adaptation, where the non-existence of a 'one-size-fits-all'-solution [10] and the importance of adapting external representations to the emotional state of learners is emphasized. Literature on learning systems proposes adapting to various constant and volatile user characteristics [1]. Crystal and Kalyanaraman used a web information-seeking task, in which they compared clear vs. unclear link-labels and informative feedback vs. no feedback while pages where loading [6]. NFC indeed influenced aspects of usability and user responses. Affective measures such as perceived speed and perceived ease of information location were positively associated with NFC. Regarding cognitive effects it showed that NFC was negatively associated with information-seeking time and navigational moves. The authors stated "*If, in a particular HCI context, individuals high in NFC exhibit significantly different behavior than individuals low in NFC, usability could likely be improved by providing* *interfaces optimized for each group.*" In context of afore mentioned research [22], this could mean that if high-NFC persons display different behavior than low-NFC persons, our interface manipulations may have different effects.

Hypotheses

We formulated two hypotheses:

H1: Internalization leads to a more plan-based strategy and better performance than externalization. Having information externalized, providing on-screen assistance tempts users to rely on the interface and *not* to form plans. The internalized condition lacks this guidance, encouraging planning and thinking before acting. We therefore expect that requiring subjects to internalize the information will lead to more elaborate planning and consequently better task performance such as more efficient solutions. We also expect better knowledge afterwards.

H2: High NFC leads to a more plan-based strategy than low NFC. People with high NFC have high intrinsic motivation to think and engage in effortful cognitive tasks, and will show more plan-based behavior. Consequently, we expect the subjects with a high NFC to have better task performance and knowledge.

METHOD

Subjects

43 subjects participated in our experiment, 17 male, 26 female. Participants were 19 to 32 years old, and were following or had recently followed higher education. The experiment took at most one hour and subjects received a €5 reward.

Material

The experiment was conducted in the Usability Lab at the Center for Content and Knowledge Engineering, Utrecht University. It consisted of three parts: a questionnaire to determine NFC, a planning task called "Conference Planner" and a post-experimental questionnaire.

Part 1: NFC questionnaire

We used a Dutch translation of the 18-item NFC scale [3] (implemented with Macromedia Authorware). Subjects rated statements on a Likert-scale (range 1-7). Score "1" meant 'strongly disagree' and score "7" 'strongly agree'.

Part 2: The Conference Planner Application

We developed an Open Source software application called "Conference Planner" which simulated the planning of speakers for a conference. The software (Macromedia Flash MX) logged all the moves participants made. The "Conference Planner" was developed by The Open University of The Netherlands and funded by the European UNFOLD Project [8]. It consists of four different components. The first one is the dynamic interface that shows each set of demands for a conference and allows the end-user to solve the problem in an easy way, based on drag & drop movements. The second one is the core of the application itself: the set of rules and related algorithms. Here is defined which actions are permitted, based on the requirements of the experiment and which are the subsequent consequences. The third component is a database, with all the scenarios used in the experiment. The forth component is the logging-module that writes all the clicks and drag & drop moves and their associated times, as well as waiting times to an external spreadsheet. The logs provide data for analysis of the results.

In the experiment, subjects had to solve 5 different conference scheduling situations. The conference speakers each had different demands, and they had to be scheduled into one of three rooms (each with its own facilities and availability). This type of task (being quite realistic compared to solving abstract puzzles) is not extremely difficult, but a certain approach is necessary to solve the situation in an efficient manner. When facing the problem, subjects can take multiple (correct) approaches to schedule the speakers. Unlike the abstract tasks used in previous research, such as the Tower of Hanoi or B&B, there are many more ways to optimally solve the scheduling task. Even with more correct solutions existing, without some degree of planning, the scheduling will not be optimal and extra moves (corrections on the assignments so far) will be needed. If the entire situation is not taken into account, the participant will get stuck in some later phase of the task because (s)he will encounter a speaker that does not fit in any of the slots that are left. Only by planning one will be able to fill the schedule without extra moves. A useful heuristic would be to look for the speakers with the most constraints and place them on the schedule first, matching their constraints with the constraints of the schedule slots. It can also be effective to start with identification of the slots that are hardest to fill (the ones unavailable to speakers needing 2 hours, limited number of people, and no beamer) and then assign speakers to those slots. An "easy" approach that does not involve much planning would be to start from the top of the list and work your way down, filling up the different slots. However, this approach is not so smart: you will find that some speakers cannot be placed anymore, after which speakers must be shifted around to make room. One can imagine still other strategies that do not involve a lot of planning, but strategies that are more like a trial-and-error approach (such as random selection of speakers).

The difference between externalization and internalization was implemented by highlighting all legal slots in the externalization condition (figure 1), where a person can be placed. In this condition, when one clicks on a speaker in the list on the left, the legal slots (those satisfying the constraints and being available) in the timetable turned green. Note that this does not show the *best* slot to place a speaker, but simply which slots are possible. To move a speaker from the left to a slot on the right, the little boxed icon in front of each speaker's name had to be "picked up" and "dragged" to its destination slot with the mouse.

Not all the timeslots in the grid are always available. Some where unavailable all the time, indicated with light-gray, for example the timeslots during lunchtime (13:00), but also some arbitrary other slots. The empty available timeslots were shown in white, and the ones that were already occupied by a speaker would display the name of a speaker.

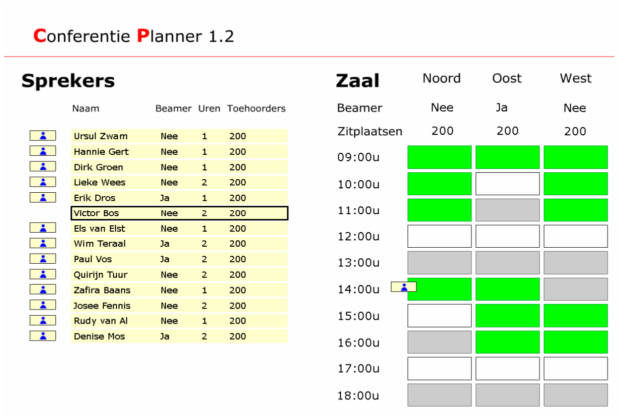

Figure 1: Conference Planner in the externalization condition: (legal timeslots turned green)

In the internalization condition the green feedback was absent, and one has to look up information and constraints by one self all the time (figure 2). No other differences existed between the two conditions.

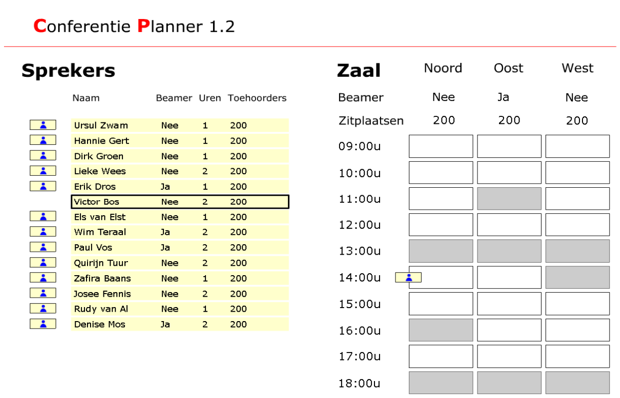

Figure 2: Conference Planner in the internalization condition

In both conditions, a list of speakers that had to be scheduled was given on the left. Each speaker had his/her own constraints displayed next to them, which could vary on a maximum of three variables:

- Number of hours they will speak (1 or 2, in Dutch: uren)
- Whether a beamer is needed (yes/no, in Dutch: ja/nee)
- Number of expected listeners (in Dutch: toehoorders)

The assignment of the participant was to place all the speakers on the schedule timetable, while taking the different constraints into account. A solution where each speaker was scheduled (and all the constraints were met) always existed. The participants had to perform 5 tasks in which the rooms, the speakers and their constraints varied.

Part 3: Post-experimental questionnaire

After completing the five trials of the conference-planning task, subjects were asked to complete a final questionnaire (implemented with Macromedia Authorware), measuring knowledge of the problem and subject's opinions.

Design

The experiment had a 2X2 design with two independent variables: interface style (internalization or externalization) and cognitive style (low or high NFC, based on the median). After the NFC-test, subjects were randomly assigned to INT/EXT. After 20 subjects, their NFC median was calculated. On this basis we labeled subjects high/low NFC. We checked whether low/high NFC subjects were evenly divided over INT/EXT. From then on NFC was checked immediately (via an automated process) and INT/EXT was continuously re-balanced to obtain equal groups. At the end the subjects per condition were: LowNFC-EXT:10 LowNFC-INT:11 HighNFC-EXT:11 HighNFC-INT:11.

Measures

In each of the three parts of the experiment we measured different dependent variables.

NFC-questionnaire

NFC-score: the average of the 18 answers (range 1-7).

Conference Planner: Time-based and move-based

- *Total time*: The average time needed to solve the tasks.
- *Time before first move*: The time between the moment the problem appears on-screen and the first move. It is an indicator for planning, telling how long subjects analyzed the problem before they started solving it.
- *Inter-move latency*: The time that passes between having placed a speaker, and picking up the next. We interpret this measure as a planning indicator.
- *Superfluous moves*: The problems have a shortest path solution, with an optimal amount of moves (speakers dragged from left to right) to solve them. E.g. in the case of a list with 15 speakers, the optimal solution would be 15 moves (placing all speakers correctly the first time). Superfluous moves are *all* the extra unnecessary actions besides this shortest path. We use this measure as the main performance measure, because it reflects the efficiency with which the task has been solved. This measure includes c*orrection moves*: all extra performed moves (more than the shortest path) that were made to "fix" non-optimal placements (a situation where one was stuck). Also *reconsidered moves* are among the superfluous moves: when subjects pick up speakers and while dragging change their minds and put him/her back before placement on the grid.

Conference Planner: Strategy Analysis
The video recordings of the subjects were analyzed. We looked (per task) at whether or not subjects started solving the problem with the best strategy, by first moving the speakers that had the most stringent constraints.

Post Experimental Questionnaire
The measures obtained from the post experimental questionnaire consist of answers to questions that tested for (a) declarative knowledge, (b) procedural knowledge, and (c) subject's opinions. For each essay question, a subject could receive a maximum of two points. Most of the knowledge questions explicitly asked for an explanation ("why?"), besides a "yes" or "no" answer. A picture of a situation as in the application they just worked with accompanied most of the essay questions. The opinion measures give us an idea of how the task was experienced and perceived by subjects themselves. Subjects could rate various statements on a scale from one to seven. The questionnaire consisted of three sets of questions:

- *Declarative knowledge*: Essay questions about *situations* that could be legal or illegal were shown, and one had to decide whether the situation was possible (could it occur regarding the constraints) and why (not).
- *Procedural knowledge:* Essay questions concerning solution procedures. Parts of situations were given and the questions were formulated as "how would you do this?" and "do you think this is a smart thing to do?"
- *Opinions:* Questions about opinions of subjects. There were 8 questions concerning perceived own problem solving and opinions about the application.

Procedure
After completing the NFC questionnaire, subjects received a textual introduction to the task explaining the general idea and saw a brief video-clip showing how the application and its controls worked. After this, subjects completed the 5 tasks. Finally the post-experimental questionnaire was completed.

RESULTS
We statistically analyzed the effects of interface style and cognitive style using ANOVA. We report on significant effects using a significance level of $p < 0.05$. Results with p-values between 0.05 and 0.10 are reported as tendencies. All the tasks were eventually solved correctly by all the subjects across conditions.

The Need for Cognition scale
The subjects scored a minimum of 2.39 and a maximum of 6.50 on the NFC scale. The mean score of the participants was 4.89 (SD 0.83), with the median at 5.06. Cronbach's alpha coefficient of the 18 statements was 0.89.

Time based results
Below are the results on the various time measures.

	Internalization				Externalization			
	Low NFC		High NFC		Low NFC		High NFC	
Average per task	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Total time	155.2	33.4	124.2	28.7	136.2	33.9	129.5	34.4
Time before first move	21.6	2.1	16.0	2.1	14.3	2.2	14.5	2.1
Inter move latency	5.3	1.7	4.3	1.0	4.2	1.4	3.7	1.1

Table 1: Average times (sec) for each condition

Total time
There were no significant main effects of interaction style or cognitive style on the average *total time* needed. However, we did find the tendency that low-NFC subjects needed more time $F(1,39)=3.58$, $p=0.07$, ($M=146$, $SD=34$ vs. $M=127$, $SD=31$). There was no significant interaction effect.

Time before first move
The main effect of interface style on the time that passed *before* subjects made their first move was significant $F(1,39)=4.34$, $p<0.05$. Internalization subjects took more time than externalization subjects, $M=18.9$, $SD=1.5$ vs. $M=14.4$, $SD=1.6$. Other effects were not significant.

Inter-move latency
There was also a significant main effect of interface style on the average time *between* moves $F(1,39)=4.82$, $p<0.05$. Internalization subjects took more time between moves, $M=4.8$, $SD=1.4$ vs. $M=3.9$, $SD=1.3$. Other effects were not significant.

Move based results
Below the results on the move based measures are shown.

	Internalization				Externalization			
	Low NFC		High NFC		Low NFC		High NFC	
Average per task	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Superfluous moves	2.9	2.8	2.1	2.3	4.1	2.1	4.4	3.9

Table 2: Average number of moves for each condition

Superfluous moves
There was a significant main effect of interface style on the number of superfluous moves that were made $F(1,39)=4.17$, $p<0.05$. Internalization subjects made fewer superfluous moves than externalization subjects, $M= 2.46$, $SD=0.61$ vs. $M=4.27$, $SD=0.63$. For cognitive style, there was no significant effect, nor an interaction effect.

Strategy analysis

Internalization subjects showed a tendency to use the 'most constraints first' strategy more often than externalization subjects ($F(1,39)=3.21$, $p=0.08$). Internalization subjects used that strategy 2.4 times (SD=1.76) out of 5 (tasks) whereas externalization subjects used it 1.5 times out of 5 (SD=1.5). There was neither a significant main effect nor interaction effect in relation to cognitive style.

Results from the post experimental questionnaire

Declarative knowledge

The effect of interface style on answers to the declarative knowledge questions was almost significant at $F(1,38)=3.73$, $p=0.06$. Internalization subjects answered more of those questions correctly than externalization subjects (M=8.0, SD=0.2 vs. M=7.7, SD=0.7). No significant effects of cognitive style were found.

Procedural knowledge

Neither interface style nor cognitive style had an effect on procedural knowledge. There were no interaction effects.

Opinions

On all of the questions internalization subjects scored marginally higher, but only one of these differences was significant, namely whether the subjects "sometimes did not know how to proceed with the arrangement of the speakers", $F(1,39)=5.91$, $p<0.02$. Externalization subjects scored significantly lower (M=5.24, SD =1.61) than subjects in the internalization condition (M= 6.23, SD=0.97), meaning that they had this feeling more often.

	Question	Internaliz.		Externaliz.	
		Mean	**SD**	**Mean**	**SD**
1	I always looked how I should solve a task and then started working on it.	5.73	1.24	5.19	1.03
2	I found the five tasks difficult to solve.*	6.23	0.81	5.90	1.22
3	I think I succeeded in solving the tasks with as little moves as possible.	5.50	1.37	5.10	1.55
4	I tried to think ahead about my steps as much as possible.	5.05	1.68	4.67	1.39
5	Sometimes I did not know how to proceed with the arrangement of the speakers.*	6.23	0.97	5.24	1.61
6	The controls to use this program are pleasant.	6.09	0.81	6.00	0.78
7	I often felt like I got stuck and could not find the solution.*	6.73	0.63	6.29	1.23
8	I always knew for sure whether I could place a speaker on a certain location.	5.55	1.54	5.52	1.25

* This item's score was reversed to the same direction as the others

Table 3: Average opinion scores per interface style

CONCLUSION AND DISCUSSION

This research investigated the influences that interface style and cognitive style have on planful behavior from the user and consequently on problem solving performance. These issues are beyond plain usability issues, and instead we focus on more (meta) cognitive aspects of interface-induced behavior, namely planfulness and user engagement.

Our first hypothesis stating that internalization leads to more planning and better performance than externalization is supported both by time-based and move-based measures. As in earlier experiments [21, 21, 22], it once again showed that (now also in this more realistic task) user behavior differed depending on interface style.

The time-based measures showed that the interface style requiring (more) internalization resulted in longer thinking times before subjects started working on the problem, and also more time between moves, which we took as indicators of planning. Both these results indicate that when information has to be internalized, more contemplation from the user is provoked. This is in line with results of O'Hara and Payne [13], who reported a longer inter-move latency for subjects in the effortful condition, indicating a more plan-based approach. Also interesting are the conclusions that can be drawn from the move-based measures. It has to be born in mind that the issue here was not "can they solve it?" but "how smart or efficiently do they solve it?", since in the end each problem had a solution, and these tasks are not extremely difficult given the limited number of speakers. Results showed that internalization subjects solved the problems with fewer superfluous moves, in a more straightforward manner, thus with less deviation from the minimum amount of moves, resulting in greater efficiency. We infer that the shorter solution paths are a result stemming from better planning. These superfluous moves included "correction moves", moves made to "repair" situations that sometimes were created by users making "not so smart" moves. Superfluous moves also included "reconsidered moves" (when a user started a move but changed his mind and puts it back, like a chess player who realizes his mistake a fraction after he picks up a piece). These findings converge with our earlier results [20, 21, 22] in which there were strong tendencies in that direction. The results are also in line with O'Hara and Payne [13, 14] who found that the subjects with a plan-based approach used fewer moves than subjects using a display-based approach. They also stated that "backtracking" (undoing a move and return to previous situation) occurred more in subjects showing display-based problem solving.

In sum, although there was no significant difference in the *total* time taken to complete the tasks, there were differences in *how efficient, in which manner* subjects solve the problems. All subjects needed about the same total amount of time, but subjects who were required to internalize thought longer before moves and came up with more efficient solutions. This converges with results from other research [7, 13, 14, 20, 21, 22]. The results reflect a trade-off between time spent thinking and planning (internalization subjects) and time spent making superfluous moves (externalization subjects). The fact that internalization subjects came out of the experiment with better declarative knowledge strengthens our interpretation that these subjects spend the extra thinking time on planning and learning. All this might be of value in for example a transfer task, or when doing a similar task after a big lapse of time.

This interpretation also gets some support from internalization subjects' tendency to use a more appropriate strategy. Regarding the strategy that subjects chose to use, a qualitative analysis of the results pointed at a more plan-based approach by the internalization subjects, showing that those subjects more often filled the timetable by first scheduling speakers with the most constraints. This strategy is again an indicator of good planning, because it shows that people think about whom, and how they are going to schedule *before* starting with the task. The issue of *strategy use* could merit more interest in future research. Also, it would be interesting to investigate whether subjects stick to the same strategy or change it, if they feel it is unproductive or if another less effortful strategy suffices. Another interesting question would be whether strategy choice can be influenced (e.g., by providing feedback, or externalizing strategy support, as mentioned in [16]).

The second hypothesis has to be rejected. Cognitive style, at least along our dimension of high NFC versus low NFC had no influence; pre-existing attitudes towards problem solving did not have significant effects on the displayed behavior and performance of subjects. Perhaps the strong effects of interface style overruled possible effects of pre-existing individual differences in cognitive style. However, the lack of findings may also be due to our subjects' high NFC scores. The difference in scores between high-NFC and low-NFC may not have been large enough to produce significant results. Another possible reason for the lack of findings is that our task might have been too easy for the subjects, not needing the extra impetus from having high NFC-scores. This is supported by answers in the post experimental questionnaire concerning the perceived difficulty. Subjects reported having little difficulty with the tasks, and the cognitive effort required was quite low.

We also tested the knowledge subjects possessed after completing the tasks. In the declarative knowledge questions, subjects had to judge situations where the rules sometimes were violated. They had to identify whether the shown situations were theoretically possible or not. They had to look at the constraints of a situation, and decide whether any "rule" was violated (such as a speaker with an audience of 120 placed in a room where only 75 fitted). The expected effect of interface style on declarative knowledge questions was almost significant, although not as strong as in earlier experiments with B&B [20, 21, 22]. It showed that internalization subjects tended to give more correct answers to those questions. An explanation for the better answers by internalization subjects could be the fact that in all the tasks before, they proved to have thought and planned more. Perhaps a training effect occurred that made them more accurate on these questions too. There were also several procedural knowledge questions. These were not so much about judgment, but about insight on what to do to solve a part of a problem. As expected, interface style had no effect here, no differences concerning answers to the procedural questions were found (performance was nearly at ceiling). In a way this is not very surprising, since the problems were not so difficult, and we also saw that all subjects correctly solved al the situations. This pattern is similar to the one found in previous experiments: a positive effect of internalization on declarative knowledge, and no effect on procedural knowledge [20, 21, 22].

Lastly, the opinion questions indicated that all the subjects had little difficulty performing the tasks and were confident about their own performance. There were hardly any differences between the two interface styles, although the scores of internalization subjects were marginally higher. One question yielded significantly different scores, pointing at an advantage of internalization. We asked whether the subjects sometimes did not know how to proceed with the arrangement of the speakers. This "feeling stuck" occurred significantly more often in the externalization condition than in the internalization condition. This lends support to the finding that externalization subjects needed more superfluous moves (they had to make corrections more often) to complete the tasks. We take this also as an indicator that the internalization subjects had better insight.

To sum up our conclusions, at not any point or with any measure did externalization result in better performance, reconfirming various earlier findings. Also in a more realistic task, we found positive effects of *internalization* on problem-solving behavior: having the user internalize the information leads to a more plan-based behavior, smarter solution paths, better declarative knowledge, and less feeling lost. Externalization on the other hand led to a more display-based approach, resulting in less efficient solutions (more moves) and less thinking about moves to be made.

It is worthwhile to reflect on what was in effect externalized and visualized. In the externalization condition this was the result of applying the set of constraints that was attached to the speaker, to the sets of constraints attached to the scheduling slots. The interface showed which actions are allowed with the object at hand. Subjects were able to see the outcome of the *application* of the rules. This is comparable to the graying out of menu items in for example Microsoft Word, showing that the current situation does not permit those functions to be used. We showed that this widely accepted and implemented guideline does sometimes have undesirable effects. In the real world, when designing interfaces we have to be careful with providing interface cues that give away too much, and must design in such a manner that the way users think is optimally supported. Interaction designers could consider making interactions for a user "more difficult", or better "less assisted" on purpose, to persuade users to adopt such specific behavior. This in turn could help the software to achieve its specific goal. How can we link all this to general aspects of usability such as learnability, satisfaction, errors/memorability and efficiency [12]? In previous research by the authors we saw that errors/memorability (especially memorability) was better in the internalization condition [20, 21]. Concerning errors, the current results concerning superfluous moves (although they are not the same as errors) point at an advantage for internalization. Efficiency in terms of "path economy" was better in the situations where subjects had to internalize information themselves. Satisfaction was not our main focus, but there was a question (question 6, Table 3) reflecting satisfaction, and there was no difference between the interfaces.

We feel that understanding how people react to interface information (based on cognitive findings) is one of the main challenges in HCI research. Our research shows that a computer mediated task can take advantage of interfaces that are designed from considerations that run deeper than plain usability. With all kinds of multimedia present in all corners of society, our research contributes to building bridges between cognitive science, HCI and current educational practices. Our findings, especially if extended to even more realistic tasks can be valuable in the development of applications in the realm of education, multimedia learning or game based learning, and still others. Examples are interactive learning devices in museums that try to explain certain principles, but also more generally other situations where learning as good as possible or when certain issues are at stake, and making as little mistakes as possible or speed are important. There are of course situations in which the investigated issues are not crucial at all, but, depending on the specific goal and situation, designing software in such a way that active learning is being provoked can be valuable. We expect that having knowledge internalized can be important when dependence on the existence of a particular interface is specifically *not* desired, or when speeding up a task is important, just to name a few situations. One can think of situations where risky and complex tasks are performed, and where a user suddenly is required to a new situation. Now his/her insights and knowledge has to be transferred to a different situation. The common guideline to "not give users the chance to make mistakes" should of course not be neglected, but at the same time, interaction should facilitate or even persuade users to learn what underlies the task they are doing. The investigated concepts are important in situations where learning itself is the aim, but one can also think of situations where making errors generates a high cost. An example would be certain types of medical software, where it is important that users master underlying concepts at all times, and are continuously provoked to "actively think". Likewise, one can also think of process operators in for example a hydroelectric power plant. They use process control software that will assist and prevent errors, especially with crucial or difficult tasks. If the system would break down and operators have to manually do the job that is normally done aided by computers, will they still know what exactly underlies the task? Examples are actions that must be performed in a certain order, such as opening valves, closing a circuit and so on. Or are they so used to the information and feedback from the interface that they will now be stuck? In this light, Fu and Gray [9] found that when inefficient procedures are chosen to solve a problem, these procedures shared two characteristics. They are often generic and well-practiced, but more importantly the procedure is composed of interactive components that bring fast, incremental feedback on the external problem states. These actions require less cognitive effort, but this bias to depend on interactive units unfortunately leads to paths that require more effort in the long run.

The title of this paper is of course a wink to the famous book chapter by Carroll and Rosson [5] entitled "The paradox of the active user". This term refers to a common observation that users never read manuals but start using software immediately. They are not very interested in the system, and don't want to spend time up front on getting established, or going through learning packages. It is a paradox because users would save time in the long run by taking some time to optimize the system and learn more about it. The paradox in our research would be that users that received an interface providing some assistance come up with less smart solutions, among other things. This assistance from the interface of course is meant to help, but appears to be counter effective. The conclusion of the two paradoxes could be the same: depending on the purpose of the software, we must perhaps design for the way users think and behave in reality.

FUTURE RESEARCH

We saw that tweaking a small part of the interface had impact on the efficiency and planfulness of user behavior. Externalization and internalization as we implemented it are not extremes on a continuum; they rather refer to the *amount* of externalization. Our two interface conditions could have represented different values on that continuum, and the difference between them could have been smaller or bigger. It would be interesting to see the effects of adaptively varying the amount of externalization over time, based on user's performance. Besides the amount of externalization, also the type of task can be varied. We also intend to look at in how far findings can be generalized to other, even more realistic task domains. The issues of interruptions and distractions deserve attention as well, especially interaction paradigms in a nowadays-prevalent area: mobile devices. Our devices are more mobile than ever before and during interactions with them a wide range of interruptions and distractions are commonplace. Typical interruptions of our modern lives are getting in/out of a taxi, reaching a subway station or temporary losing your wireless connection. One could argue that it is preferable that users resume interacting on the basis of solid internalized plans and knowledge, rather than catching up on the task on the basis of an interface assisting the user to a high degree. If the former is the case, it means that users still need solid plans. The assistance as we implemented it is only one way to externalize information, and it will be interesting to look into other ways. We will continue the research and broaden the types of problem solving activities, and include other variables related to attitudes to problem solving.

ACKNOWLEDGEMENTS

We thank Information Science students Imke de Jong, Marijn Plomp & Jasper Roques for their assistance.

REFERENCES

1. Brusilovsky, P. Adaptive hypermedia. *User Modeling and User-Adapted Interaction, 11* (2001), 87-110.
2. Cacioppo, J.T., and Petty, R.E. The need for cognition. *Journal of Personality and Social Psychology, 42* (1982), 116-131.
3. Cacioppo, J.T., Petty, R.E. and Kao, C.E. The efficient assessment of need for cognition. *Journal of Personality Assessment, 48* (1984), 306-307.
4. Carrol, J.M. *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. MIT Press, Cambridge, MA, 1990.
5. Carroll, J.M. and Rosson, M.B. The paradox of the active user. In J.M. Carroll (Ed.), *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. MIT Press, Cambridge, MA, USA, 1987.
6. Crystal, A. and Kalyanaraman, S. Personality variables and information processing on the web: Methodological issues and empirical evidence. *Paper presented at the International Communication Association Annual Meeting* (2005), 26-30.
7. De Mul, S. and Van Oostendorp, H. Learning user interfaces by exploration. *Acta Psychologica, 91* (1996), 325-344.
8. European UNFOLD Project. http://www.unfold-project.net
9. Fu, W. and Gray, W.D. Resolving the paradox of the active user: stable suboptimal performance in interactive tasks. *Cognitive Science, 28* (2004), 901-935.
10. Kort, B., Reilly, R. and Picard, R. External representation of learning process and domain knowledge: Affective state as a determinate of its structure and function. *Paper presented at the 10th International Conference on Artificial Intelligence in Education, San Antonio, TX* (2001).
11. Mayes, J. T., Draper, S.W., McGregor, A.M. and Oatley, K. Information flow in a user interface: the effect of experience and context on the recall of MacWrite screens. In *Proc. of the 4th Conference of the British Computer Society on People and computers* (1988), 275-289.
12. Nielsen, J. *Usability Engineering*. Morgan Kaufmann, San Francisco, CA, USA, 1994.
13. O'Hara, K.P. and Payne, S.J. The effects of operator implementation cost on planfulness of problem solving and learning. *Cognitive Psychology, 35*, 1 (1998), 34-70.
14. O'Hara, K.P. and Payne, S.J. Planning and the user interface: The effects of lockout time and error recovery cost. *International Journal of Human-Computer Studies, 50*, 1 (1999), 41-59.
15. Palmiter, S. and Elkerton, J. An evaluation of demonstrations for learning computer based tasks. In *Proc. of the SIGCHI conference on Human factors in computing systems: Reaching through technology* (1991), 257-263.
16. Payne S.J., Howes, A. and Reader, W.R. Adaptively distributing cognition: A decision-making perspective on human-computer interaction. *Behaviour & Information Technology, 20*, 5 (2001), 339-346.
17. Scaife, M. and Rogers, Y. External cognition: How do graphical representations work? *International Journal of Human-Computer Studies, 45*, 2 (1996), 185-213.
18. Svendsen, G.B. The influence of interface style on problem solving. *International Journal for Man-Machine Studies, 35* (1991), 379-397.
19. Tognazzini, B. Ten Most Wanted Design Bugs (2004). http://www.asktog.com/Bughouse/10MostWantedDesignBugs.html
20. Van Nimwegen, C., Van Oostendorp, H. and Tabachneck-Schijf, H.J.M. Does context sensitivity in the interface help? In *Proc. of the Twelfth European Conference on Cognitive Ergonomics,* European Association of Cognitive Ergonomics (2004), 29-33.
21. Van Nimwegen, C., Van Oostendorp, H. and Tabachneck-Schijf, H.J.M. The over-assisting interface: The role of support in knowledge acquisition. In *Proc. of HCI2004: Design for Life,* Research Press International (2004), 133-136.
22. Van Nimwegen, C., Van Oostendorp, H. and Tabachneck-Schijf, H.J.M. The role of interface style in planning during problem solving. *In Proc. The 27th Annual Cognitive Science Conference*, Lawrence Erlbaum (2005), 2771-2776.
23. Van Oostendorp, H. and De Mul, S. Learning by exploration: Thinking aloud while exploring an information system. *Instructional Science, 27* (1999), 269-284.
24. Zhang, J., and Norman, D. A. Representations in distributed cognitive tasks. *Cognitive Science, 18* (1994), 87-122.