NEW BULGARIAN UNIVERSITY

Central and Eastern European Center for Cognitive Science

COGS MoStaCon

Usability study for an experiment-design tool

Radu Luchianov

A thesis submitted in partial fulfillment of the requirements for the degree of M.S. in Cognitive Science

Supervisor: Prof. Stefan Mateeff

Sofia, June 2000

Abstract

This paper contains a discussion on cognitively accessible human-computer interfaces and presents a usability study which compares expert and beginner performance in working with MoStaCon (an application of the principles discussed) and other existing statistical tools. Following the results on **user-adaptive navigation** presented in a series of papers culminating in Brusilovsky, P., Eklund, J. (1999), my study aims to explore **additional cognitive principles** (gathered by the author of this paper under the name of MonDoc). The implementation of these principles, with the addition of **managerial techniques** (MoStaCon), could speed up and make the work of empirical researchers more error-proof. The participants in this study were an international group of students and researchers. The results demonstrate that two of the additional principles (**emphasis** and **active-oriented navigation**), verified on a prototype of MoStaCon, significantly affect learning, accuracy and the speed of design decisions.

The working version is on-line at:

http://www.nbu.bg/cogs/personal/radu/proj/msc/COG699.pdf

The final version will be available at:

http://www.nbu.bg/cogs/personal/radu/port/cogs/cog699.html

Contents

1. Introduction 4
1.1. Overview of some technical problems in empirical research. Goals of the thesis. 8
1.2. Cognitive and psychophysical principles used in MoStaCon to alleviate these problems. 10
1.3. Summary 35
2. Testing MoStaCon: usability study 36
2.1. Preliminary survey and Discussion. 37
2.2. Task, conditions and stimuli 38
2.3. Measures, dependent and independent variables 41
2.4. Results of MoStaCon vs. standard research 43
2.5. The effect of graphical emphasis 45
2.6. The effect of action-oriented navigation 46
2.7. Discussion of the study 47
3. General discussion and conclusions 49
Glossary
References
Appendix 1 commented screenshots: the elements of the prototype and example of StaCon filter results
Appendix 2 example of the two versions of the interface used in the usability study
Appendix 3 instructions for the survey and for the usability study
Appendix 4 list of empirical research steps

1. Introduction

The paper you are about to read is not a study of a specific, specialized, synthesized and spelled-out process. I am not going to go in the deep details of a cognitive process as the others of my colleagues did before and will surely do after me. Rather, I will present the **application** of many useful, disparate principles I have learned by and large since I was born and specifically since I started studying Cognitive Science. It is an application which I hope will help many people in the 'research trade' - and not only - to **achieve** their goals faster and more accurately. Additionally, this very paper was developed using the presentation techniques I describe in it.

1.0.1. What is MoStaCon? Why? How?

MoStaCon is a modular, easily-modifiable research tool and tutorial; the two main modules are the MOnitor and StaCon. The MOnitor is a research-**management information system** that includes a **primer in empirical research** (mostly experimental psychology and psycholinguistics, for the version under study). StaCon is a modular visual **interface for visual programming** of data filters and analyzers, paired with a **library of the functions** available in the interface. The MOnitor is the first application of **MonDoc**, another project of mine which is a set of cognitive principles (between a model and a theory) embodied in an **engine, document format and editor** for cognitively accessible documents (see section 1.2.1. on page 17 for a description of redundant hypertext).

1.0.2. Goal of the thesis

Since a usability study has the goal of **testing** a software tool, I test two hypotheses in several experiments presented in chapter 3. As a summary of these hypotheses, the main goal of the present paper is **to verify whether the implementation of the MonDoc principles, MoStaCon, is indeed positively affecting user performance.** In the analysis included in Chapter 2, I do a general evaluation of researcher performance while using MoStaCon versus their performance when following their usual strategy, then I verify the individual effect of two of the MonDoc principles on user performance.

1.0.3. Summary of the paper

One can look at the system I am describing in this paper on several levels.

In this introduction, I am presenting the **Idea**, the **Software** and a **Disclaimer** with pointers to some other similar existing software. I also detail my **Thanks** to some of the people who helped me, directly or indirectly, during the development of this paper and of the software this paper presents.

Further (section 1.1), I **outline the technical problems** which prompted me to start developing the system. In the last introductory section (1.2), I **present the cognitive principles** which I currently call MonDoc, their theoretical provenience and the resulting features of MoStaCon; I hope I managed to convey how this interface came up in the interplay of several distinct, but by no means separate, domains summed under the umbrella of Cognitive Science.

The second chapter is a presentation of the **usability study** I ran in order to assess how researchers react when faced with an actual implementation of the theoretical principles discussed in the first chapter.

In the last chapter I discuss the **results of the study**, in the context of other existing software, I **evaluate the fidelity of implementation** of the MoStaCon principles in the interface features and I list **planned features** and planned modifications to existing features.

If you find anywhere in the paper **technical terms** not explained immediately before or after their first mentioning, you mat take a look at the included Glossary.

Appendix 1 is the description of **the prototype MoStaCon** (the Console, MOnitor and StaCon interfaces), used in the study. It also contains the description of the results from a StaCon filter.

Appendix 2 contains two **example pages** from the 6 modified for use in the experiment on emphasis.

Appendix 3 contains the **instructions** I used for the subjects in the pilot study and in the usability study described in the fourth chapter.

Appendix 4 is the **list of empirical research steps**, as compiled from Elmes, D.G. et. al. (1992), and augmented from the audit trails of the survey. This list was used to calculate subjects' accuracy in design during the usability study.

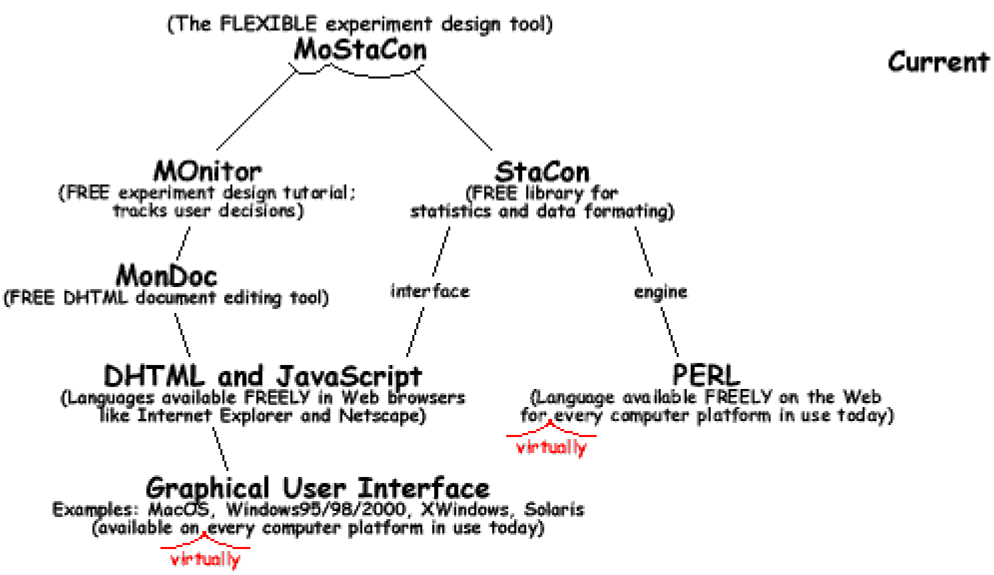
And now we start with the idea: the originator of reason… and the reasons for MoStaCon's existence.

Idea.

MoStaCon is the implementation of **a collection of principles** which I gathered during my experience as lab assistant (dealing with software support), about how people (both beginners and experts in the respective field), use software. During that time I was constantly relating the **observations** I was making with my previous experience. As a programmer, I had experience in **relational databases** (RDBs), **formal languages, project management**, and **software engineering**. During my research assistantship at NBU I was also taking courses in **physiology**, **perception**, **thinking and reasoning**, **experiment design**, and I was helping colleagues with **statistics**. All this experience, coupled with the material from my courses and my own interest in **automated learning** and **human-computer interfaces** (HCI), prompted several hypotheses about how a **user interface** may be done so the use of software is **less demanding** (on the perceptual and cognitive capacity of the user), and **more streamlined** (allowing to achieve targeted goals quicker). I have observed for close to three years the **kinds of errors** that researchers of all levels (including myself), make while designing experiments and interpreting their results. During the same time, I fought with all my programming abilities the **bottlenecks** which tend to crop up at specific points in the research, points which may be irrelevant to the research itself, but are decisive to the conclusions and (re)usability of the research. Points like **data formatting**, **result interpretation**, **generating reports** (writing papers, presentations, etc.) and **archiving**. The idea with which I put it all together is a **Moral** one, though. It is that of the freedom of knowledge. I believe that scientific advances are speeded up by **open collaboration** within the scientific community and are hindered by things like idea ownership and brutally law-enforced software copyrighting. I totally embrace the need to **give credit** where it's due, embodied in the original literature copyright laws, but forcing people to pay for products they may not use or limiting development in some area to a select group of scientists seems unfair. For this reason, MoStaCon is a FREE tool developed in FREE environments (NoteTab Light and HYMNS, then MonDoc) and running its OPEN scripts (see Software below) on virtually any platform on FREE interpreters (PERL, Netscape, Internet Explorer). For more on the moral topic see Luchianov, R. (1995).

Software.

Mentioning applications and programming languages brings about the point that MoStaCon is a software tool. More important, it is a **modular** one. The core modules are a shell for developing the rest of the system itself. I'm trying to make it as **flexible** as possible so that even non-programmers could make additions and modifications to the system and securely alter it to fit their own purposes. The system architecture is **open**, in three ways: (1) easily **updateable**, as mentioned above, (2) **transparent** to the user (allows the user to know at any time what's happening to the data which is being stored in or processed by the system) and (3) **free** for use and expanding by anyone who finds a use for it. There are three parts to this tool, as its name suggests: **MOnitor**, STAtistics and data CONversion. STA and CON go together in the same module which I will call **StaCon** from now on.The underlying principles and the description of their implementation as the MOnitor interface are the subject of Chapter 2.



As you can see in the image above, MoStaCon is based on **freely accessible tools**, some of my own creation (1-5 years old) and some with a long history (5-8 years) in the shareware and freeware domains. The shell of MoStaCon, the MOnitor, is a primer graph developed with MonDoc on which I added path tracking through an embedded DM graph. All actions the user takes (including the path they follow through the problem-space of experiment design and other decisions) are tracked by the system and can be recorded as a text file.

A practical note: the **size** of MoStaCon is insignificant, as compared with other systems in the same class. Everything needed for its installation can be downloaded as a ZIP archive which fits comfortably on a 1.44M floppy disk (including the `perl` interpreter and the optional web server). It is also **compact**; the whole page graph can be included in one file, together with the MonDoc engine, instead of the tens or hundreds of DHTML pages one would expect.

Tools equivalent to MoStaCon and compiled as executables (with BC, VC++, VBasic and even Java), can contain hidden malicious software (viruses, worms, Trojan horses); MoStaCon uses the **best**

security available on-line, the one provided by the leading browsers, whose teams of programmers do their best to fill up every hole a hacker can find.

Other **experimental tools** on which I could find any literature on-line, (see Eklund, J. et.al. (1997)) tend to be conservative in presentation and accept the principles of hypertext as implemented almost 10 years ago. I discuss these principles in more detail and I compare MoStaCon with one of the other existing experimental tools later, in Chapter 2.

Not on the last place, I considered the portability of the interpreters: Perl has very good ports on the three platforms we use in the COGS Department of NBU (MacOS, DOS/Windows and UNIX/Solaris). JavaScript is included in the major Web browsers so it can also be found on all the mentioned platforms the only things which may very from browser to browser and from platform to platform is the 'look' of the pages (since browser implementation varies) and some functionality (since Internet Explorer and Netscape have different implementations of the language.)

1.0.2. What MoStaCon is NOT?

I am NOT trying to compete with commercial statistical packages like Statistica or SyStat. I currently don't implement graphical output and I don't think I will, unless more than 50 users will demand it. Statistica is a great tool for that purpose. I don't intend to add complex modeling modules or intricate algebraic methods. Use MathCAD or Matlab if you need that. I'm not concerned with the newest and fanciest statistical methods (unless I come to use some of them in my later research), nor do I provide extensive database integration or programming languages. SyStat is the best candidate in that case, and Statistica is not far behind. Finally, I am not after any poor student's money and I require all prospective distributors, packagers, etc., to refrain from making any profit out of it. MoStaCon is NOT a commercial product and cannot be sold by anyone under any circumstances. You may check the Reference section for on-line pointers to some of the software mentioned here.

In short, I don't try to implement everything possible in MoStaCon; Its purpose is to be a goal-directed tool, which may be **developed** by a programmer to speed up research for an individual, a group or a laboratory, and can be **modified** by non-programmers as a **dynamic repository** of research information.

1.0.3. Thank you

The completion of this paper is not due entirely to me. **I thank many people** for helping me, including **Stefan Mateeff** (for coordinating my efforts, structuring my work -- and reminding me of deadlines!), **Elena Andonova** (for support and understanding), **Boicho Kokinov** (for making it possible for me to study Cognitive Science at the best international level and for suggesting the idea of the experiment presented herein), **Encho Gerganov**, (for his explanations during the course in experiment design and the choice of textbook), **Angel Vassilev**, **Naum Yakimoff** and the other professors in the Cognitive Science Department (for their inspired presentations of course materials), **Svetlana Popova** and **Irina Gerdjikova** (instrumental in finding participants to balance the sample for the experiment) **Vassil Pavlov** (for helping with translations); to **the participants in the experiment** (for their time, ideas and for beta-testing MoStaCon); to **Larry Wall** (for creating PERL), **Mark Andersen** (for creating Mosaic, then Netscape and pioneering JavaScript), the **Microsoft JScript** team (for including in JavaScript good ideas from PERL and for making

DHTML less strict), to **the people who wrote the great books and articles** listed in this document's Reference, to the **public domain, freeware** and **shareware communities** (for the wonderful free tools they allowed me to use), and happily, to my wife **Irena** and to our newborn **Christopher** (for moral support and understanding my responsibility as a researcher.)

1.0.4. Disclaimer

I wrote MoStaCon as a part of my MonEx project (of a WWW-based on-line experiment-creation tool). MoStaCon is a **student-created tool**, and as such, it does not claim to be professional in its performance. I guarantee the quality of the results (up to the precision allowed by the implementation of PERL the user installs and limited by the understanding the user has of the methods s/he employs), but I'm not (much) concerned with the speed of processing or with the resource-efficiency of the algorythms I use. I'm writing it for my own use, but I provide it to the research community, FREE, with a very low level of support (I'm reserving the right to change this policy at any time in order to preserve under the letter of any restrictive law-to-come, the spirit of the Moral idea described above.)

1.1. Overview of some technical problems in empirical research; goals of the thesis.

There are **philosophical problems** with empirical research such as (1) the impossibility to describe single-occurrence or low-frequency events or (2) the correctness of predictions for events which take place in a chaotic environment (which our Universe seems to be). However, in this paper I'm not going to discuss these. Unlike Hume ☺, I'm (still) inclined to **believe** that observed systems continue to behave with an approximation of the observed regularity even after direct observation ceases, if the conditions of the system are in the previously observed range. Since empirical research is the only method that I know of which allows us to make grounded and dependable approximate predictions (short of the semi-magical mathematical models of chaos), I am doing my research following the empirical methods of research.

In this paper I will also not be concerned with the **conceptual problems** of empirical research. For the most experienced researcher, just as for the innocent philosopher, it is difficult to keep track of all the **constraints** which mar the challenging pleasure of "answering questions or describing processes through empirical methods."

So what remains for me to discuss? Speaking generally, I can isolate three different sources of **technical problems**.

Problem #1: The amount of **experience** of the researcher **doesn't fit the task.** It may be **too detailed** and specialized, generating fixations (paradigm-addiction :) and automatism, **or too little**, as in undergraduate students, in this case generating frequent errors of design and implementation and redundant work. There are several good textbooks treating the matter of empirical research, many articles and even journals dedicated to this topic. Still, in my opinion, the amount of information that needs to be **perceived**, **retrieved** from memory, **integrated**, **compared**, and **applied** to the analysis under study exceeds the capacity of the human mind. Also, keeping track of the newest methods and models is no easy task.

Traditionally, researchers solve this problem by adhering to methods which in their experience (and their school of thought), have turned out to be successful in modeling the processes they research. As a result, as new schools of thought on specific topics appear, and as research gets closer and closer in depth to the limits of the empirical method, the available explanations for observed events expands. At the same time, terminology gets denser and stuffy with annotations which specify, among other things, limits of the term, its accuracy, and even 'copyright'. This specialization in the analysis makes synthesis (drawing conclusions and making generalizations and predictions), complicated and sometimes impossible And after all, the goal of research are synthesized results.

So what happens is that in order to design an experiment, the researcher writes an initial proposal, then considers it for some time, refines it, as the goal of the experiment becomes more specific, and as specific necessary constraints are evaluated (measures, methods of measurement, control conditions). During this process, the researcher uses the literature intensively, consults colleagues and/or experts in the field, and does surveys and pilot studies to check for practical feasibility. All this process is very information-intensive and in this paper I will address ways (mostly **the MonDoc principles**), in which that information can be **accessible** from a high-level perception point of view, **structured** and recorded in order to supplement memory recall and to **simplify summarizing** large quantities of data (both theoretical and empirical)

> Problem #2: The **planning is sloppy**. The dependencies inherent in any research method (what should happen before and after each step in a method, coordinating the schedules of a team or time-tables of participants and the conditions under which they have to be observed, etc.), make planning very difficult and error-prone. Also, handling intricate and large amounts of data correctly is a rare skill, requiring attention and a detail-oriented mind.

Organization is very important in any rational human activity. In competitive fields like software engineering, there are methods for global planning, making dependencies obvious, schematic representation and so forth. These methods can be used to record, automate and present for a 'manager' accurate current status, summaries of previous activity and if necessary predictions of future results for alternative current approaches. Most of these methods are using results coming from research on cognitive topics. It seems paradoxical that most researchers (which I met), prefer to rely on their experience and paper-and-pencil organization, treating computers and software as some sort of corruptive influence, to be used grudgingly if no other choice but hard (and error-prone) manual treatment of data. Part of this paper deals with ways (mostly **the MoStaCon implementation**), in which theoretical project management can be applied to **planning** experiments, **keeping track** of key information during the design process, **limiting processing errors** by automatically applying some general constraints on the procedure and the data

> Problem #3: **Unpredicted external events** prevent the successful completion of some intermediary step(s). I can't actually criticize this point, because I can't make it up with constructive ideas later on.

So if I don't have a plausible documented opinion, I think I'm entitled to note some introspection and informal observations. I noticed that people whom I consider **wise** deal with the inevitable by raising shoulders and paraphrasing "c'est la vie!" On the other hand, I keep hearing many people informally stating some Murphy-type pessimistic prediction like "research-related work always takes more than you think it will" In my (untested) opinion, the latter group of people confound the inevitable of the third general problem presented above with the other two types of problems. This strikes me as an escapist, negative type of thinking; the fact that researchers think this way may make them at an unconscious level, less critical of their own errors. It's easy to say that "problems are", but problems should be fought, not only accepted. Hopefully, **the implementation** of the MonDoc and MoStaCon principles will be a step in the direction of separating the chaff of uncontrollable external events from the grain of research.

Unfortunately, in most cases of experiments I participated in, including the one I'm presenting in this paper, all three reasons hold true. Here are the principles I've been mentioning and their implementation:

1.2. Cognitive and psychophysical principles which MoStaCon uses to alleviate these problems

It is in the nature of Cognitive Science, which spans a diversity of fields of study, to be **synthetic**, so very close to the motto of New Bulgarian University: "Ne Varietatem Timeamus" The section divisions I make may be contested by some readers; this is just a result of the overlapping which exists between the fields of Cognitive research. I will begin by **structuring the literature** I consciously used in creating MoStaCon and MonDoc and in writing this paper (I am aware that I also used lots of other material unconsciously but un/fortunately I can't tell that material apart from my stream of consciousness). Some of the **research** I present here might seem to you ancient history, or the 'ABC' of your field, while with other entire topics you might not be familiar at all. I'm explaining every concept in order for the paper to be understandable by people from all the fields involved and even to non-specialists. (I don't believe in using obscure **terminology**: it gives scientists a bad reputation.)

The structure of this section will be done according to cognitive and psychophysical **principles** partially implemented in the document-presentation engine and the editors available throughout MoStaCon. And 'partially' is a key word; the current implementation is not very accurate because I wanted to check **whether such a tool would be necessary**, before continuing its development. You should expect a **discussion** of a specific principle, a **description** of how the principle is implemented (or planned to be), then the same treatment for the next principle. After that, I'll take a look at the ways in which other researchers put together some of these principles in **theories and user models** (the field of Human-Computer Interaction). Finally, I will discuss some of the existing implementations of these principles in software I worked with and I will discuss the worst practical problem that I see in HCI studies: **cultural-centrist** models.

If at any time you need an example of what I'm talking about in this section, you may check Appendix 2 for an illustrated description of the MoStaCon prototype on which I ran the usability study. Even if in this chapter I just skim the surface of the theory available on usability studies, you may expect a relatively long discussion.

1.2.1. MonDoc principles

As I mentioned already, I consider the MonDoc principles in three groups, based on their order on the timeline followed by most cognitive processes: **perception**, **retrieval**, **integration**. The first two are alternate sources (external, respectively internal), of data for the third one, which I consider the process (or set of processes), which gives rise to human thinking. This subsection is structured along the lines of these groups. Before starting, though, let's look at the object of study: documented thoughts.

The implementation of the MonDoc principles is a document editor and engine (see Luchianov, R. (1999)). We use editors to write documents and engines to read documents. But any document is only the means of storage and distribution of what we try to communicate. **Thought** is what we try to convey, to document so others can verify; fortunately for thinkers and unfortunately for speakers, ideas are related to each other in complex **webs** of meanings, relationships, sources, probabilities (to name only a few ways); it takes Art (rhetorics) to successfully communicate even relatively simple thoughts. **Speech** is linear and a very difficult medium for getting across the structure of ideas described above, naturally n-dimensional. So we started making marks (signs) on different surfaces, from cave walls to the silver platter of a compact disk, in order to organize, record and verify what was being said. Meanwhile, Gutemberg introduced the accessible **printed medium**. Books, articles, papers and so on, allow for a more coherent presentation, but because just like speech, they are intrinsically **serial**, the graph-like dynamical aspect of thought is simulated by using hierarchies of sections (book, volume, chapter, paragraph, line, etc), references (both internal and external), footnotes, parentheses, abbreviations, etc. **Specialized software**: help engines, authorware editors, help-desk solutions, research on natural language interpretation and novel human-computer interaction principles (e.g. mnemonic gesture interpreters) are all attempting to find and harness cognitive principles which would make communication through computer-supported media closer to the thinking processes, faster and more accurate. **The Web** is a very dynamic part of the Internet. Until now it is the medium with the greatest potential for efficient conveyance of thought; unfortunately, the majority of existing Web sites overcrowd the visual field and the short-term memory of users and use the helpful features of **hypertext** in a divergent way, prompting the user to skip from one topic to another instead of converging their attention on the ideas presented on specific pages. MonDoc expands on the ideas of hypertext. In a sense, it is a visual DHTML editor implemented as a DHTML document. From a certain point during development, I was able to use it in order to expand itself.

The basic **definition of hypertext** was introduced at the beginning of the 70s as "a method of storing data through a computer program that allows a user to create and link fields of information at will and to retrieve the data non-sequentially." (Random House, (1996)) HTML was an excellent application of markup language combined with simplified references, allowing a web of content, hence the name of the World Wide Web. That being said, there are problems inherent in the implementation of HTML and the way it is being used. Further down I discuss some of these problems and I propose the solutions which I implemented in MonDoc.

And now let's take a look at some the tricks employed by the human mind in order to acquire information from the environment.

Perception

The **information processing** approach to perception is among the theories which state that at several stages in processing sensorial information, neurons simplify the original information by automatic rejection of meaningless or unimportant inputs (McConnell, J.V. & Philipchalk, R. (1992), p.108). For example, in a (moderately :) noisy party you could still understand what a friend says because the brain filters out the unrelated, background noise. Similarly, when concentrating on solving a task you can be interrupted by someone mentioning your name in a nearby conversation because your brain learned to consider your name as a trigger for attention.

On the other side, studies on the **influence of context on the human cognitive processes** has shown (among others, Vlaev, I. (2000), Hadjiilieva, K. (1997), Kokinov, B. (1997)) that especially if the irrelevant stimuli are not consciously attended, they do influence the interpretation of the target stimuli in processes like image recognition, decision-making, problem-solving. Context is the set of stimuli, circumstances or facts that surround a particular target stimulus, situation or event.

HTML and the way it is used currently, has an acute case of **diffuse context**: the way links work, moving focus from one document to another, enforces the natural tendency to avoid boredom; people tend to get sidetracked when using links and often go from one topic to another, making learning slower. Also, some terms have a **different meaning** depending on the situation in which they're used. Often, a consistent description, has to be available in several places throughout the document.

MonDoc suggests writing term definitions, notes and references only once, as in a **glossary** or a **literature reference** section; the system, then places markers after the words for which it has definitions and if the user needs an explanation of the term, it appears on top of the text currently read, not in a different window or in a different page. This presentation allows the processes of learning to use the context for a better anchoring of the description read. Another way to make context work for the benefit of the user is to **reduce window clutter**, by removing irrelevant menus and toolbars, and in return, **added consistent context** (headers and footers, present on every page), necessary to navigation and orientation through the document.

Besides the obvious, everyday use of contours as applied to written text, I used from the top-down, Gestalt school of psychology the principle of continuity, the principle of similarity and I am expanding on the **Gestalt motto**. "The whole is greater than the sum of its parts" More precisely, a perceived whole is something else than the sum of its parts. Gestalt psychologists showed that the final pattern of an object influences the way we perceive its parts, in seemingly recursive ways. This goes for simple (left image) as well as complicated scenes (right image)

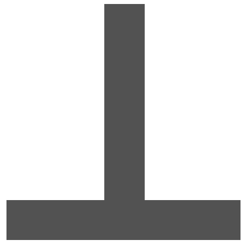


Figure 1: Is this inverted T longer or wider?

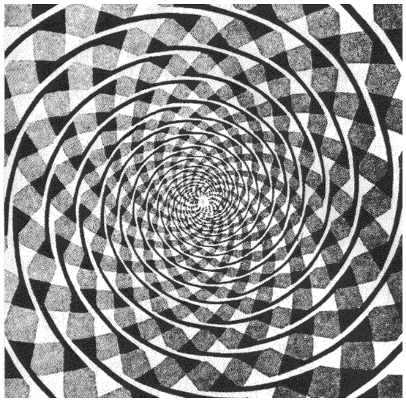


Figure 2: Do you see circles or spirals? Combination of figure-ground effects which leads to automatic misinterpretation of the components

The **principle of continuity** (the third Gestalt principle of perceptual grouping), states that things which are connected through some feature (like time of occurrence, position, color), belong together. (Arnheim, R. (1986)) When people look at separate pieces of a puzzle, they see feature-hints like the thickness of one of the lines below; when they put together those pieces, they see the picture. The **principle of similarity** (the fourth Gestalt principle of perceptual grouping), describes the automatic grouping of stimuli exposed to a perceptive field, based on their similarity. (Kemp, S. (1987))

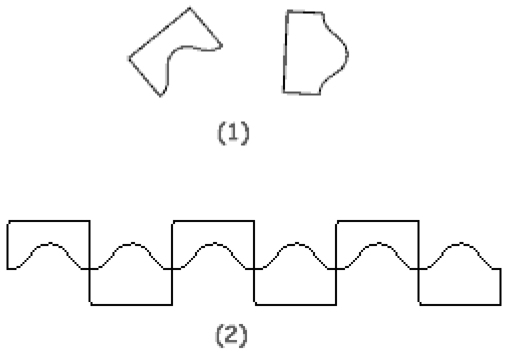


Figure 3: Illustrating the continuity principle

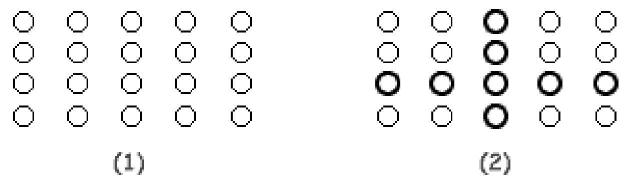


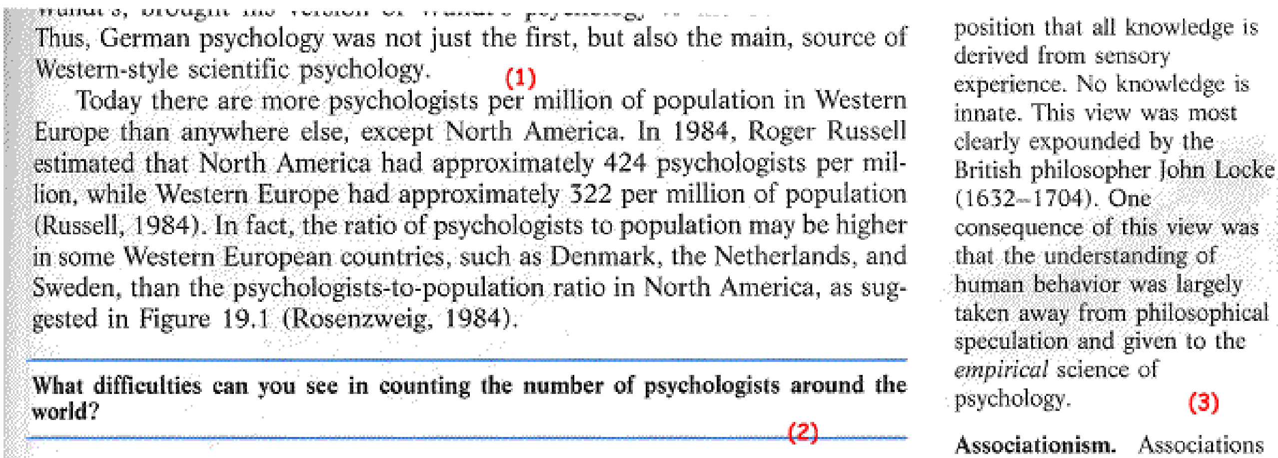
Figure 4: Illustrating the similarity principle

Research argumentation (in articles and even in books), tends to expand quite dramatically, but large bodies of text tend to be hard to read and **monotonous**; they get easily skipped by all except the most steadfast readers; it was also shown that 'chunking' and creating visible structures for information greatly improves learning speed, the quantity and quality of the memorized material.

MoStaCon Solutions - This requires added **emphasis** for keywords in a text and specific mentioning of the features of a window which are always present and tend to 'fade' away from awareness in the long run. When this rule is well applied, the processes described by the Gestalt principle of similarity tend to simplify reading by guiding the reader to read 'between the lines' a **shorter phrase** made out of the highlighted words; later on this initial phrase is **shaped out** and detailed by reading the whole text. The same principle makes **orientation** on the page easier (scanning the text to find a specific description), and helps with the **iconic memorization** of each page. I also use, rather freely, the principle of continuity by **color-coding** the background of the text: depending on the source: glossary gets one color, references another color, which simplifies for the reader internal categorization of the material read. Another solution against monotony is practice with **illustrations**: within the MOnitor and in the quickHelp system of the StaCon module.

Repetition has also a **negative side**. As every student can tell, even the most interesting participant of study can be rendered obnoxious by too much repetition. On an unconscious level, the influence of every stimulus which does not help an immediate goal is reduced by the law of adaptation: "the increase or reduction of sensibility following the repeated action of the same stimulus or due to the modification of the environmental conditions (context)" (Zlate, M. (1999), p.64). Helson, H. (1964) considers that there are three classes of stimuli influencing adaptation: focal stimuli (F) which are directly evaluated by the subject, background stimuli (B) which surround the focal ones and residual stimuli (R), which are supplied by previous experience, not necessary related to the focal stimuli (anywhere on the quasi-temporal axis of personal experience from latency of the sensorial system until automatic retrieval from LTM). Not surprisingly, Helson's law states that the adaptation level $LA=F^{W1}B^{W2}R^{W3}$,where the three Ws are weights determined experimentally. This psychophysics law is formulated in sufficiently general terms to be applied in other domains too – like learning. This is why if we want it to be efficient, repetition has to happen at intervals and if possible, to be done in applied, practical ways.

Being convinced of the validity of the law of exercise, I'm using **repetition** at every step in the MOnitor: the user can read reference material or see descriptions of terms whenever they meet the term in the text. Imagine a good textbook you know. Does it have side-columns, information boxes, graphs, captions with questions related to the material? It's likely that its whole text is structured that way. Information is presented over and again, in several forms and from different perspectives. This redundancy is necessary for faster and long-lasting learning.


Thus, German psychology was not just the first, but also the main, source of Western-style scientific psychology. (1)

Today there are more psychologists per million of population in Western Europe than anywhere else, except North America. In 1984, Roger Russell estimated that North America had approximately 424 psychologists per million, while Western Europe had approximately 322 per million of population (Russell, 1984). In fact, the ratio of psychologists to population may be higher in some Western European countries, such as Denmark, the Netherlands, and Sweden, than the psychologists-to-population ratio in North America, as suggested in Figure 19.1 (Rosenzweig, 1984).

What difficulties can you see in counting the number of psychologists around the world? (2)

position that all knowledge is derived from sensory experience. No knowledge is innate. This view was most clearly expounded by the British philosopher John Locke (1632–1704). One consequence of this view was that the understanding of human behavior was largely taken away from philosophical speculation and given to the *empirical* science of psychology. (3)

Associationism. Associations

Figure 5: Good textbooks provide in the context of the main text (1), captions with questions which apply the material discussed (2) and definitions of key words in side columns (3) (from McConnell, J.V. & Philipchalk, R. (1992))

This solution is possible due to a characteristic common to low and high cognitive processes. On the background of adaptation, and as a result thereof, sensorial systems exhibit (an evolution-induced) sharp **sensibility to contrast**. The sensibility of a sensorial field increases in places where regions of different intensities meet (Zlate, M. (1999), p.66). For example, in the circles from Figure 3 it may seem to you that the white right next to the gray circle is brighter and the gray close to the border is darker (lateral inhibition). This principle seems to hold for every human sensorial field. It allows us to differentiate small amplitude and frequency variations, it improves the precision of our sparse 'grid' of temperature and pressure-sensitive sensors (touch), and it does similar things for the olfactory and gustatory mucous epiderma.

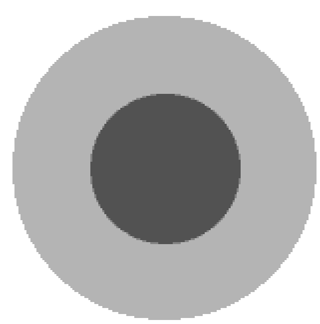


Figure 6: example of reciprocal lateral inhibition

The process is not limited to low-level perceptual processes. A good way of varying repeated presentation is by **providing contrastive examples**. Also, on a higher level, contrast helps defining the limits of concept definitions (compare and contrast). When explaining to a kid what a cat is, you may point that "it's like a dog, but smaller and not as friendly"

Perception may be the **most natural** way to acquire information from the environment. But many times the senses give us distorted information; evolution found ways to allow us to compare what we perceive with what we have perceived before. We do that using retrieval from memory.

Retrieval

The ways we retrieve what we have learned (or experienced), and the success of the process is studied either by considering the process of **learning** or by considering the object of that process, **memory**. I think that we study learning and memory because we are interested in the ways available to increase or control the (volume- and time-) **efficiency** of retrieval. This section describes the **mnemonic** principles of MonDoc, and their implementation in MoStaCon.

One of the mysteries of the mind is how we do **association**. How does a stimulus 'resuscitate' a memory, and how do concepts interact? How can we answer this question if we don't yet know the locus of memory? Of course there are theories on this topic, there always were. Currently it is believed that memories are coded at the synaptic level of the neurons (with factors such as number of connections, their position on the neuron, the type and concentration of neurotransmitters, the rate of creation) and interact through organized 'firing' of related groups of neurons. This still leaves unanswered But not understanding it does not mean we can't use it. There are historical written records that the use of rhyme, acronyms, alliterations, familiar associations, vivid and unusual imagery and creating images in a way that flows from one to the next (in a word, mnemonics) have been used since ancient times to expand the limitations of short-term memory and improve the quality of **recall -** the reconstruction we make from long-term memory.

When they refer to different memory processes, psychologists use two temporal labels: STM and LTM. They call **short-term memory** (STM) the processes which take place in the few seconds between the sensory-data stage and the actual storage of that data as memories we may later recall. Early literature even referred to STM as physical storage (Lindsay, P., Norman, D (1977)). This definition obviously asks for the second one, **long-term memory** (LTM). Inputs important enough to survive in STM are transferred to LTM in a schematic form, with only certain features (acoustic, visual and/or emotional), preserved: the ones meaningful at the time of acquisition. Later, when used, LTM memories are reconstructed as needed for the occasion and are strongly influenced by the stimuli which prompt the recall and their context (Rodgers, J.E. (1982)).

STM ordinarily **cannot store more** than about 7 items simultaneously; Miller, G. A. (1956) published his article on the magic number 7±2, documenting one more limitation of the short-lived processes of the human mind referred to as short-term memory. In order to allow the user to effectively store the cognitive map of the MOnitor, I never include more than 7 items, features or new ideas on the same page. The first illustration of this principle is that I present the process of experimenting separated in seven parts (designing, preparing, running, formatting, interpreting, presenting, archiving).

With documents, this limitation translates into a problem **- reading capacity overflow**; the WWW is known for its explosive growth. When using search engines, one can find thousands upon thousands of pages on a topic, most of them unrelated to that topic or redundant. This prompts the **'surfing' behavior**: the overwhelmed user, with the 7±2 STM capacity long submerged and with the attention shredded by numerous unrelated banners, ads, sidebars and slow-loading tables, is reduced to clicking his or her way through the maze, balancing his or her **original goal** less reliably and for less time than a wave surfer on his or her board. The browsers don't help much, with their ever-increasing collection of menus and toolbars, included originally as welcome navigation aids, but most times left open by users who don't know (or care about) how to turn them off when not in use. This compounded continuous attack on the visual field is forcing users into unconscious **attention withdrawal** and they very often miss the few hints with which the browser and some good search engines try to help them organize. A good example is the fact that almost none (1 out of 8) of the participants in this experiment noticed the fact that the title bar of the MOnitor always contains the name of the current page, thus helping them to find the MOnitor window after they lose it (when it got covered by other windows.)

MonDoc solution - de-clutter. The MOnitor window is stripped bare of clutter like toolbars, menu bars, link bars, search engines etc., allowing the user to concentrate on the task at hand (learning or experiment design) MoStaCon solution - each topic is broken down to pages containing no more than 7 (possibly) new pieces of information

As research goes on, more and more categories of connections appear to take part in storing items in LTM; we realize more and more how **complex** is the memory system.. Goodglass, H. (1980) lists eight different categories: **identity** (the item/idea itself, e.g. "chair"), **class** (one or several hierarchical groups in which the item takes part, e.g. "furniture"), **attributes** (various physical attributes of the item and psychological effects it generates, e.g. size, preference), **context** (location where the item is normally encountered, e.g. "next to a desk"), **function** (verbs with which it is normally employed, e.g. "sit"), **sensory associations**, **clangs** (words which sound like the item e.g. "pear") and **visual patterns** (salient features of any of its representations you have met, e.g. a sketch or the pattern of the letters c-h-a-i-r) and **reproductive information** (the muscle movements needed to say or write the word or to draw a sketch). This suggests that our brain builds incredibly complex webs of ‘anchoring’ concepts for each item it stores.

It's common knowledge even for non-specialists that learning involves **repetition**. Before the middle of this century, rote learning used to be the chief teaching instrument in universities, schools and with private tutors. Even now repetition is at the core of most teaching methods. As early (or as late) as 1883, Ivan Pavlov discovered experimentally that **low-level association** is stronger the more frequently a conditioned stimulus is paired with an unconditioned or previously conditioned one (McConnell, J.V. et.al. (1992), p.265). Pavlov's observations were coded by E.L. Thorndike (1898), following his own research, as the **law of exercise**.

Thorndike's second law of learning, the **law of effect**, made more precise the way S-R (stimulus-response) connections are created, by (1) strenghtening S-R connections by offering desired stimuli (satisfiers) while or immediately after the presentation of the stimulus and the appearance of the response and (2) diminishing S-R connections by offering feared or repellent stimuli (punishment) at that moment. But the learning I am interested in is not the respondent conditioning described above, which is more physiological than psychologic; still, since part of what my interface is supposed to do is to teach the beginners and experts alike to use the MoStaCon system, I'll describe in a couple of words B.F. Skinner's technique of **behavioral shaping**, not using his bowling pidgeon example (McConnell, J.V. et.al. (1992), pp.273-275) but the operant techniques I built into MoStaCon's interface. The terminal response (goal) of the

MOnitor users is to streamline the process of designing, preparing and running experiments; that is, to achieve the final experimental results, do it faster, with fewer errors and with a better understanding of those results and of the steps taken to achieve them. Software in general can't afford to use punishment, so according to Skinner it has only **positive and negative reinforcement** to shape the behavior of users and increasing their learning slope. The MOnitor provides positive reinforcement by offering quick access to terminology, by giving solutions to everyday problems and in many other ways discussed in the following chapter. In addition, every time the user does not supply all the necessary information for making a decision at a certain point, the MOnitor points that out in **descriptive alerts** and does not advance until the necessary decision is emitted (negative reinforcement). The activities which are reinforced tend to repeat, while those which are not - or are negatively reinforced - tend to drop out of the behavioral repertory.

One of the goals of the MOnitor is to teach how to design and run experiments; Actually, what the MOnitor does is to present in a structured way (**a page graph**), any information that the user stores within it. From this point of view, the MOnitor is a database-oriented 'slide projector'. In order to help fix that knowledge in the memories of the users, MoStaCon also has a section with applications: pages which take the users through the **complete procedure** of designing show-case experiments, thus providing them easier retrieval from episodic memory of the relations within MOnitor's decision-making graph (process I call mnemonic fixation).

Many authors write good books by **addressing the audience** and creating a holistic view of the subject matter (McConnell, J.V. & Philipchalk, R. (1992), p.xi). We have also seen (above, about complexity) that human reason is based on a highly redundant **relational web of symbols**. With MonDoc I'm addressing exactly those features of the mind. When a reader goes through a text, s/he unconsciously (automatically) looks for clues to help place that text in a context. This is the reason for which (good) printed materials put **key information** (author, title, page number, chapter, etc.) in headers and footers. This is also why authors write definitions in parentheses next to key words and why writing standards require references next to ideas taken from other sources. HCI literature on ANS (adaptive navigation support, see Brusilovsky, P., Eklund, J. (1999)) approaches this problem through the use of **adaptive presentation**. In this approach, pages of a document are composed for each user from pieces, depending on the level of expertise of the current user and depending on the path used by the user to reach that specific page. This involves more intense processing in the system and a user database which keeps track of what each user has seen already from the document (or what other documents the user has seen). Experience (Brusilovsky, P., Eklund, J. (1999), §5.2) showed that the user model built by such a system can't accurately model random users, mainly because the user model doesn't have any way to check what the users read on their own.

I took a simpler approach, **allowing the users to decide** when they need more information. I decomposed topic redundancy so that the writer will have to mention a reference only once, after which the system takes over and places that reference in all necessary places, thus generating **redundant hypertext**. All the users need do is to click on **note markers**. In the prototype of MoStaCon I decided to present note markers as blue question marks placed after the words or phrases which they need to explain. The types of references which MonDoc is handling now are **glossary entries**, **replacement bits** and **literature references**.

Replacement bits are pieces of text which the writer uses often; things like long institution names, full names for complicated procedures. **Glossary entries** are special definitions for specific words or phrases, expansions of acronyms or footnotes (full-fledged notes which can in turn contain references). If this strikes you as **recursive**, possibly infinitely so, remember that this is also one of the positive features of thought. **Literature references** are defined once in full, the way one does in the references section of an article or book, and then for every mentioning of the reference, only variable information (like page numbers or chapters) have to be added; the system fills the rest of the reference text (publication, publishing house, city, ISBN, etc.) and presents it in a pop-up, differently colored than the other notes.

Replacement bits are **directly expanded** in the text. Unlike them, glossary entries and literature references appear as **question marks** next to the word they explain; when the reader wants an explanation of a word or a reference, s/he can point to the question mark and a pop-up tip appears for a few seconds. If the time the system allocates is not enough for reading the whole reference text, the reader can click the question mark to get the reference floating over the text until it is removed by a second mouse click.

This system **reduces** the amount of **time** spent by the reader of printed matter who would normally go back and forth in the text looking for references, endnotes, and glossary entries. My implementation of redundant hypertext is also 'backwards compatible' with readers who don't like computers or don't want to read long texts on a monitor screen. Writers can use MonDoc to compose a document, then **run a converter** to get it from the MonDoc DHTML format to the linear, bulkier, heavier, RTF format which can be printed out from any document editor (MSWord, WordPerfect, etc.) . The type of **traversal** I use in order to unfold a redundant hypertext document, is a simple depth-first one, which also looks for special markers that a writer can use to specify a different order of traversal. This is provided in case a writer decides to present in the virtual version the text in one order, but in the classical, linear version in another order. Anyway, notes and glossary entries are expanded only the first time they are encountered; making a proper connection between the paragraphs of the resulting linear document may need some manual work and ingenious writing. I am also planning the reverse filter, a tool which will simplify converting text files and other formats into my MonDoc format.

Another idea which I try to incorporate springs from the early work of B.B. Murdock on learning interference (McConnell, J.V. & Philipchalk, R. (1992), p.302), namely the **serial position effect**. By giving participants lists of words to memorize, Murdock discovered that they tend to remember better items from the beginning and the end of the list. From studying the recall percent curves and with additional experimenting, he inferred that items at either end of the list interfere with the ability to recall. Items presented before a specific item effect it retroactively, while items presented after that item effect it proactively. To apply this, in writing MOnitor pages I try to present more important points at the beginning and at the end of the page.

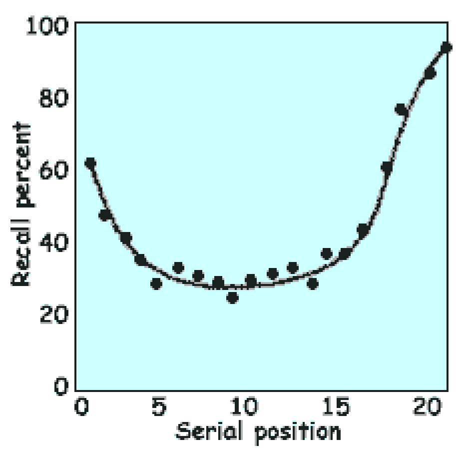


Figure 7: early items are 'victims' of retroactive interference, latter items 'suffer' from proactive interference (milder), while middle items are affected by both processes (Murdock, B.B. (1962))

language. A current (neurologic) view on the topic is that those abilities are lost due to a breach (or breaches) in one of the three stages of the process of naming: (1) retrieval or recognition, (2) search for an auditory representation, (3) search and execution of the respective motor commands (Goodglass, H. 1980.) An example of stage one aphasia present even in people with normal memories is the **tip-of-the-tongue** aphasia. (Schacter & Worling, (1985)) The affected person can't utter a specific word, though the shape and sound of the word, as well as the meaning and synonyms are perfectly accessible. MoStaCon 'treats' aphasia the way it deals with other retrieval problems. It uses priming and **forced choice** to allow the user to pick the next step in the learning space or in problem space, by clicking buttons or choosing items from menus.

Priming is the exposure to specific stimuli shortly before presenting them in an experiment. Such exposure is also observed to speed up the process of retrieval from memory of the primed stimuli and other stimuli which the participant connects with it.(Schacter &Graf (1986), Kokinov, B. (1997))

I will end the topic of retrieval by mentioning the basics of J.R. Anderson's **Adaptive Control of Thought models**. Based on an impressive amount of research in psychology, physiology and the social sciences, Anderson proposed in 1976 (McConnell, J.V. & Philipchalk, R. (1992), p.306-308) a model of memory which is in wide use and being developed by an international body of researchers and was revised twice: ACT* in 1983 and ACT-R in 1993. In my usability study I am not modeling the users in too much detail, so I won't enter the details of STM (referred to in ACT as working memory), or the interesting modeling of retrieval as spreading activation. To put it in an extremely schematic way, the ACT model assumes the existence in humans of separately processed LTM 'banks' for procedural memory and for two types of declarative memory: episodic and semantic.

Procedural memory stores actual physical chains of actions necessary for accomplishing tasks (like the movements necessary to move the mouse cursor on a computer screen which are different if you use a mouse than if you use a trackball or another pointing device; the MOnitor does not go into such detail, leaving it for another MonDoc document I plan, the iMMedia).

Declarative memory (contains factual knowledge, describable in words); **episodic knowledge** the more easily retrievable type of declarative knowledge records chains of things one has done or imagined doing), while **semantic knowledge** is made out of abstract, definition-like 'chunks', with no reference to where and how it was learned.

Procedural knowledge may be gained from a virtual medium through **observational learning.** For example, in order to teach MoStaCon users how to perform practical tasks (using a window, navigating using the pop-down menu, etc), at the places in the tutorial where these tasks are described (declarative knowledge), I added animations which show every step of the task, by example.

Integration

The integrative processes which happen in the human brain, as low as stimulus recognition and as high as inference, generalization or analogy, are the building blocks of our successful survival as a highly adaptive species. I'll discuss low-level integration first.

Many philosophers argue convincingly that **adaptation** is the major trait of life (rats and cockroaches are highly adaptive species too), with **intelligent life** as its epitome. So we should teach our tools to be adaptive too. A piece of software which adapts to the style of work of the user has better chances to 'survive' in the virtual world of software utilities. I don't mean malicious adaptation, of the kind implemented in many intelligent viruses, but useful adaptation, as described in Eklund, J. et. al. (1997) Namely **adaptive navigation** and **adaptive presentation** for tutorials, primers, other kinds of electronic textbooks, and even user interfaces.

Some principles of **software engineering** are more specific versions or combinations of the perceptive and mnemonic processes discussed until now. Since software is trying to speed up or even model the natural processes of the human mind, its guiding principles should be based on integration of input and goal-directed processes (algorithms); the goals are, most of the time, some sort of useful output from the system.

Any programming language (and in structuring MoStaCon I use three – perl, JavaScript, HTML) has **rich syntax and an extensive object hierarchy**, which challenges even the memory of experienced programmers. StaCon's solution are intuitive, visual lists of modules with the documentation present in the same context: quickHelp windows appear on top of the filter in progress, and if the few hints from quickHelp are not enough, each term and module has more detailed description within the MOnitor – one click away.

As I mentioned already, MoStaCon offers ready-to use **pre-programmed modules** available in pop-up menus with mnemonic descriptions and modules which facilitate writing new modules. MoStaCon is friendly to the programmer too, and I tried to make the process of updating it - adding new modules or modifying existing ones **-** as understandable as possible. To begin with, the **modular structure** of the library helps a lot. One may develop a module by using the existing ones or totally independent of them. That module will still use the StaCon interface, unless, of course, the programmer doesn't decide to use the library directly, by writing the filter script manually, with all the syntax and typing errors this involves.

As noted in the introduction, the choice of the programming languages was not random. There are `perl` fans who would program everything they do in their favorite language (and might do it more efficiently than I did on StaCon); my choice, though, was based on several objective facts:

The **'bastard effect'**: perl was put together by 'borrowing' syntax from other existing languages; a programmer who knows C or Pascal or Fortran or any other programming language can understand perl and will find many familiar elements in its syntax;

Freedom of the syntax: with its many structures, perl allows the same task to be achieved in many ways, some of which read just like spoken English;

Ease of development: being interpreted when it is started, the script doesn't need to be compiled, which saves considerable amounts of time while testing additions or modifications to the filters.

About the **document format**. When we use a book or an article, we think about it as a single object. We use our hands to peruse the book, to turn pages, etc. So we have an object and we employ a learned mechanism in order to use that object. MonDoc documents are also composed of these two parts. The document itself is a file on disk (with the extension .js, because it is an active document), and the mechanism which we employ is the so-called 'engine' (with the extension .htm or .html, because it is a DHTML document.) However, MonDoc architecture allows the author of the document to incorporate the document and the engine in one file, thus generating a single file which can be used as an object in itself. With MoStaCon I used the latter approach. It is only one file, almost like the Help files from Microsoft. The only exception is that the images used in the document, following HTML standards, are still outside the document, in a folder, allowing the author and the user to treat them individually (within the folder), or together (by handling the whole folder).

Bound to impress MonDoc users (readers and writers alike), is the **size** of the MonDoc documents. I calculated that with the word-based redundancy compression I am building in the MonDoc engine, the text of the resulting documents (the HTML file) will be under 10% the size of the Web area it generates, For comparison, Info-Zip compression, the one used in WinZip and other professional archive tools, reduces Web-area (D)HTML documents to an average of 48%. So if someone wants to compress a MonDoc document even further, using WinZip, they would get a final compression level of under 5%. If we take into consideration disk space lost as slack for every HTML file in a normal Web area, the space taken on disk by a zipped MonDoc document will be under 2% of the space taken by the corresponding traditional Web area.

1.2.2. Project Management principles (MoStaCon implementation)

Project Management techniques and rules-of-the thumb used in organizations (or in teams or even by individuals) who want to plan their projects in order to know in advance how long it will take and how much it will cost. Management Information Systems (**MIS**) are programs or procedures used by managers in order to keep track of the information they need on a specific project. In my understanding **a project** is any (research) task which is formalized, planned and documented, usually requiring the concerted effort of a whole team of diverse specialists. The periods of time which should mark the be finalization of specific phases of a project (**deadlines**) are more and more sensitive in a research community increasingly dependent on commercial applications for funding and research grants.

Keeping track of information

What information does a researcher track? The constraints of empirical research (detailed in the primer), the design of the current experiment, subject data, ideas and suppositions, references and relevant literature are only few of the large amounts which a researcher is supposed to organize and use as material for further research.

The main body of MoStaCon, the MOnitor is in fact a **primer**. But the core module of the system, StaCon, the one which I developed first, may wind up saving researchers more time by allowing them to write automatic data filter/analyzers.

One may look at MoStaCon as **a low-level MIS**. As such, it identifies the problem areas of the process it addresses (experiment design) and offers the manager (the scientist coordinating the experiment) ways in which to track the advance of the project (the experiment log), ways to automate repetitive tasks (the DM graph, the StaCon filters), ways to check the accuracy of used data (the filter report), tutorial to speed up the training of new personnel, archival systems (built into the MOnitor) and a global interface (the Console, the MOnitor, the StaCon interface).

Primers are sparse tutorials used to remind their readers of the specific meanings or usage of specific terms or procedures (see priming, above.) For the MOnitor, I used **class material** from the course in Experiment Design at the New Bulgarian University, I adapted some of the **textbook** we had for that course (Elmes, D.G. et. al. (1992)), as well as **advice** from Elena Andonova and **feedback** from the participants in the survey presented in Chapter 6. An unfolding of the text of the MOnitor prototype is available in Appendix 1.

The excellent **short article** by Hendricks, B. et.al. (1990) does in my opinion, a good job of summarizing the diversity of psychological research in a relatively simple, four-dimensional structure. The dimensions described in this article (see the two pictures below) are: number of subjects to be used (one vs. groups), data collection technique (subjective vs. objective), control (manipulative vs. observational) and setting (laboratory vs. ecological)

Given this large amount of tasks and the multitude of existing paradigms in existence, it is no wonder that beginners or busy experimenters **forget to include** control factors in the design or that they skip entire seemingly non-essential steps (like getting descriptive statistics on the data before deciding on the method of analysis.)

Planning

When we 'meta-think' (think about thinking), we tend to evaluate the process as a **rationally planned** one, guided by our consciousness toward a desired goal. Just one of the few BIG Aristotelian oversimplifications – or maybe simply a problem of translation. By the middle of the 17^{th} century, Hobbes described in his Leviathan the other type of thinking, older from an evolutionary point of view. Thinking in an **unplanned, undirected way** leads scientists (and not only them) to many powerful 'gestaltist' insights. Nevertheless, early psychology considered fantasy, reverie, as a loss of time or even a pathologic case. Recent research, after the 'anthropologic revolution' of the 60s describes both types of thinking. Following the opinion of Scarr, S. & Zanden, J.V. (1987), **directed thinking** is systematic and logic, deliberate and intentional, guided by purpose; it is used for solving problems, formulating laws and reaching goals, by using symbols, concepts and rules. On the other side, **undirected thinking** is characterized by the spontaneous, free interaction of seldom formulated thoughts (the tough subject of mental images), without the guidance of a plan. This is the engine of imagination, fantasy, reverie, and people use it as much for its productive, creative results, as they do for relaxation and freeing themselves up from their daily constraints. Research involves both of these kinds of thinking. The first and last phases of empirical research, (finding and operationalizing research questions, as well as interpreting results and writing interesting presentations) make plentiful use of undirected thinking, while the technical phases (preparing a design, an apparatus, collecting data, analyzing it and archiving) need applied, directed thinking.

MoStaCon can help mainly with directed thinking. It stores already made decisions in a file called **'experiment log'**. It may display these decisions (working hypotheses, variables used and analysis design, etc.), organized and summarized in a window called **'summary pane'** (planned feature). The StaCon module generates customizable **analysis reports** (the user can decide to include only a summary, or a complete description of every operation done on the data), projected archive catalog)

Decision-making (like the graph) is another interesting case of recursive definition; it is a special case of problem-solving because it involves defining the situation, adaptive answers to it. At the same time, decision is involved at every step in problem-solving: what method to use, which sub-problem to analyze first, when to stop following one path and switch to another, etc. In this paper I will look at this field of psychological study in a highly abstract way, by discussing only one of the models used in the field, the decision graph.

I mentioned already that the MOnitor is a decision-making graph. As such, it helps beginners to build a **cognitive map** of the processes involved in experimenting, and it helps beginners and experts alike to be consistent and avoid skipping steps. For the term of cognitive maps we have to thank E.C. Tolman and his early experiments with rats in mazes (McConnell, J.V. & Philipchalk, R. (1992), p.279).

Artificial intelligence has many ways to deal with mazes, but most of them involve the reduction of the maze (the problem to be solved) into a mathematical model called problem space. In order to understand problem spaces, let's examine its representation, the graph.

Graphs happen to be such basic concepts that I shudder every time I think about the self-references involved in any attempt at definition. Everyone reading this paper should already know which of the definitions of the word graph I mean: a structure of distinct entities called nodes which may stand for specific states of a physical or abstract system, or for positions in an n-dimensional space (with n>0), or for pages in a book or for whatever 'things' in the most general sense you can imagine (and want to study). The nodes are connected among themselves by entities called links, which stand for the operations necessary to 'move' from one node to another (e.g. turn a page in a book). Links can be bi-directional (e.g. if you turn a page, you can turn it back to get to the previous page), or unidirectional (meaning that they can be traversed only from one node to the other and not backwards; e.g. if you shred a page you can't 'unshred' it in any one operation I know of). When the graph has several operations associated with it, links can also be labeled (with the names of those operations). In order to be part of the graph, each node has to be linked with at least one other node; the upper limit is scary, though: one node can be linked with each other in the graph (including to itself) by as many links as operations the graph has. If you pick a node (A) and follow links to another node (B), listing the names of the nodes and the labels of the links (if any), upon reaching B the list you made is called 'the path from node A to node B' (by the way, a list is a special case of graph too.) Another special case of graphs is the tree: a graph in which each node can have only one path to special node (the same for every node in the graph) called root. Nodes which are linked to only one other node are called leaves (they are the farthest possible from the root on their path)

Traversing every node of a graph by moving through links from one node to another may seem a trivial task to anyone who didn't try to model it. After all, even a mouse can do it. Skipping the fact that a mice were chosen as… lab-rats exactly for their strong survival instincts (drat! can't find the reference for this one!), a maze is a pretty straightforward type of graph. Literature gives many examples of graph traversal, from **breadth-first** (for every node all the links are checked and only after that the links of the 'children' nodes) and **depth-first** (for every node the link of the first 'child' node is traversed to a leaf, and only then are the next links checked) to heuristic methods of **pruning** (discarding improbable sub-problems) and **ordered search** (using specific evaluation function to decide which link to follow at any moment)

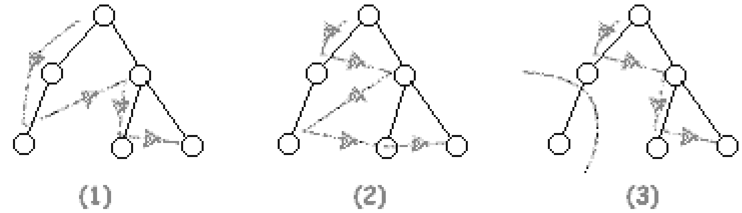


Figure 8: traversal: (1) breadth-first, (2) depth-first, and (3) pruning

Now why did I get through all the tedium of redefining the wheel? In order to highlight some features of the mathematical model which tend to be swiped under the carpet. When we try to solve problems, we recursively split it into smaller sub-problems until we find obvious answers – or contradictions. The collection of all sub-problems of various sizes is the original **problem's space**. Such spaces are represented as graphs. When all the immediate sub-problems of the original problem are solved, the original problem is solved. Technically put, achieving a desired goal can be equated with finding an appropriate finite sequence of application of available operators (Barr, A. & Feigenbaum, E.A. (1981)). This may seem at the first glance a relatively simple recursive definition. This is what researchers originally thought – and were surprised. In the previous example with the mouse in the maze, there are no steps the mouse can make which change what happens somewhere else in the maze. But this happens frequently in problem space: solutions to specific sub-problems may modify the already traversed graph, or even (rarely, but it happens) add new operations to the graph. I didn't find any traversal algorithm which takes that into consideration. A partial solution used, among other places, in Anderson's ACT models, is **spreading activation**.

Research has shown that we perform much better in **forced choice** tasks than in recall tasks. The tip-of-the-tongue aphasia (or tip-of-the-finger, while using a keyboard :) illustrates the fact that choice is much faster than production. A good reason explaining this fact is that during choice, the time-consuming, heuristic process of retrieval from memory is replaced by the faster, evolution-enhanced process of direct perception. This is why the psychologists at the Palo Alto Research Center developed long ago the 'menu-and-dialog-box solution' of the generic Graphical User Interfaces. At the time, the current research on Human-Computer Interfaces, mainly done on experts, was adamant in its adherence to conversational, command-prompt interfaces. It made sense, since the experts were those who mainly used computers. The GUI solution was included, with enormous success in the interface of Apple computers, among others and later on in Windows. I adopted and expanded upon that solution so that most things which require predictable response from the user are presented in pop-down lists, check boxes or radio buttons, but with the context preserved (in plain text, pop-up hints and descriptions or image schematics placed around the dialog box).

Planning in MoStaCon is achieved through prior listing of all types of notes to be made on each experiment in the **notes menu**; there is also a planned **to-do list**, with which the coordinator of the project can delegate tasks to the members of the team (and set deadlines). Finally, if while going through one experiment, the researcher gets ideas which may be expanded in a full-fledged project, s/he can create **empty experiment logs** (with only the name and the ideas specified), which will act in the **catalog of experiments** as a reminder of future plans.

I will quote Peale, V. (1994): "When you have what you need for discovering the creative aspects of things versus the proof, sometimes embarrassing, of human existence, when you continue to believe in a happy ending, you are a really positive thinker." **Positive thinking** is characterized by rationality, an actively constructive orientation towards the problem, helps to easily overcome difficulties, while **negative thinking** is visible in the passive, distrustful, uncommitted attitude. The pre-made answer "it's impossible" or "it's difficult," to questions which require the application of cognitive processes available in any human brain may be a mechanism of defense against the psychological shock of failure. Both types of thinking have benefic effects on the behavior of the individual. Positive thinking affords a better adaptation to environmental stress and boosts self-respect; negative thinking increases the chances of repeated success of the tackled tasks. For example, I tend to think positively about my own capacity, so as to increase my productivity; the danger there is that I tend to over-appreciate it and to miss deadlines. It's a philosophical point (generalized as freedom over security), whether I choose increased productivity or I do the conventional thing so that people don't consider that I'm "living in the clouds of my own fantasy."

One type of interface, which may allow positive thinking and still keep to a good schedule is used every day in project management: the **time-table**. Lists of tasks which can be sorted by several characteristics and which serve as reminders to the people from the same team working on a project – or series of projects. The research coordinator may use such an interface (planned) within MoStaCon, so that at the same time with deciding what the steps in the design of a specific experiment are, she could specify target deadlines and assign research assistants or lab assistants to specific tasks.

Let's look at a **specific task**, which exists in every project and is silently hated by researchers from the humanities: data formatting and statistical analysis. I try to keep the readers/users in a positive frame of mind when presenting the intricacies of experiment design and especially when I face them with the task of writing their own data filter/analyzer. Statistical packages are nowadays excellent in their computing power, but they fall short on several grounds, as discussed below

Providing effective help to the beginner. Besides the instructions given by the MOnitor, StaCon offers suggestions and gives directions through its own internal Help system. I named it quickHelp because it is one click away from each term used in the interface (to help enforce the connection of the semantic description to the procedural knowledge necessary in using the interface) and because I try to keep the sentences short (easy to read.)

Transparency to the inner works of the statistical method. All good statistical packages have excellent documentation, but very poor ways of relating the provided help to any specific results; the displayed numbers seem to appear by some magical, computer-entrusted process. So beginners -- and sometimes even experts -- when looking at some displayed results can't tell which specific version of which function was used to generate them. To prevent this, I built into the StaCon filters two types of transparency: quickHelp in the interface and a reporting system which (if the user requests a verbose report) can enlist absolutely every step in the conversion of the data from the provided input to the generated output. This system also allows for easy verification of intermediary results.

Flexibility in input. Since most statistical packages are using either a list paradigm or a spreadsheet one, raw data has to be formatted in specific ways by hand or with previous analyses before entering it in most analyses.

Flexibility in output. Similarly, the output is fixed by the program and it takes learning a new (programming) language built in the package (like Statistica BASIC) if the user wants to get only specific information formatted differently in the output. .

The best comparison I can find for StaCon to an existing statistical package is with Excel's AnalysisPack. A limited but sufficient set of statistics, with an accessible interface which makes easier the job of entering data and understanding the analysis itself. If it was not so darn time and memory-consuming to use AnalysisPak remotely, from a web application, I wouldn't have gone through the pain of creating a 'yet-another' statistical package in `perl`. But by doing so I gain not only speed and reduce memory constraints, but I get the flexibility incumbent in relatively easy updates.

The next MoStaCon principle deals with how to handle all the data mentioned until now.

Storage

Relational DataBase Management Systems (usually referred to as RDBMS), are programs which specialize in storage and retrieval of data; they are relational because they allow the database architect to set specific data to automatically perform operations on other data (searches, consistency checks, etc.) In developing MonDoc, I kept only few of the RDBMS concepts from its predecessor, HYMNS. What does a document need which can be abstracted as consistent data series? Glossary entries, pages, references, replacement bits, layout elements. MonDoc achieves the high levels of modularity and compression exactly by separating the redundant parts listed above into separate series and relating them to one another using automated, data-oriented routines. From that point of view, it does exactly what a RDBMS does: separates the tedium of dealing with text consistency and formatting, and allowing the user to concentrate on the structure and content of the document.

Data is information in a latent form, preserved in documents of any kinds. Data can be stored in archives, banks, libraries, on different media, each with limited storage life. In all os these storage media, data is classified, categorized, fragmented and often compressed. Collections of data about similar things are collected in tables; individual table rows contain data about the same thing and are called records; record information is split in fields; all the fields with the same name (or ID) from every record make up a column; in the each table, each column contains the same kind of information (text, numbers, codes, time, etc)

HTML documents are **distributed**. They may stretch over entire directory structures, or even parts of them may exist at different locations. Sometimes, it's difficult to say which file one has to double-click in order to start a document. Also, it's easy to delete parts of a distributed document without noticing the fact until we have to deal exactly with that part. With large documents, hypertext runs into a problem with **download times**; at high network speeds, this is not a problem, but at low speeds, the reader has to interrupt the flow of reading in order to wait for the next page to load. This fact prompted the apparition in the online community of a new meaning for the acronym WWW: the World-Wide Wait. Joke aside, the interruption of reading any text is reducing the attention of the reader (consciously reported by readers as "bothering"), and may generate loss of context as the reader attends to other activities.

MonDoc has one solution for these two problems **- encapsulation**. All MonDoc documents maintain the portability of DHTML and one may design a document so that different parts are at different locations, but they include whole documentation graphs in compressed form, in one single file. This makes the navigation less diffuse and eliminates the waiting times between pages. For the programmers, MonDoc architecture is reducing development times and the stress associated with the need to keep navigation consistent.

The 'storage' model of human memory and the need for **dedicated storage and retrieval** of data correlated with the development of DBMS. Later on, the need to make that data storage medium behave more like the way the mind works, coupled with the evolution of the models of structure-based retrieval from memory (Rumelhart, Gentner), correlated with the development of RDBMS. I'm saying 'correlated' because both advanced might have been caused by the same 'ripening' of the ideas in different areas of the scientific community, as well as by mutual influence.

Based on a special formal language with a syntax very close to Basic English (**Structured Query Language**), these 'bases of data' allow a very readable description of data in terms of objects with properties and methods. The system **abstracts** away from the user the details of implementation, which can be perfected as technology offers new solutions. Theoretically, an SQL-described database can be ported on any new system (driven by SQL or a superset thereof), with a minimum amount of work.

I do not use SQL, though, because it places **too much strain** on the web server. Rather, with MonDoc, the document is assembled and rendered on the client computer. A MonDoc document consists of a series of lists of text-chunks and templates which are used to put the text chunks together in the way selected by the user.

I have to mention another bit of tedium: **archiving**; the last step in any sort of work with the digital medium, and often neglected once some project is ready and the paper is written. Episodic memory is very unreliable when it comes to "where did I put that file" or "which were the stimuli used in that experiment". MoStaCon has planned features which will allow the lab manager or supervisor to keep track of the experiments developed with MoStaCon through the mediation of an experiment catalog.

The next section deals with theories of active perception, models of users and implementations which may have elements in common with MoStaCon.

1.2.3. Human-Computer Interfaces

From even before the beginning of modern computing, visionary project managers and innovators used psychological and psychophysical research to invent devices and make them **more easily handled** by human anatomy and better understood by experts or the average person. I will enumerate only the abacus (simplified way to use stones for (ac)counting), and the azerty/qwerty keyboard (structured to speed up typing the words of the English language with both hands). The first 'computer revolution' during the 50s and early 60s was dominated by the uses of psychological research; however, the high R&D prices involved and a new generations of 'canned' computer scientists, more interested in algorithms than in user interaction, coupled with a strong tendency of commercial firms for copyright protection and branding, pulled the mainstream of computer software away from the principles of HCI. Meanwhile, in universities and research labs the research continues and the field is constantly gathering new applicability.

Often defined as **'one-part psychology and one part engineering'**, HCI is a booming new field of study, with University departments all over the world hastily offering courses, concentrations and even degrees in it. In this section I will briefly present some of the theory and terminology of interest for the type of interface which MonDoc creates for MoStaCon.

The current generation of technology addressed by HCI is an interface made out of windows, icons, menus and pointers, on short **WIMP**. The emphasis of operating such interface(s) is on the object which is manipulated. Here are the principles of direct object manipulation expressed in 1983 by Ben Schneiderman (and adopted by Apple Computers and –at least in theory – by Microsoft): the object of interest to the user should be continuously **visible** in the form of a graphical representation on screen; operations on this object should involve physical **actions** as opposed to commands with difficult syntax; these actions should be **fast**, offer **incremental** changes and be **reversible**; the **effect** of the action should be immediately visible to provide feedback to the user; there should be a **modest set** of actions, easily accessible to novices, but it should be **possible to expand** these, gaining access to more functions as the user gains expertise.

When first implemented in HTML, **links** were reduced symbolically to **underlined text**. It was a minimalist representation. As a result, everyone who learns about the Web has to pass the hurdle of interpreting underlined text of various types (color may vary, even the underlining may not be always there) as an object which allows performing an action equivalent to 'turning the pages' in a book.

MonDoc suggests using **buttons**, which are much more associated with actions than is underlined text. As the usability study showed, the users had trouble clicking on links (the Console, Start the Monitor, 'call it up' and StaCon ones), before seeing the interface, but buttons were no problem and even made participants click on them before finishing reading the page.

Mental models

The basic claim of any mental model is that if one knows the **beliefs** of the users about some system, one can **predict their behavior** with that system. The user imagines the result of any action (**mental simulation**) before taking any action on the system. Having such a model, users can plan future actions by using inference on the model or by using analogy with similar devices when the current model is incomplete.

The average user considers the computer as a **unitary tool**. As an example, when I was presenting the introductory lecture on Multimedia to the members of the teams who were supposed to work on web projects and stand-alone multimedia at the American University in Bulgaria, everybody, including students with average experience with computers laughed when I started the lecture by showing them what a computer is. By the end of the lecture, though, they were amazed at the complexity, both structural and functional, of this tool which they were taking for granted. The average user is not interested in what's in the black box, though. They want to achieve a task and expect the instrument they use to do more than it can. Typically, when starting to work with a software tool, almost nobody reads its documentation. The **active user** jumps right into the task s/he has in mind and expects the interface to guide the work.

That was my piece of introspection. Now let's survey (some) of the theories and models used to predict user behavior, thus help evaluate user interfaces.

The location and size of interface elements can be modeled if we take in consideration **Fitts' law**: that the time it takes to point (with a mouse cursor) at a given location is shown to be $T=K \log_2(A/W + 1)$, where W is the distance between the current place of the cursor and the element and A is the distance from the geometric center of the element and the point on its contour closest to the cursor. (Blackwell, A. (1999)) What this psychophysical law suggests, is that it's a good idea to place interface elements as close to one another as possible and to make the most used ones larger.

The Model Human Processor is a very simple model of user performance based on psychophysical data, proposed by Card, Moran & Newell (1980). It predicts the speed with which users may carry out specific tasks on a computer by breaking down any action in **perceptual, motion and cognitive events**, then adding up the individual results. in short, $T=n_pt_p+n_ct_c+n_mt_m$, where the n`s are the number of events of every type necessary to solve the task and the t`s are as follows: t_p belongs to [50,200]ms, typically 100, t_c belongs to [25,170]ms, typically 70 and t_m belongs to [30,100]ms, typically 70. Later on, the same team refined this model by adding other possible operations, as the **Keystroke Level Model** and then the **Goals, Operators, Methods and Selection** model which predicted almost perfectly the work of GPS – Newell's famous General Problem Solver model of human problem solving.

Nielsen, J. (1994) further developed Schneiderman's object manipulation principles into a larger spectrum of **development heuristics** to be used for interface evaluation. Here are some of them: **visibility** of system status, **match** between the terminology used in the system and targeted users, user **control** and freedom, **consistency**/standardization, error prevention and **informative** error messages, graphical **salience** of design elements, **flexibility** and efficiency of use, aesthetically **minimalist** design, help and **documentation**.

Kaptelinin, V. and Bonnie, A. (1997), described another set of principles which they call **Activity Theory** (AT), more concise and modeling a wider variety of users on the common tasks of exploiting an interface. The principles are integrated system, each associated with various aspects of the whole activity. They are: the adaptive **hierarchical structure** of activity (activity/goal/action/operation/object/direction), **object-oriented**, **internalization/externalization** (mental simulations vs. actual actions), **mediation** (tools act on objects) and **development** (study of the ways users adapt to the actions and how they adapt the actions to their own experience.)

Measures

I can hear a weathered researcher reading through all my theoretical presentation: "OK, but where are the hard numbers?" Current HCI research describes four major design methods for assessing interface usability: **prototyping, controlled studies, questionnaires and think-aloud protocols,** and **cognitive walkthrough**.

Prototyping comes from software engineering; it involves running clients or prospective users through (1) **rapid** prototyping: successive low-fidelity approximations of the interface, built out of colored paper and more recently with multimedia tools or (2) **deep** prototyping: developing a complete aspect of the system which would allow the interface to be tested. These methods allow developing interface alternatives. In order to evaluate which of the alternatives is better, empirical methods are used, as follows.

Controlled studies are experiments based on measurements of two variables: amount of time to finish a task (RT) and number of errors (NE) made while doing it. This is again based on the assumptions made in the Model Human Processor (good interfaces are faster and less error-prone). Two or several versions of the interface are provided to participants chosen from the population of prospective users, measurements are made and if the means of the distribution for RTs of participants using one of the interfaces is significantly lower than all other ones, that is the faster interface; if the same holds for the NE, we may have a winner. In proper research style, HCI literature insists on statistical coverage as complete as possible of the sources of variation for the measurements as factors in the analysis: variation in the task to be performed and the instructions the participants receive, user interface improvements, if any, individual differences between participants (qualification, background, computer knowledge, IQ, etc.), distractions during the trials (sneezing, dropping things, interruptions, etc.), participant motivation, hints or interventions from the experimenter, with relatively more emphasis placed on the replicability of the results and environmental validity.

Statistics give hard numbers, but hide the qualitative feedback the users may offer voluntarily. For this purpose I found documented the use of surveys: **questionnaires** with open and forced choice questions or **thinking aloud** studies. Close questions use binary (yes/no) answers, multiple choice or the Likert 1-5 scale of agreement with the question. **Open questions** and thinking aloud studies require coding to structure and summarize the content of the user report. The latter methods provide information which the experimenter was not expecting.

A somewhat newer technique based on thinking aloud is **co-discovery learning** (Zhang, Z. (1998)). Two participants are used at the same time and they're asked to collaborate in solving a problem (or using an interface.) Instead of the weird instruction to 'verbalize their thoughts', the users are asked to explain to each other what they are thinking about while working on the given task.

Finally, a very recent theory, (CE+ by Lewis & Polson (1995)), spawned a method of assessment of the usability by non-experts of an interface: **cognitive walkthrough**. The model of the user details four phases for the user doing a task: setting a **goal**, searching the interface for available actions, selecting an **action** which seems to **achieve** the goal and evaluating system **feedback**. The evaluation procedure described is a sort of collective thinking aloud, performed by the interface designer and a group of peers (so actual participants are not involved!). This group evaluates the following items: the **accessibility** of the controls, the **match** between the goal and the label of the control(s) used, the **feedback** provided by the action.

After an interface (or another software product) is developed, HCI researchers may study users in their daily application of the product, **field testing**, to gauge additional information and measurements in the most ultimately valid environment: actual user sites.

Surprising as it may be, I could not find any HCI literature on **attention-grabbing**. This field seems to concentrate on efficiency and scarcity! The only mentioning I met of an argumentation for more glamorous interfaces, was in some statement of purpose of Kai Krause of MetaCreations Inc. Everyone else advocates compatibility over boredom and leave the beautification principles in the court of advertising journalism and multimedia. In his introduction to a course in HCI at Cambridge, Prof. Alan Blackwell even comments that "The software of ten years ago was considered attractive at the time, as were the cars of twenty years ago. The sexiest software of today will look just as old-fashioned before long." I find his statement evasive, negative thinking: a good interface is a good one; I've no idea who considered attractive the command line of ten years ago; maybe he was referring to the Apple GUI or something… In that case, it's not that it got old-fashioned, but it matured and it gathered more and more appeal. I still enjoy the 8 years-old interface from the LucasArts adventure game "Day of the Tentacle" and I don't think any software since then reached its simplicity, and 'sexy' usability.

General comments on HCI literature

I mentioned at the beginning of the HCI topic that several factors affected the drift of the software industry away from the principles described in HCI theory. In perspective, looking back over the papers, course notes and articles I read on the topic, I can see another paradoxical factor which may have influenced that drift: **research usability**. It is interesting how HCI theorists do not follow some of their own advice: the research papers are heavy with numbers and relatively few explanations, the course materials are going back to the beginnings, crowding an already multidisciplinary field with notions which dropped out of use long ago, and the summarizing and advice articles become heavy with words (just as this paper does in its unfolded version)

I didn't find any HCI literature on the effects of separating documentation from the application itself; even professional authorware packages use separate documentation. Surprisingly, MonDoc might be a first in allowing the user to **include the documentation in the product** itself – or the **product in the documentation**. I wish someone would disagree with me on this topic so I can learn from someone else's experience too. All I can study is the result of the work of project managers and programmers of existing mainstream software. They prefer to simplify their task by providing disturbingly non-descriptive error messages and separating documentation in a separate module; sometimes, within large projects the documentation module is written by a completely different team, making the conceptual gap between application and documentation even larger. During my experience as a software support technician, I isolated this separation as a major obstacle at the user side in learning specific applications. As a result, I started developing MonDoc, a document editor which allows writing the documentation within the application – or vice-versa, as is the case with MoStaCon.

Also, in the English-centered virtual world of computing, **cultural differences** tend to be neglected for a number of reasons. (1) Creating a Babel of languages reduces the possibility of understanding between researchers. (2) To avoid reason#1, one language has to be chosen for international cooperation. (3) English, French, Russian and German were historical candidates (4) Since the US of A are the most obvious candidate (large, resourceful, research-oriented nation which finances several international research funds), the chosen-by-default language happened to be English. There are almost three times more psychologists with English as the mother tongue than any other language. In a way, research happens where it is needed, thus where there's funding money.

However, the world shows a tremendous **diversity** on many grounds (economic, social, political, religious, philosophical, etc.) In turn, this diversity influences local ways of thinking, of understanding the world and subsequently, of action. That means that an Indian or a Chinese will find software tools written in the West as strange and uncomfortable to use as the other cultural products – like tall chairs and forks. Similarly, a psychologist will have trouble using the software written using accountants or engineers in the usability tests.

The solution implemented by big-budget projects is to develop a product for a specific nationality and then to **localize** it to countries (see Apple and Microsoft products) or specialist groups (see Statistica, SyStat) which have a good potential of returning the investment. Unfortunately, localization does many times only a translation job, with no research done to see how that specific group interacts with the resulting system. American psychology tends to dominate the literature through a generally active governmental process of worldwide dissemination of information. McConnell, J.V. & Philipchalk, R. (1992), say (p.599) that this "domination has led to a certain amount of provincialism and isolation" of psychological research of foreign countries. Most notably, "most research reports by American psychologists do not mention a single foreign source." As a result, user models developed by American psychology are very localized to the groups which were used in the original studies.

Unfortunately, until psychologists outside the United States stop using the results 'on the tap' of American literature and learn to develop, and implement **independent** scientific psychology, with studies on the local population, and until software companies start using the results of that research, there is little hope for successful localization, and software will continue to silently repel international differences in the human way of thinking and action.

My experiment was run on a large distribution of researchers – so it may work for a pretty large population of researchers… with the mention that they should know English – or find someone with enough time to adapt and translate the interface and the MOnitor primer.

1.2.4. Existing implementations

Let's take an example: the MacOS **balloons** system, one of the first implementations of HCI research at the Palo Alto Research Center. They are hints which can be turned on and off; while they're on, any time the user points to an interface element which has 'balloon help' programmed, a cartoon-style balloon pops up with a description of the element, suggestion for its use or even links to further, more detailed help in the accompanying help module.

Windows, though, doesn't have a system-wide default for such a feature, and leaves the decision to use it, and the implementation up to the individual applications. The MFC package provides a solution for dialog boxes in the form of a **question mark** on the title bar of the dialog box; the user is supposed to click on that, then on the dialog box element s/he wants information about.(not intuitive + too many actions) Microsoft Office developers, maybe under the influence of development on the Mac platform, implemented

pop-up tips for toolbars and even within documents...(much more intuitive) Later on, the same team expanded the idea of **Wizards** (applied mini-tutorials which walk the user through a set of predefined tasks), to another level of help with the context-sensitive **Assistants** (initially a good idea, but over-implemented too crowded, too pushy in its implicit affirmation that it knows better than the user what the user needs. To add to their unfortunate fate, the technology behind Assistants is virus-prone; see ZDNet, (2000)). A variant is the simplest type of adaptive presentation which turned up to work well and many applications implement, is the **tip-of-the-day** information about things which may be done or about non-evident ways to simplify actions.

Such a diversity if implementations is confusing and distracting. the user is offered information (unfortunately sometimes conflicting), on too many channels and s/he gets confused or overwhelmed.

A problem which often hinders communication at any level has at its source **projection** (the social-cognitive feature which can be resumed as 'If I know this, everyone else should know it too'.) This feature is conditioned by a common social source. We learn that the others in our group are like us, and this subconscious piece of knowledge has a strong tendency to overgeneralization. It makes us suppose that if we know, do or like something, everyone else should more or less know, do or like the same things.

Lacking **psychological research training**, in their haste to meet deadlines, programmers get used with sparse hints and bad documentation, so, unless trying hard to do it, they generate the same bad interfaces. Something of a GIGO effect, coupled with the a cognitive problem detailed above (projection). The situation is embarrassing, since there is plenty of research available on the topic.

But it's not only about software documentation. As I mentioned, there are many **document authoring systems** which allow users to structure the presentation of specific types of information. Some examples and HyperCard on the Mac, multimedia authoring tools like Director or Flash from Macromedia, electronic textbooks like InterBook. Why do I consider MonDoc superior to these? First, the engine and the document can be merged in one object. Second, the portability of the system depends on the lower layer (that is DHTML in Internet Explorer), so there is no need to develop different 'players' or plugins. Third, the user can modify the document without the need to start another program (the editor is part of the engine.) Fourth, the size of the system is insignificant (hundreds of kilobytes as compared with tens of megabytes) Fifth, MonDoc allows the user to include the whistles'n'bells of a Director or Flash animation, Java applet or any other Web technology.

1.2.5. Comparison with InterBook (an existing ET system analog to MonDoc)

Prof. Brusilovsky from Carnegie-Mellon University is working for almost four years on InterBook. As a result, that system is much more mature than MonDoc; from the several articles I read on the topic, the main principle behind IB is **adaptive navigation**. This seems to be a hot subject in HCI for a couple of years now – see Eklund, J., Brusilovsky, P. & Schwarz, E. (1998). Unfortunately, in order to include those principles in MonDoc, I had to use slightly more intricate DHTML, so the prototype version presented here does not contain it. There was no actual need, though, since the effect of adaptive navigation was thoroughly verified in international research (France, USA, Australia), presented in several articles (see the reference section of the article mentioned above)

So what's **adaptive navigation**? It is a specific technique aimed to help users find an appropriate path in a learning and information space by adapting link presentation to the goals, knowledge, and other characteristics of an individual user.(Brusilovsky, P., Eklund, J. (1999), §1) That means that IB (for example), is using links marked with small icons which reflect the state of the page behind that link: if it was visited, how often, and, based on knowledge of previous documents read by the user, which are the expert-suggested paths through that document. In contrast, MonDoc doesn't yet follow that strategy. To begin with, the effects observed by the authors in the mentioned article are so little in favor of the link-marking method that it seems too much effort and a heavier presentation for too little user leading. What MonDoc supplements is **semantic lead**. The links are replaced with buttons, for a more active-oriented presentation, and the buttons are included in the text of the page, mainly as verbs or keywords. The **verbs** supplement the hint inherent in the iconic form of the button and describe the action that would follow a mouse click over the button. The **keywords** invite the user who doesn't know their meaning, or wants extra explanation on them. In a way, the markers used in InterBook are still present, but they are incorporated in the semantic content of the text.

There are two main points to which published research on InterBook has mentioned but has not implemented. In the next chapter, I will explore in detail the relative benefits of **links vs. buttons** and **plain text vs. emphasized text.**

Just as MonDoc, InterBook implements **glossary** entries, but it does so in a classical manner: the user has to manipulate an index in order to find the needed term. The glossary does indeed appear in a separate frame, so the InterBook implementation does preserve the context of the description too.

A more technical difference between InterBook and MonDoc is the way of reading documents: **server vs. client** In order to use an Interbook document, you have to be on-line and to be able to connect to one of the CMU servers, or you may install a server on your own computer (if you have a Mac, that is.) MonDoc's use of a server is optional. Reading can be done locally, and editing the document is possible, even though without a server, the author has to save the modifications manually (copy-paste)

I have already discussed the cognitive and technical advantages of a **single document** over a **web area.** This difference does exist between InterBook and MonDoc. My implementation agrees with the HCI principles: one document, one file. Also, the concept of server is "the element which supplies documents" – not parts. Intensive traffic may cause problems to servers which instead of doing their job of serving document, spend time in processing them.

In order to end this comparison on a positive remark, I left for the end the superior **usage of HTML frames** implemented in InterBook. Currently, MonDoc creates a new window and closes the old one every time the user moves from one page to another or regenerates the page. The process is somewhat confusing, especially for first-time users. I am considering placing it in a set of frames and renouncing the even more confusing use of two windows (the Console, which stores the document database and the MOnitor, which is the dynamically-created current page.)

1.3. Summary

I will put together here the proposed set of principles (MonDoc) and its proposed implementation.

MonDoc

Current literature on ET (electronic textbooks, not extra-terrestrials :) considers the current technology employed by hypertext as the best available, and concentrate on details of the navigation (the 'hyper-' part), leaving the '-text' part up to the experts in textbook-writing, the authors. But, as I discussed above, in the section about redundant hypertext, writing text has many aspects which can be automated.

Page context (headers, footers, relevant information) automatically generated on every page;

Adaptive presentation generates the page from pieces according to the level of detail set by the user;

Removed contextual **clutter**;

Access to references, notes and glossary entries, **in the context** where they're used;

Encapsulation of document and engine (and reduced file size);

Graph-oriented presentation as opposed to serial presentation;

Variations on the same theme (**rephrasing** of important material, **informative** error messages)

Few ideas per page;

Emphasized key-words;

Active path markers (buttons).

The influence on performance of the last two principles is evaluated during the experiment presented below, in chapter 2.

MoStaCon

A researcher, or a research group (the user) could take a 'standard' version of MoStaCon (or even an empty one, just the engine), and start adding relevant information to it as s/he encounters, practices and tests it. Using this strategy, the information needed will always be there for consulting, in as concise and structured manner as the user makes it. I have met people who do this on paper. Fortunately, the digital medium is easier to modify, allows automation (which allows storing procedural knowledge as well as declarative) and minimizes access times.

Compendia, dictionaries, thesaura are all specialized versions of using one organization principle in order to provide access to a large amount of data.

Graph-oriented **decision tracking**;

Priming with menus and multiple-choice questions;

Teaching by example and animations (targeting faster episodic memory);

Exposing the **modular structure** of the information presented;

Recording information from the user in **structured**, relational ways;

When necessary, prompt for **directed or undirected** thinking;

And now it's time to test these principles.

2. Testing MoStaCon: the usability study

When I write software I like to know whether it's useful. In the case of MoStaCon, I wrote a small prototype in which I implemented few of the principles it inherits from MonDoc (emphasis, structure, action-oriented active objects, some adaptive presentation, contextual notes, etc) and few of its own principles (some transparent statistics, the experiment log, a sample DM graph.) For control purposes, I made two versions of the prototype between which I distributed in a controlled manner pages which implement those principles and pages which don't. See the **task, conditions and stimuli** section for a detailed description of these two versions.

How does one design a **controlled study** (an experiment) to assess the influence of a **set** of principles on a varied, lengthy and heterogenic process such as research? Normally, researchers reduce the span of their interest to include only 2-3 factors at a time. This tactic makes the results easier to gather and interpret and the resulting presentation more understandable. Unfortunately, the described tactics oversimplify the model of the participant; as a result, the experiment needs many pretests and post-tests to insure the 'control' over the conditions in which the experiment was conducted, namely the factors which might influence the performance of the participants. Fine approach if the model one uses is modular and if the researcher can abstract most of the model as s/he studies one part. This may happen in psychophysics, if one studies reflex responses to sensory stimuli and there mainly by using a bottom-up model, strongly behavioristic, and if one chooses to neglect the influence of internal factors like attention, fatigue, stress, (and higher, less-studied processes, like parallel thinking) the interactions thereof, or assigns them to 'variability'.

Obviously, in the case of the complex processing which happens in the privacy of the researcher's own mind (sic!), this approach may be inappropriate. The alternative, in order to **gather sufficient evidence** for the superiority of a tool (MoStaCon) over a set of tools (the wide variety of methods of research and of software used during research for gathering, formatting, analyzing and reporting), was to incorporate several experiments in one session in which all participants researched one and the same artificial question.

There are two more reasons for which I favored this approach. Researchers are busy people (ars longa, vita brevis); I felt uncomfortable to ask a large number of researchers to participate in several separate sessions of experiments in order for me to be able to gather data. This approach would have lasted several years, anyway. The second reason was that I had started this project in order to crystallize my knowledge of experiment design, data formatting and analysis for the benefit of the Research Laboratory of the Center for Cognitive Science at the New Bulgarian University, where I worked for the last three years. I wanted to leave for them a well-documented tool which could cut corners (specifically in data formatting), the way I was doing by programming individual scripts which prepare (rearrange, combine and verify correctness of), raw data for statistical analysis.

2.1. Preliminary survey and Discussion

But before going in the details of the experiment itself, I will mention the pilot study I made for it. In order to determine the way MoStaCon's target audience actually applies the theory of experiment design and in which ways they use the already existing software I did a survey. The full text of the questionnaires I used is available in Appendix 2.

Part 1: Free recall on specific topic. During this part I expected the participant to get into the context I'm interested in: the participant's memory of the situations in which s/he was supposed to design experiments. Also in this part the participant provided insights usable to interpret their responses to some questions in the next part.

Part 2.Directed recall. During this part I probed the experience of the participant on various topics related to experiment design, such as general phases encountered while running an experiment, or how they prefer to run experiments, on-line or off-line.

Part 3. Remarks and debriefing. During this part I checked how the participant considered the pilot study s/he's been through and I gathered any problems s/he may have encountered during it, as well as capturing any related information, not included in the questionnaire in Part 2.

Procedure and apparatus

In their daily working environment (lab, office, classroom), the participants read and completed a questionnaire following the **thinking aloud method**. I used a cassette recorder to obtain their productions. The quality of the resulting recording was **improved** (digitized, amplified and filtered for continuous environmental noise), using CoolEdit, then compressed with MP3 Producer. I listened to the resulting MP3 sound files with WinAmp, and using its Lyrics plugin, I **typed in** the coded audit trails; the Lyrics plugin **automatically added time-labels** to every coded phrase. Later on I used these labels to reconstitute response times. The **coding** consisted in rephrasing the stuttery, redundant speech of the participants into coherent, formal phrases which I later used in conjunction with their **time labels** in order to calculate RTs. For details on this, see the Measures section below. All operations took place on an IBM-PC-compatible, under the Windows98 operating system.

Discussion

Besides the information which I used in order to augment the list of empirical research steps (see Appendix 4), the survey provided both expected and unexpected information. (diverse opinions about the use of automated research and analysis tools; different groups approach the task differently – experienced/perspective vs. students/theoretical;)

I expected that every participant will tackle the task with the specific approach and design they used in their recent experiments, and the audit trails confirmed this. (the T-test is two-tailed, for independent variables.)

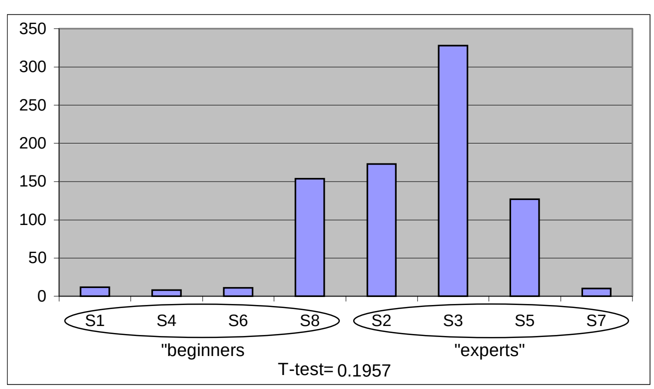


Figure 9: Number of references to recent experiments made by each participant ('experts' on the right, 'beginners' on the left). Participants #3 and #8 talked almost continuously about their own work.

2.1. Task, conditions and stimuli

To begin with, choosing a task was a challenge: to find a problem which could fit the domains of study of all the participants, and then twist it in such a way that every participant may find it non-trivial enough to arise a small level of curiosity – necessary to keep their attention on the task. I finally settled on sketching **an ad-hoc, bogus** 'meaning-detection theory' based on psycholinguistic theory (lexical priming), and on the psychophysical signal-detection theory.

The participants received a page with printed **instructions** followed by the task itself. The task was to answer the question "How does using one eye only (2-dimensional perception) as opposed to using two eyes (3-dimensional perception) influence the threshold of understanding written one-syllable words (in college students)?" See Appendix 2 for the full text of both versions of instructions used.

In the **control condition**, they were asked to use the procedure they normally employ for designing an experiment; in the test condition, they were asked to use MoStaCon to design an experiment. In both conditions they had to answer **the same question**. The 'even' participants had to solve the task in the test condition first, the 'odd' ones had to perform in the control condition first. Before starting the control condition I ran a short **questionnaire**, to assess their level of familiarity with the software used, to make sure that they all have a similar level of knowledge necessary to solve the computer-related parts of the task, and to bias them uniformly toward running t-tests for their analysis (since this is the only discriminatory analysis implemented in the prototype of StaCon). For the items of the questionnaires for which they didn't know the answer, I explained in **formal detail** (declarative knowledge) and by showing them within the MOnitor and in Excel exactly how to perform the task (procedural knowledge). After the control session, I asked them for their **subjective assessment** of how familiar the procedure was.

Before the **test session**, they had to get familiar with the interface, then to answer another **questionnaire** about different elements of the interface. During the test itself they had a choice of following the lead of the experiment-design tutorial or using the 'expert' interface (choosing the steps from a pop-down menu), in order to answer empirically the question mentioned above. After the test, they made two **subjective assessments**, one of the familiarity of the procedure and the second about the simplicity of the procedure as compared with their usual procedure. Finally, I asked them for their opinion on MoStaCon and I eased that into the **debriefing session** for the participants who were interested.

While they were getting used with the interface, I subjected them to the conditions for the two experiments in which I am testing two of the MonDoc principles: They had to **read** through 6 pages of the MOnitor prototype, among which I had distributed in a controlled manner navigation **buttons and links**; three of the pages were not using any **emphasis** while the other three had the key-words in every phrase highlighted (bold face). For control purposes, there were **two complementary versions** of the interface (moa.html and mob.html) Where there was a button in **moa**, there was a link in **mob**. The pages which were using emphasis in **moa**, were plain in **mob**. I distributed these versions randomly among the participants (so that the interface-version condition doesn't overlap with the test-control order of the instructions), but because I was targeting a subject pool of at least 10, I have a skewed distribution in favor of **mob** (5, vs. 3

moa). In order to be able to apply t-test, in the experiment (2.6) which uses the skewed variable (buttons vs. links), I will use only the data from the first three participants who performed on **mob**.

Participants

Since the experiment is directed at **people who do experimental design** in Cognitive Science and the related areas, my choice of participants, and the pool thereof, was very limited only 18 people who accepted to participate. Due to the time constraints of some of these and other technical problems, I was forced to settle on a sample size of 8. Such a **small number of participants** may seem problematic to some, since it restricts my analysis to small-n methods. If it seems so to you, think that fewer participants in any analysis reduce the level of F (through the number of factors), thus making p (level of significance) larger but the results of the analysis more usable. Also, the number of empirical researchers to which I have access is relatively small. In gathering usable data for 8 participants (out of 14 attempts) I came pretty close to using the whole population available to me, even with the bonus presence of many foreign researchers during the Summer School. Due to technical problems (heat-induced computer crash, schedule incompatibility, recording errors), I had to ignore the results for 6 of the participants who participated. Many of the experts were, as expected for experts, very busy, and this made the task of finding participants and running the experiment even more difficult. The worse problem that these technical errors gave me is that it left me with an **unbalanced set of conditions** for one of the experiments, which I had to solve by further reducing the subject sample for that particular analysis – the section on the effect of action-oriented navigation (2.6.)

On the bright side, due to the **very diverse sample**, the results should have **more external validity**:

The table below shows a large range of ages (22-64), **approximately normal-distributed** since the MOnitor primer targets mainly students. The distribution of the participants' **field of study** emulates well my targeted audience (see the table), and a good distribution of **nationalities**, thus increasing validity along the cultural axis I mentioned in the previous chapter.

I controlled for **language proficiency** (the quality of read and spoken English), by choosing non-native speakers who could fluently express themselves in informal discussions. I agree that this method is somewhat unscientific, but I'm ready to compare it with any other proposed method for cross-cultural studies.

As shown in the table below, the participant sample was **perfectly balanced** over the test and control groups for qualification, the first age group (most numerous) and gender. Due to the small sample, the field of study happened to be balanced only for Cognitive Science and Psychology between the control group and the test group for the experiment on navigation (2.6.). Participant age was definitely skewed to the left (towards the low age group), and I could not find a female participant for the high age group; I attribute this to the cultural bias of the research community during the 50-60s which more or less formally rejected women. I limit this explanation to the European research community, since the participant pool is European.

					language	
condition	field of study	qualification	age group	gender	native	most comfortable
control:4 test:4	cog.sci:3 psychology:2 cog.ling.:1 psych.ling:1 physiology:1	M.S.:5 Ph.D.:2 B.A.:1	20-30:5 30-40:2 60-70:1	F:4 M:4	Bulgarian:3 German:2 Russian:1 Ukrainean:1 Armenian:1 Polish:1	English:5 Bulgarian:4

Table 1: the distribution of participants over control variables: somewhat reduced internal validity in order to gain external validity

Procedure and Apparatus

For the main test I used the same **procedure** for recording participant production as in the survey, but I skipped the time-consuming step of digitizing and I recorded the productions on-line, with a microphone connected to the computer, directly in CoolEdit. Also, following an example from HCI literature (Zhang, Z. (1998)), I used a **modified** version of the thinking aloud paradigm**, co-discovery learning**: I asked the participants to explain to me every step of the process they were going through. This made the task somewhat more internally valid than the normal thinking aloud paradigm: people don't normally speak continuously while they think, so the additional task adds extra processing and the measured variable is distorted. For a longer discussion on this topic, see the reference mentioned above.

For the **control part** the participants used the question, paper, pen, Excel on computer, my notes on the bogus theory and an optional textbook in research methods (the same I used to generate the information space of the MOnitor). I used a microphone to record the speech of the participants while they were solving the task using their own knowledge, the materials and the software they use normally.

During the **test part** they used the MoStaCon prototype displayed by Internet Explorer on the same computer. I used the same microphone to record the spoken train of thoughts the participants shared with me as they were solving the task using the prototype.

The **setting** of the experiment was a quiet (but not sound-proof), classroom, ensuring a low level of random environmental distractors in the background (chirping, muffled discussions on the corridor, etc). Due to the time constraints, I could not control strictly the tiredness of the participants. I did make a note of this variable, though, with three-levels: fresh (for the participants who came in the morning), alert, and tired (for those coming after dinner), for further analysis.

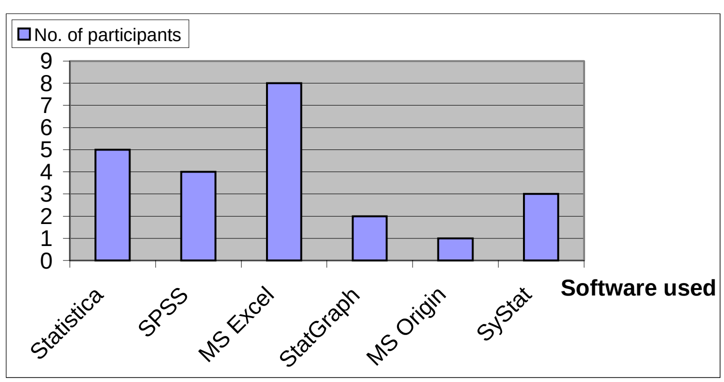


Figure 10 shows the distribution of analysis software used by the participants in the survey

I also had to reduce the many types of software people use, to a common denominator. From the pilot study audit trails, it turned out that most people used Excel, for formatting data, filtering trials and simple analysis.

Further, in order to assess the prior proficiency of the subjects with the software and to achieve a smaller variation of the amount of expertise the subjects had prior to running the experiment I submitted the subjects to a quick test with answers available after every question. These questions were given quasi-randomly. Randomly in order to avoid confounding the usual two control variables: task and order. 'Quasi-random' because I wanted to keep the sensitive questions in the middle of the test, so as to use the serial position effect in order to reduce priming. Sensitive questions are those which may provide answers to be used directly in solving the task of the experiment, like 'how do you calculate a t-test?' for Excel and 'how do you declare a variable' for MoStaCon.

Of course, I had to account for the order of running the experiment: half of the participants solved it first using Excel, half used MoStaCon first.

All operations took place on an IBM-PC-compatible, under the Windows98 operating system.

2.3. Measures, dependent and independent variables

From the protocols, I extracted qualitative data and quantitative data, each treated as follows.

Qualitative data extracted was: details of the process of experimental design, methods and paradigms used by the participants, subjective appreciation of existing software, opinions for improvement of experiment-related software, etc. I used this sort of data for making adjustments to MoStaCon, for clues to interpreting the quantitative data and for the insight and discussion on the informal aspects of scientific research.

In turn, **quantitative data** consisted of the values of the answers to multiple-choice questions and of RTs (Recall Times for the control condition and Response Times for the test condition.) This data went into simple t-tests in order to provide hard numbers in support of the trend observed in the qualitative data. These are the independent variables used throughout the study:

- **participant number** (discrete- 1-8),
- **solving condition** (binary- Excel first vs. MoStaCon first),
- **order of presentation** (binary- Excel first vs. MoStaCon first),
- **version** of the interface used (binary:)
- **access** (binary: leading vs. unguided choice)
- **page** under test (discrete: 1-6)
- **cognitive load** (discrete: the level of cognitive load of each page)

In order to verify the sources of variability in the experiment on emphasis (2.5.), I calculated four measures of **cognitive load** for each of the 6 pages used in that experiment: **information load** (number of distinct ideas per page, averaged from the values counted by two independent judges), **pragmatic load** (number of verbs per page, including verb-forms), **morphologic load** (number of words per page) and **perceptual load** (number of characters per page).

I know that when solving the task for the second time, the participants already have an idea of how to solve it, and they may just go through the motions of using the method. Incidentally, **it is mainly these motions that I'm interested in**. In HCI the accent falls on how the user understands and applies the interface, not on what goes on in the black box. As you read in the previous chapter, HCI user models are strongly behavioristic. Anyway, I was forced into a **within-subject** design by the reduced participant pool; this is both a positive and a negative fact, according to me. **Negative** because of the priming involved, **positive** because the assumptions on which between-subject methods are based (all humans think and behave similarly), are very suspicious to me.

These are the dependent variables I measured:

Subjective familiarity with the software used, using the Likert 5-point scale (5=most familiar)

Subjective appreciation of MoStaCon method simplicity, using another 5-point scale (1=more complicated than the usual procedure, 5=simpler than the usual procedure)

Instead of the number of errors (NE) used in classical usability studies, I used the reverse measure: **empirical accuracy**, as the number of experiment stages mentioned (count of insight -aha- events, after some time spent by the participant in a 'what's next' state; see Appendix 4 for the complete list of stages I used) I consider this a better measure of performance than the alternative, and it's easier to specify increased performance through increased accuracy. Also, since there are no a-priori lists of all the steps possible in empirical research, it was less internally valid to calculate the number of errors (by subtracting from an arbitrary number.)

Speed: measured by the reverse of the recall times (RT) necessary for remembering each 'next' task goal (in seconds; see the discussion of the study for the reason of the coarseness of this measure)

Goals of the experiments

Besides applying (beta-testing) MoStaCon, the experiment I ran was aimed at verifying the ideas behind MoStaCon. More specifically, I examined the basic claim that **performance** in solving the task **is better** (especially for beginners) **when using MoStaCon** than when using the tools they are already accustomed to, through the following hypotheses:

H1:. When using MoStaCon, researchers forget fewer steps of empirical research than when using traditional research design sessions

H2: The principles behind MoStaCon allow experimenters to do their work faster and with less technical error then when using the standard procedure (recall and linear, or even hypertext books, and general-purpose statistical packages, like Excel's AnalysisPack.)

Data treatment

Initially, I was thinking about a 2x2x2 Anova design, as in the figure below, in order to verify not only individual effects on the dependent variables, but any possible interactions among the factors.

Software	Tutorial (adaptive presentation)	Action-oriented links	Emphasis
MonDoc	Y	Y	Y
	N		
InterBook	Y	N	
	N		
DHTML	Y	Y	N
	N		
HTML	Y	N	
	N		

Table 2: shows the attempted Anova design and the associated 'evolution' of hypertext tools leading to MonDoc

Since this option was not supported by the analytical method, given the small-n character of this study, I decided to keep the analysis as a series of means comparisons (one-tailed Student t-test), which I present in the following three sections. This is also consistent with previous research in the area (see Brusilovsky, P., Eklund, J. (1999))

The effects I'm mainly interested in are in the way participant performance (**accuracy**, **speed**) varies between the different manipulations in the high-level perceptual conditions (**emphasis** and **action-oriented navigation**), I'm going to present these first, and in full. However, there are other effects which I planned to observe: how the individual participant's field of study, qualification, age, and even native language and sex influence performance. From these effects I will present only the significant ones.

2.4. Results of MoStaCon vs. standard research

There were significant mean differences in the predicted direction for almost all comparisons I made. The subjective opinion of the participants also supports these observations.

Subjective appreciation of **familiarity** was checked from participant answers on a 5-point Likert scale. The difference in means, $\bar{x}_{MoStaCon} = 2.625 < \bar{x}_{\text{standard research}} = 3.75$ was verified with a one-tailed t-test for paired observations (t-test=0.0398), which **confirms** the expected unfamiliarity of the users with MoStaCon, significant on a 95% confidence interval and **underlines** the next result:

The average of the subjective assessment of MoStaCon's **simplicity** was $\bar{x}_{MoStaCon} = 4$ on the same scale (Likert 5-point.), with a standard deviation of 0.8997. This places the result's 95% confidence interval above the level of indifference (3 on a 5-point scale). Thus, even if MoStaCon was unfamiliar, it was judged **significantly simpler to use** than the standard research method.

The **objective results** were as follows.

Accuracy: the average of empirical research steps mentioned while using MoStaCon was 19.625 (with a SD of 3.02) and in the control condition it was 16.25 (with a SD of 3.26). The calculated T-test of 0.0724 does not insure significant difference between the two series. But an analysis of variance shows significant independent effects both from the order of presentation (meaning that when MoStaCon was used first to solve the task, the accuracy was significantly higher) and from the condition (when using MoStaCon the participants have been significantly more accurate)

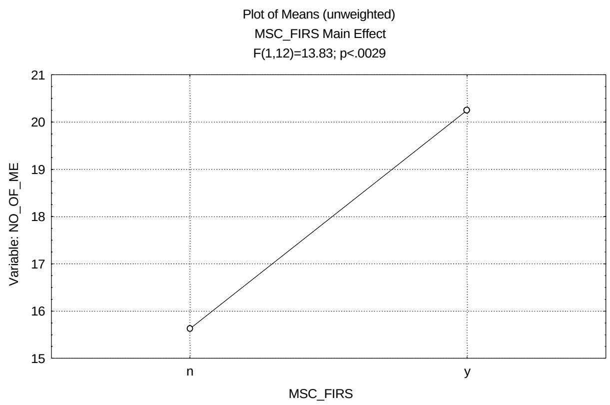


Figure 11: Effect of the order of presentation (MSC_FIRST) on accuracy (NO_OF_MENTIONED_STAGES)

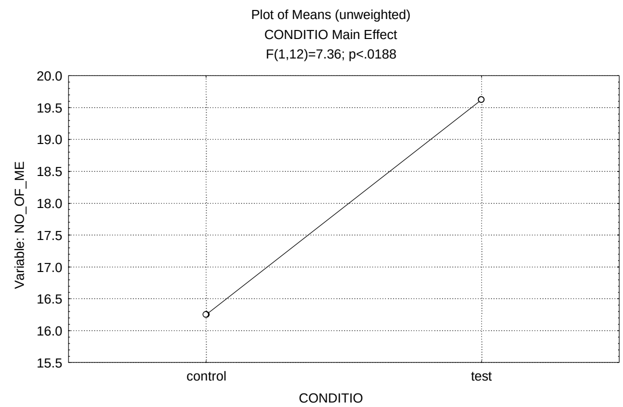


Figure 12: Effect of the CONDITION on accuracy (NO_OF_MENTIONED_STAGES)

In the speed analysis I included only the stages mentioned by all the participants (10 out of 28).

Speed: the RTs were significantly (T-Test=0.0044) smaller for MoStaCon with an average of 16.775 (and SD of 11.49) than the control condition's average of 22.425 (and SD of 13.19).

While calculating accuracy I noticed a trend in the values, suggesting that the participants who had first solved the task in MoStaCon, had higher accuracy in the control condition. Like before, this prompted me to use an analysis of variance in order to determine the significance of this trend.

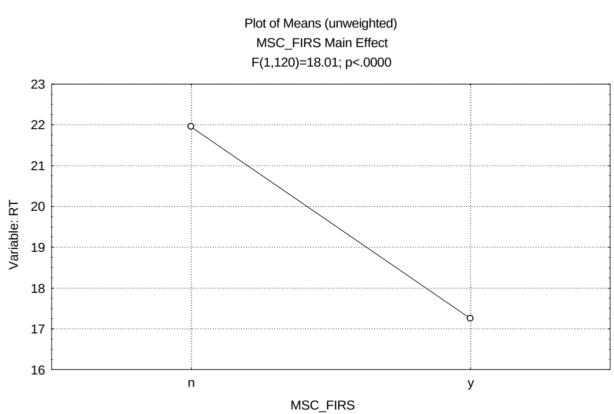


Figure 13: Effect of the order of presentation (MSC_FIRST) on the time of recall (RT) for all stages

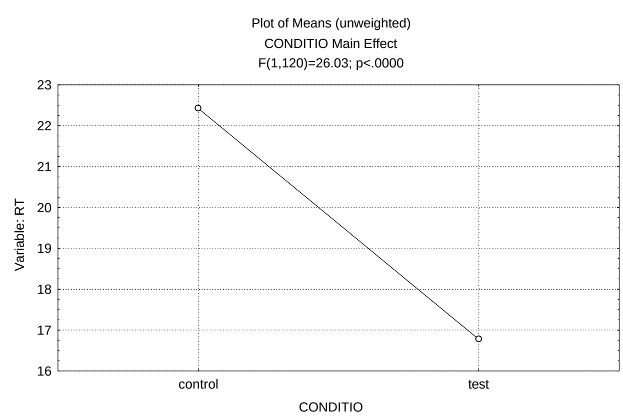


Figure 14: Effect of the CONDITION on the time of recall (RT) for all stages

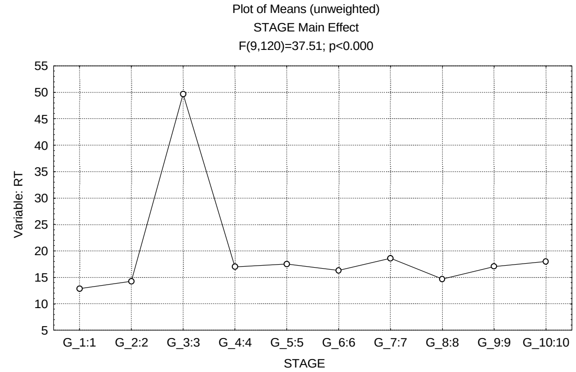


Figure 15: Effect of the specific STAGE on the time necessary for that stage's recall (RT)

It turned out that the speed of recall for the next stage was significantly dependent on all three factors (whether MoStaCon was used or not, the fact that MoStaCon was used first and even which stage was recalled). And there were no significant interactions between the factors, which is an indication that they are **relatively independent**.

It was interesting to see that the stage which took longest by far (the third that I used in the analysis) is **"drafting a hypothesis"**. See Appendix 4 for the complete list of stages.

2.5. The effect of graphical emphasis

In this experiment I was expecting to see a positive relation between graphical emphasis and user performance. See Appendix 2 for examples of two pages from the test and control conditions. The results in the following table suggests that such a relation does exist.

	answer	
	correct	incorrect
control	6	8
emphasis	15	3

Table 3: summary of answers to the post-test questionnaire about information presented on pages with emphasis or plain text (control)

As described before, during the test condition, participants were asked to read completely 6 of the pages from the MoStaCon Console and MOnitor. Some pages were plain, with no emphasis, on the other pages, the key-words were shown in bold face. This table shows the number of correct and incorrect answers given by the participants to questions from the post-test. The questions refer to material described on four out of the 6 test pages

When I discussed the Gestalt principles of similarity (grouping) and continuity, I noted that I expect the reading time per page to decrease in the test condition (because structured text is easier to grasp). Unfortunately, as seen in the table below, the results came out insignificant. I tried to find other measures of speed than time per page, so I calculated for the pages under test: **information load** (number of ideas per page, and the resulting RT/idea), **pragmatic load**, **morphologic load** and **perceptual load**. None of these measures got usable significance levels, but I included the table for the discussion the relative influence of emphasis on the speed of reading the units (page, idea, phrase, word and character).

			Control	Test	T-te
RT/page	Avg	240.333	224.458	0.144495	
	SD	47.715	54.563		
RT/idea	**Avg**	**13.403**	**12.499**	**0.1339**	
	SD	**2.560**	**3.002**		
RT/phrase	Avg	6.072	5.909	0.349124	
	SD	1.017	1.782		
RT/word	Avg	0.790	0.813	0.345089	
	SD	0.091	0.265		
RT/char	Avg	0.140	0.147	0.264	
	SD	0.018	0.051		

Table 4: different measures of page load and their influence

The strongest (almost significant) effect was, of course, on the speed of going through **ideas**. Next is the global effect, on the entire pages, then the perceptual task of reading individual characters and only last are the morphological and pragmatic means of conveyance of the ideas. Note that for words, the effect is even reversed (it took insignificantly less time to read plain words than emphasized ones.)

I guess I have to look at the way the individual item (page, idea, phrase, word) influences that time. Some pages are heavier, some ideas are more difficult than others, and that should add variability to the measurement. Since the only accurate measure I had was the time per page (the other ones are averages calculated from the time per page), that is the measure that I used in the analysis of variance.

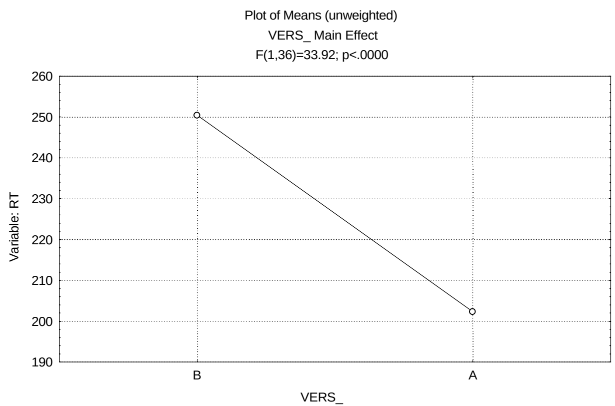


Figure 16: Effect of the version of the prototype (VERS_) on the reading time (RT) for all pages

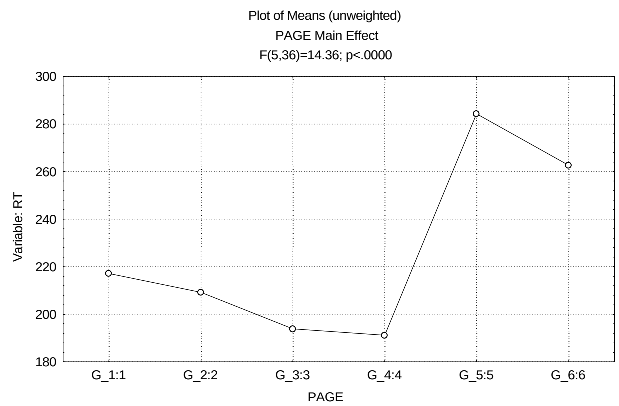


Figure 17: Effect of the PAGE on the reading time (RT) for all participants, both versions of the prototype

The result that I didn't expect was the fact that even the version of the prototype significantly influenced reading time per page. I will take a better look at this in the discussion section. The effect of the participant (not presented here), is not significant.

2.6. The effect of action-oriented navigation. Results.

The last effect I considered in this study was that of action-oriented navigation. As I discussed before, hypertext links are very poor at describing their own active potential. More salient are the buttons, introduced to the web by dynamic HTML. In order to test this assumption, in the moa.html version of the MOnitor 4 paths were normally accessed through buttons, like in the rest of the interface, while in the mob.html version they were reduced to hypertext links. I measured as RT (recognition time), the amount of time it took participants to identify the object as a path and to follow it.

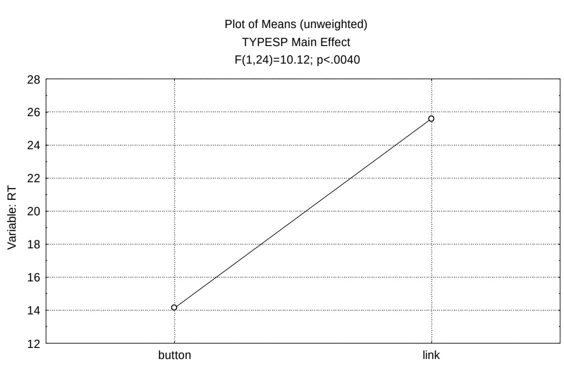


Figure 18: Effect of the type of path marker (TYPESP) on the recognition time (RT) for all markers

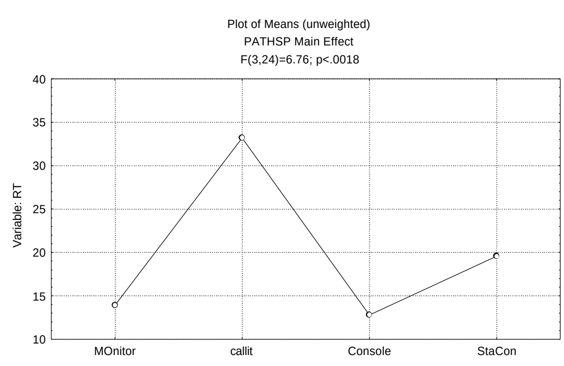


Figure 19: Effect of the individual path (PATHSP) on the recognition time (RT) for the respective marker, regardless of type.

Average RT for the test condition (using buttons) was 15.750 (with a SD of 10.162), while for the control condition (using links), it was 24.294 (with a SD of 14.614). The one-tailed Student test gave a probability of 0.0198 for the null hypothesis ($\mu_1 = \mu_2$), placing the hypothesis that participants are faster when using buttons just outside the 95% confidence interval.

I was surprised by the low effect, but the large standard deviation was a hint that built within the structure of the interface was a fact which could affect the analysis. My hypothesis, then, was that individual buttons have different functions and their text is formulated in different ways. To verify this hypothesis, I did an analysis of variance.

When ANOVA separated the two effects, both turned out to be significant, and as predicted, the participants were significantly **faster when using buttons** than when using links. As for the path labeled "call it up" (which seems to have introduced the variance noticed in the t-test), it looks like that the label was not descriptive enough and subjects had to go back and scan the sentence before recognizing the path marker.

2.7. Discussion of the study

Qualitative data from the protocols indicated for the majority of the participants (6 out of 8) that they appreciate the primer and that they see the need for a tool of MoStaCon's class to organize and summarize their empirical work.

The significance level of simplicity received for MoStaCon is larger than (for example) the one received in the InterBook study I mentioned (with averages of 4 on the scale of 5 and SD of 0.9 for MoStaCon as compared with a global 3.61 on the scale of 5 and SD of 1.3 for InterBook). Of course, the results on InterBook are more valid externally (but only intra-culturally), since they have used 25 (North American) participants as compared with 8 in the case of MoStaCon.

The familiarity measure (2.62 for MoStaCon vs. 3.75 for the experiment's approximation of the real-life process), is somewhat more difficult to interpret. It may be that researchers are not used to be lead through the process of empirical research (especially by a web page!). It may be that the prototype didn't manage to capture many of the techniques used on paper by researchers. Qualitative data corroborates both points. On the first point, as already mentioned, most participants **welcomed the unfamiliarity** of adaptive navigation, while on the second point, they provided **useful suggestions** on how they would like to use the interface so that it **better emulates and helps** their empirical work. These hands-on suggestions were much more applicable than the results gathered from the survey.

The **comparison** made for the speed of mentioning the next step in the design is problematic because it is actually comparing the results of **two different processes** (recall times for the control condition and reaction time in a decision-making situation for MoStaCon). Nevertheless, when assessing performance, we are interested in the time measured, not the process which generates that time. The **contribution** of forced choice implemented in MoStaCon simply appeals to a faster process.

It was no surprise that the **effect of emphasis on the time** necessary for reading a page, I guess I have to look at the way in which the individual item (page, idea, phrase, word) influence that time. Some pages are heavier, some ideas are more difficult than others, and that should add variability to the measurement. But the effect visible in Figure 20 seems to me **the most interesting**. We know from the individual effects that the difference between the two versions in the experiment were given by the presence (or absence) of emphasis. In that case, could we attribute to **adaptation** the converging evolution of the reading time per page for the two versions? In that case, adaptation to what? To a variation in the content of the global stimulus (if we consider the page as such)? Or is there some other factor, like **boredom** (observed in the protocols), which increased the reading rate of the participants to the detriment of understanding? Somewhere in the introduction I have mentioned that under constant repetition, any subject could become a bore. Maybe boredom was an effect of adaptation and the increased speed of reading was a result of dwindling attention.

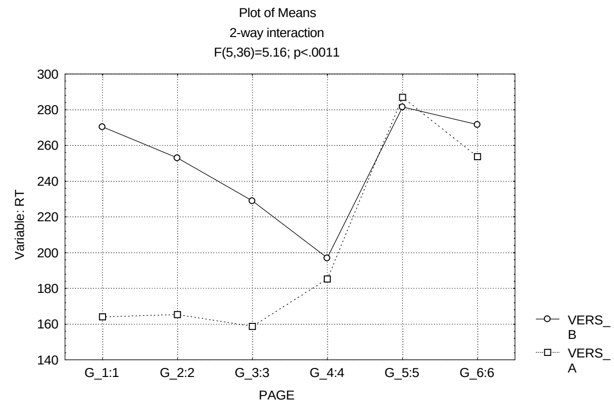


Figure 20: Interaction between the effects of the version of the prototype and a specific PAGE, on the time necessary for that page's reading time (RT)

In the last experiment, as it was expected from the previous analysis made by Brusilovski on InterBook, the **presence of buttons significantly decreases the amount of time** spent on making decisions about how to continue from the current page. If given the choice, buttons are used more often than links. Sometimes, in the link condition, I had to specifically point out what I expect from the participant to do, while in the button condition they were ready to press even before finishing the sentence in which the button was a verb or verb form. For my experiment, however, there is an alternative explanation for these results: the fact that buttons are the main object used to move among pages might have biased the participants into ignoring the links. The fact that the MOnitor path didn't show almost any effect with the type of path marker seems to support the alternative explanation. My counter-argument is that the wording before that path marker is unequivocal: "**To begin using MoStaCon**, click on: <u>Start the MOnitor</u>" At the same time, the 'call it up' path, which was also among the first of any kind they encountered on the page, was of the slowest of them all, and the Console (the absolute first), shows (in an of itself) a significant difference. If we add the fact that before they came to the experiment, the participants used links much more often than buttons (due to the nature of standard HTML), this eliminates the alternative explanation mentioned.

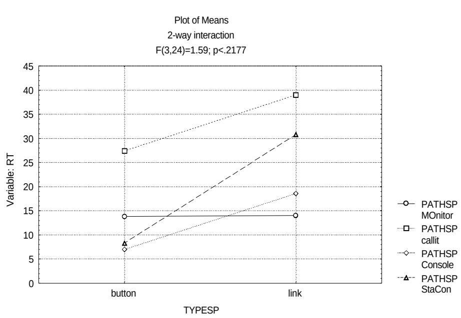


Figure 21: Non-significant interaction of the effects of the type of path marker (TYPESP) and the specific path (PATHSP) on the time necessary for the recognition of that path marker (RT)

3. General discussion and conclusions

My expectations were not entirely met. The hypotheses presented under **Goals of the experiment** (section 2.3) were based on the amount of interface and process improvements which I built into MoStaCon. All these hypotheses were verified with the exception of the statistical significance of emphasis on reading time. But first, let's consider the survey.

Besides the data I gathered from the **survey**, after running it I noticed several ways in which I could improve the procedure itself. As I expected before pre-testing, some questions provided information for further ones. For example, the question about the types of information the participants are tracking during the process of empirical research, supplied results which had to be used in the next question (about the medium they use for storage.)

In these cases, it may have been better to structure the survey in several steps. First run the participants through Part 1 and ask the initial, general questions from Part 2, after that analyze the answers of the whole set of participants and prepare a more "multiple-choice" answer sheet for the questions in which This would create a 'context breach', because when starting on the multiple-choice part of the experiment, participants will not have their recall context built by the free recall of Part 1. This may be solved by preparing individual answer sheets for each participant: providing at the beginning of the multiple-choice part a short summary of the recorded recall of the respective participant. On the down side, individual answer sheets may introduce new factors in the design (length of the first stimulus, memory load, variable context, etc.), reducing the level of control of the survey and ultimately the comparability of the Recall Times. For this reason I chose to run the survey in one session and I had to help some participants understand some of the more intricate questions. In retrospect, the choice was not the best, but time constraints and the lack of appropriate participants prevented me from running a revised survey on a new set of participants.

For the **main test** it's interesting to note that absolutely none of the participants mentioned the steps involved in **assessing the validity** of the experiment. It was one of the problems which prompted me to start writing the MOnitor. Unfortunately, in the prototype I had placed the discussion on validity on a glossary note and very few of the participants actually read that note. This result is one of the strongest criticisms of the prototype which this study provided.

Another criticism, based on the **incompleteness of the prototype**, was that the step of comparing results with the literature was missing from the notes menu, it biased some of the few people who had mentioned it in the control condition towards not mentioning it in the test condition.

An interesting fact was that in their designs, some participants **didn't consider** at all **the composition of the subject group** according to some specific principle. Two of the participants took for granted the only constraint presented in the task description, with the assumed opinion that "subjects are there just to give us data." These are moments when advisors and project supervisors come in and say "but why didn't you control the subject groups?" I don't know whether I can make the MOnitor as selective as an advisor (after all, adaptive presentation gives the user only the information s/he asks for); as I mention below in the section on further developments, there are several techniques which can be implemented in that direction.

I will discuss now a few ways in which I explain the **amount of variance** recorded in the experiments, especially for the RT measures. As much as I controlled for it, I suppose **language proficiency** did have its strong influence of performance.

I tried to reduce the boredom and to increase their attention level by **comments** with which I was prompting them to advance in the analysis or giving them hints when they were stuck on some detail. This may have introduced variation, but since I was not doing it following a specific schedule, I consider it was random.

The participants used different **strategies** while using MoStaCon, resulting in increased variability and reduced significance for the global test. However, this didn't manifest itself significantly (the variable Subj didn't have any significant effect on anything)

The **seriousness** with which they approached the test was an unexpected factor. Some of them took it for real, doing their best under the artificially controlled conditions to solve the task; others just complied with my suggestions, visibly 'disconnected' from what they were doing (for example, some were using very large numbers of stimuli and didn't consider my warning that they will not be able to find actual examples with the control constraints they were imposing). I had to drop the less serious participants from the analysis; as some (analytically-oriented) participants pointed out maybe this was due to the **confused nature** of the experiment session. Several overlapping tasks proved to be heavy on their discerning abilities; even if I warned them not to think about the goals of my experiment, some of them (in some of the control conditions), kept protesting and explaining to me how the method I use may be inappropriate. Also the way they **recorded** in the experiment log: some jotted a few words in the notes area, others dutifully typed in complete and precise phrases. This may have affected the response times I calculated in the direction of reducing the effect (because typing and formulating consistent sentences takes more time). The same thing happened in the control situation (on paper).

In the control condition (paper+Excel), they tended to go back to previous steps when they remembered that they forgot some important detail (like an independent or a control variable); this happened less in the test condition (MoStaCon), but unfortunately I didn't keep track of that variable in order to see the interaction of order of presentation and the amount of **backtracking**. A more detailed analysis of the protocols may be necessary for this.

The **coarse unit** (seconds), which I used for the measurement of different types of RT was not due only to poor apparatus (from a technical point of view). **Particularities** of the human nervous system (there are low-level examples such as, the latencies in the afferent and efferent arcs of the peripheral and median nervous system, added to the variable times necessary for the formulation of a plan, plus individual participant characteristics such as level on the axis introversion/extroversion, etc.), **make it almost impossible** to gauge accurately the moment when one starts on a task and the moment when one has found a solution. The **precise measurement** of objective time in most cognitive tasks seems to be unnecessary precision. The accepted approach which I found in the literature for cases like these is to ask at least one assistant to make the **coding in parallel**, so there can be al least two sets of measurements to average and achieve a more objective measure. Unfortunately due to the time-consuming nature of this method, I class it for now below, in the section about future developments.

One of the reasons for the partial inadequacy of MoStaCon is for sure the **novelty of the concept** of an electronic textbook (the MonDoc engine) for the participants in the experiment, and the novelty of finding express functionality in a document of any kind. This didn't happen to MoStaCon for the first time. For example, Microsoft has included a lot of functionality in all the types of Office documents through the use of Visual Basic macros; so far malicious software (e.g. viruses etc.), is the only consistent application of this functionality. The **'fear of complexity'** has driven the average user away from computers for more than 10 years now. Huge amounts of advertising money and wonderful leaps of technology were necessary to make the computer a companion accepted and not feared (even if it sometimes gets hated as a scapegoat for the lack of user knowledge). These feelings tend to be smoothed by the increased 'friendliness' and 'appeal' displayed by recent interfaces and by the advantages brought by increased complexity.

Many reasons which the audit trails revealed, both in the pilot study and the main test, can be summed up as **'product immaturity'**. One example is the number of ideas per page. The principle of 7±2 derived by research on other types of stimuli (individual words, pictures, numbers, etc.), didn't seem to work with phrases, in the implementation of MoStaCon; some participants reported as 'too much information at the same time', which suggests an even lower number of ideas per page; the upper limit is not comfortable.

That is, even if the principles on which they operate are mature (the oldest are as old as humanity and the newest, hypertext proved its maturity through engendering one of the most booming industries ever – the Web), both MonDoc and MoStaCon are very young (several months old.) They still need to gather muscle (content) and brains (additional functionality and refining of the existing functions), until they can be used in everyday research. See the section on further developments (below) for a list of planned implementation features.

A **methodological point** on which I am not satisfied is that due to time constraints, I coded the protocols myself. Though I tried to be precise, and, when in doubt about the length of an interval I opted for the version against my theory (e.g. I would choose the shorter amount of time in the control condition and the longer one in the test condition), I may have introduced unconscious bias toward my hypothesis. Additional verification of my protocol-coding is one of the first additional analyses I would run.

An important **theoretical point** against the solutions MoStaCon suggests is that the digital medium has some problems which can't be easily overlooked. It adds increased data volatility and the interposition of layers of hardware and software between the user and the information. There are ways of dealing with volatility (backups and hard copies), and nowadays user interfaces are sufficiently transparent for the average user, and getting more transparent every year. What remains is that it's much very difficult to teach an adult to use a tool. This later problem may be solved in one generation, and by then computer interfaces might get as user-friendly as pen and paper are. It may be wishful thinking, but I trust that the MonDoc principles exposed in this paper are a small step in that direction.

The observed lack of need for MoStaCon in the high age group, given both by subjective and objective measures, may be attributed to a **high** existing **level of experience** in senior researchers, to their dependence on assistants for the more time-consuming steps of research, and especially to the fact that in that age group I had only one participant in this experiment.

Leaving aside the problems exposed until now, the majority of the participants (6 out of 8), was positively impressed by MoStaCon and heartily encouraged me to continue developing it, just like the participants in the survey (7 out of 8), If we don't consider the social constraints (e.g. the participants trying to be polite), which may have prompted these comments, they suggest that MoStaCon belongs to a class of software for which **there is a real need** in the research community.

Another global observation was that some participants tended to 'take over' the experiment, asking detailed questions, worrying about the experimental conditions, explaining in detail the theories they were mentioning, all the while trying to speed up the procedure, at the expense of their reading accuracy. On the other hand, the participants whom I consider good researchers concentrated on reading (both the tutorial and the 'literature') and applied strategies they had used in their own research, without directly referring to the previous research itself.

Since the need for a tool of MoStaCon's class **has been established**, from the qualitative data and from some of the quantitative results of this study, further research is suggested, both in the **development** of the tool (MoStaCon) and in the empirical **verification** of the rest of the principles behind it.

Further developments

The current study left some conditions uncontrolled and some questions unanswered. In order to increase internal validity, at least one other person has to go through the protocols and **generate an independent code set** to be compared and averaged with the code set I made. Additional analysis has to be done on **the effect of tiredness** on the performance, if I can find an analysis that would allow the use of an unbalanced three-level variable; otherwise, this variable will enter the design described below.

Except for these, I would like to perform **at least three more tests** in order to make the results I present here more specific. First, if I have the time and find another sample as knowledgeable, charming and helpful as the participants in this experiment, I might develop the 8 versions of the MOnitor necessary for a full Anova design, as presented in Table 2. That would not only allow me to verify all the interactions between the effects of the three principles on participant performance, but will also provide enough data for a multiple regression analysis. This result might describe better, and in a more concise way the strength of each individual effect over each of the two performance criteria I used (speed and accuracy).

Second, I might test MoStaCon on a larger participant base, during a larger period, in its complete form (without the control constraints which hindered performance), in order to assess whether **subjective appreciation** of the program increases and in order to have enough data for plotting a **learning curve** for it. Of course, during such a test, the adaptability of the system, as well as other features, may improve due to constant feedback from the participants. The perfect environment for such a test is, of course the classroom. I look forward to the teaching assistantship I will assume this fall (September, 2000), which, with the approval of the teaching professor and of my Ph.D. advisor, might provide a large (but unfortunately biased), participant pool.

The third test involves two variables (participant characteristics) which I could not control in my current test: **area of specialization and native language**. In my search through HCI papers I didn't find any international project directed at the compilation of a more inter-culturally-valid user model. If the first test suggests an increase in the subjective appreciation, as the current results seem to indicate, and the learning curve happens to be steeper than what the current survey data suggest, I might attempt to find students from other universities interested in a collaborative experiment around a user model which might contain the two 'background conditions' mentioned.

As for the interface itself, there are many ways to improve it. Examples of functionality which waits to be added are: increased **adaptive navigation support** (history-based link tracking adaptive navigation, informing the user the degree in which specific topics were read)**, adaptive presentation** (if the user decides to use the system as a management tool, only a minimum of necessary paths will be open, in order to reduce 'clicking through' for experts; if 'tutorial mode' is chosen, though, all the paths should be available), an **outline system** for the presentation editor module (because the outline system implemented in MSWord is too rigid). As for refining existing functions, it was obvious that the participants expected the still not functional **expansion module** (to allow them to modify the interface without leaving it and without touching the disgusting bowels of the system – the source code), **positional pop-up** for the notes (the notes should appear right under the text which opens them, not stacked at the top of the page, to reduce eye strain), more **detailed helpers** (in StaCon and in the editor, the dialogues which pop-up should be more automated: the user should click more and type less – especially if any strict syntax is involved). It also became necessary to implement **HTML frames usage** in order to keep the context and navigation visible at all times (the header and footer of each page) and as a way to deal with **window drift** (Windows Explorer tends to open each window 16 pixels lower and to the right of the previous one, which is, to put it in a lay phrase 'pretty darn annoying').Another remark offered by one of the participants during her debriefing session I can rephrase as **adaptability to the personality** of the participant. (the system should have at least two levels, one containing declarative knowledge and the other containing procedural knowledge – as mentioned before, in the form of animations)

I might also try to organize more strictly the material presented in the tutorial in order to reduce the **information load** per page and per note, as analysis on the prototype indicate a very large average for information load (on average 18.33 ideas/page on the pages used for the emphasis test), and participant reports (and performance) show that they were sometimes overwhelmed. As far as organization goes, most participants preferred a sparse, outline-type of presentation, with only one(!) idea per page (or note) to any other number. Their performance shows a very fast reading of pages which were structured as index for other pages, with good recall of the information presented there, and exactly the opposite (slow reading, low recall rate and even exhibited frustration) on pages which follow the linear style used in textbooks and articles.

Finally, I will say a couple of words about two other projects of mine which I develop in order to help **cope with the large amounts of data** like the ones involved in the usability study I just presented. For experiments like the one just presented, I would prefer to use an application which would transparently allow on-line experiments, automating stimulus presentation and response collection, instead of the simple apparatus I used. Existing tools, like **PsyScope** and even **ePrime** from Carnegie-Mellon University offer very good precision for the recorded time measures, but are limited to laboratory settings. What I'm targeting with two other projects of mine (MonEx and WARPe), are two alternative situations: **MonEx** is a tool planned for creating experiments accessible through the Web; **WARPe** is a tool planned for collecting large amounts of data in ecological settings again, on the Web, while participants interact within a virtual, but consistently defined reality. Though these tools don't have the time-recording accuracy of the CMU tools, they may allow a better recording for data coming out of experiments on more complex processes, like problem-solving or decision-making. Plus, with their Web orientation, MonEx and WARPe experiments may run concurrently at several locations, lowering the adaptation or localization costs of international, intercultural studies.

General Conclusion

The goal of this study was to verify whether the use of the MonDoc principles and their implementation in MoStaCon have a positive influence on the accuracy and the speed of empirical research.

As was already stated in the discussion of the experiment, both qualitative and quantitative data, subjective as well as objective, support the thesis of this paper and suggest that a tool of the class of MoStaCon **would be well received** at least by the low and middle age groups of researchers. Even with small-n data, MoStaCon was shown to **significantly reduce the time of analysis** and **increase the accuracy** of the research process. I feel confident that in the cases where it was not very clear, significance would be increased by reducing confounding sources of variability through a design closer to the ideal I had planned (2 researchers per gender, per native language, per age group, per condition; 2 x 2 x 6? x 3 x 8 = 576) I base this assumption on the trends obtained in the three experiments presented here, and on the previous research done on other similar projects, like InterBook (described in the Introduction), or other ones (see Eklund, J. et. al. (1997)). Another source of lost significance was the fact that the usability study was **done on a prototype** which had implemented only a few of the planned features.

I note the surprising result of the overriding effect of emphasis in reducing sheer perceptual overload much more than higher types of overload (reading time per character rather than per idea).

While designing and conducting this study I was faced with all of the technical problems that I discussed in the Introduction. I definitely hope that by improving different versions of MoStaCon, and by continuing development on my other projects (MonEx and WARPe), I will have less technical problems in my future research.

7. Glossary

The following pages contain specific explanations of:

- acronyms
- terms used throughout this document;
- their meaning, as I understand and use it,
- details of a specialized meaning in a more obscure field (like programming :),

all of which didn't find their way into the Literature (Chapter 4), but are used in the paper.

activity
: A goal-directed set of tasks; also the top of the hierarchy described by activity theory in HCI; for example, cooking a meal, washing, writing a paper, etc.

AnalysisPack
: a module of Excel which acts as a sort of Statistics wizard and tutorial.

bastard effect
: During the Middle Ages, in-breeding was a serious problem for aristocracies; marrying into royalty from other countries just postponed the problems of gene-pool decay. The illegitimate children were very often more able, stronger, smarter and more driven than their legal half-siblings. If we apply this principle to programs as a population - including programming languages - and considering the features pool as a gene pool, we get the conclusion, observed in perl, that creating new languages by borrowing good features from several existing ones (especially by people who know these 'parent languages' well), increases the usability and performance of the children languages more than if one decides to develop a totally new language for a specific purpose.

beginner
: throughout this experiment I refer as beginners to the participants who are studying in a cognitive field and have designed and coordinated at least two experiments of their own

clients
: in virtual communications terminology, these are the hardware and software on a network used to reach a specific service provided by some servers on the same network.

data
: individual facts, statistics, or items of information; it is a plural of DATUM, which is originally a Latin noun meaning "something given.'

(D)HTML
: Dynamic HyperText Markup Language. The text- and layout-descriptive language pioneered by the National Center for Supercomputing Applications (NCSA) for sharing documentation. In less than 10 years since then, it became the basis for a worldwide communications medium (the WWW or Web)

experiment
: a test, trial, or tentative procedure; an act or operation for the purpose of discovering something unknown or of testing a principle, supposition, hypothesis etc.

experiment log
: file used by MoStaCon to keep track of all design decisions and all notes the researcher makes for one experiment set.

expert
: throughout this experiment I refer as experts to the participants who design and coordinate experiments on cognitive topics routinely in their -scientific- activity

free tools
: A list of most of the (partially) free software I use can be found at http://cogs.nbu.acad.bg/cogs/personal/radu/proj/share.html

GIGO
: Garbage In, Garbage Out: principle in programming which states that wrong input data generates wrong outputs.

heuristic
: a guideline, general rule, simplification, educated guess, strategy or simply a shortcut that can be applied to a problem and that usually – but not always – yields a correct solution. In domains which I difficult or poorly understood, people use heuristics to reduce complex activity to simpler cognitive operations.

HYMNS
: a project of Radu Luchianov's: a Web area editor built on Object-Oriented DataBase principles, it was the predecessor of MonDoc.

interface
: a common boundary or interconnection between systems, equipment, concepts, or human beings

Internet
: One of those 'things' with dual nature, the Internet is on one hand a network of communication hardware made out of computers, cables, automated switches, and other paraphernalia. On the other hand, the Internet is a set of communication protocols which run on top of TCP/IP, the Internet Protocol/Transfer Control Protocol. As inherited from communication theory, Internet uses servers to provide information and clients to use it.

Kai Krause
: creator and president of MetaCreations, Inc.; he has pushed forward the development of the digital design industry, first by creating Photoshop filters and teaching on Compuserve and for free medium to advanced numerical techniques of image processing, then by creating a very successful line of innovative image editors (KPT, Convolver, Painter, Bryce, Goo)

markup language
: descriptive language characterized by enclosing the described object (usually text), in recursive sets of markers which contain specific description code. HTML is such an example.

MOnitor
: MoStaCon's shell. It is a decision-making graph ; it can be used by beginners as a primer in Experimental Design, and is used by beginners and experts alike as an archiving system, since it tracks every decision made while designing, writing the PsyScope script (or calls MonEx - planned feature), running and interpreting results of an experiment set, and generates a customizable report containing sources (references), data files listings, ideas noted along the way, intermediate and final results, step-by-step suggestions for the archiving of all the materials, and help with writing the report (paper, presentation, poster, chapter, book, etc.)

MonDoc
: another project of Radu Luchianov's: a cognitively accessible editor for cognitively accessible documents implemented entirely in DHTML and used to write MoStaCon's MOnitor.

MonEx
: another project of Radu Luchianov's: an ecological, MUD-based experiment design tool.

protocols
: strictly documented, algorithmic ways of exchanging data or executing actions, in the human society as well as in virtual communications

scripting language
: a rather free-syntax, interpreted programming language characterized by the direct usage of the

source code (compilation is hidden from the programmer), in order to speed-up development.

servers
: in virtual communications terminology, these are the hardware and software which provide a specific service on a network.

software
: the programs used to direct the operation of a computer, as well as documentation giving instructions on how to use them. Sometimes, data is also referred to as software.

StaCon
: The module called by the MOnitor when data conversion and/or statistical analysis is needed. StaCon's user interface is also written in DHTML; it helps the user (no previous programming knowledge required) to write a perl script (totally readable in plain English) which can then be ran on a reduced port of PERL on a Mac or IBM-compatible computer (and with few modifications to the system-sensible part of the StaCon library, on any platform in existence). This script uses the formatting and analysis modules (functions and variables) from the StaCon.pl library (another perl script, longer and not as easy to read :).

statistic
: a numerical fact or datum, esp. one computed from a sample; mainly used as a summary of large quantities of data.

system
: very basic term: an assemblage (arbitrary, natural or human-made), or a combination of things or parts forming a complex or unitary whole with specific interactions between its parts; being a part in itself, it is used as the reductionist concept used in scientific analysis

Web
: World-Wide Web, the huge, global, multimedia hypertext document distributed through the most popular Internet protocol, the hypertext transport protocol (http:). It contains virtual versions of magazines, books, company and private publications; it is free for reading by anyone who has a personal computer (or specialized equipment) connected to the Internet; everyone who can read it can also add to it their own publications if they configure their machine as a WWW server.

8. References

ActiveState, (2000), Downloading perl, on-line at: http://www.perl.com/download/

Arnheim, R. (1986), The two faces of Gestalt psychology, American Psychologist, 41, pp.820-824.

Barr, A. & Feigenbaum, E.A. (1981), The Handbook of Artificial Intelligence, vol.1, HeurisTech Press, Stanford, CA, USA. ISBN 0-86576-005-5

Blackwell, A. (1999), Notes for a course in Advanced Graphics and HCI, Cambridge computer lab, http://www.cl.cam.ac.uk/Teaching/1999/AGraphHCI/HCI/

Brusilovsky, P., Eklund, J. (1999) A Study of User Model Based Link Annotation in Educational Hypermedia, Institute for Information Processing and Computer-related new Media, issue 4, vol.4, on-line at: http://www.iicm.edu/jucs_4_4/a_study_of_user/paper.html

Card, Moran & Newell (1980), The keystroke-level model for user performance time with interactive systems, in Communications of the ACM 23(7), pp.396-410

CoolEdit excellent editor for digital sound; on-line at: http://www.syntrillium.com

Eklund, J., Brusilovsky, P. & Schwarz, E. (1997), Adaptive Textbooks on the World Wide Web, presented at the Third Australian World Wide Web Conference, 5-9 July 1997, Southern Cross University, Australia; on-line at: http://ausweb.scu.edu.au/proceedings/eklund/paper.html

Elmes, D.G. et. al. (1992), Research Methods in Psychology, 4th edition, West Publishing Company, MN, USA. ISBN 0-314-92999-1

Excel the spreadsheet editor part of Microsoft Office; on-line at: http://www.microsoft.com/office/

Goodglass, H. (1980), Disorders of naming following brain injury. American Scientist, 68, pp.647-655

Gurevich, N. (2000), Psychological elements for the World Wide Web, FREE course at the ZDNet SmartPlanet, on-line at: http://smartplanet.zdnet.com/

Hadjiilieva, K., Kokinov, B., Yoveva, M. (1997). The influence of pragmatic distance on context effects. In B. Kokinov (Ed.), Perspectives on cognitive science, Vol. 3. Sofia: NBU Press.

Helson, H. (1964), Adaptation-level theory, Harper Publishers, New York, NY, USA

Hendricks, B. et. al. (1990), The dimensions of psychological research, Teaching of Psychology 17(2), pp.266-274.

Kaptelinin, V. and Bonnie, A. (1997), Activity theory: Basic concepts and applications, Proceedings of CHI97 annual conference of the Association for Computing Machinery. on-line at: http://www.acm.org/sigchi/chi97/proceedings/tutorial/bn.htm

Kemp, S. (1987), Gestalt grouping effects in locating past events in timelines. Acta Psychologica, 64(2), 139-149.

Kokinov, B. (1997). A dynamic theory of implicit context. In Proceedings of the Second European Conference on Cognitive Science. Univ. of Manchester Press.

Lewis & Polson (1995), Cognitive walkthrough, ACM archive, on-line at: http://www.acm.org/sigchi/chi95/Electronic/documents/tutors/jr_bdy.htm

Lindsay, P., Norman, D (1977) Human Information Processing: An introduction to psychology. Academic Press, New York, NY, USA

Luchianov, R. (1995) Ideas are born Free, on-line at: http://www.nbu.bg/cogs/personal/radu/pas/freespee.html

Luchianov, R. (1999) About MonDoc, on-line at: http://www.nbu.bg/cogs/personal/radu/proj/mondoc.html

Marks, M. (1999). A test of the impact of collective efficacy in Routine and Novel performance environments. In Human Performance 12(3/4), pp.295-309, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA. ISSN 0895-9285

MathCAD mathematical formulae editor with powerful graphing capabilities; on-line at: http://www.mathcad.com/

Matlab mathematics package which tries to implement every possible mathematical formula (web address)

McConnell, J.V. & Philipchalk, R. (1992), Understanding Human Behavior, 7th edition, Holt, Rinehart and Winston Inc., FL, USA. ISBN 0-03-055747-X

Miller, G. A. (1956), The Psychological Review, vol.67, pp.81-97, on-line at: http://www.well.com/user/smalin/miller.html

MP3 Producer digital sound compressor; on-line at: http://www.opticom.de/

Murdock, B.B. (1962), The serial position effect of free recall, Journal of Experimental Psychology, 62, pp.618-625.

Nielsen, J. (1994) Heuristic evaluation, on-line at http://www.useit.com/papers/heuristic/heuristic_evaluation.html

NoteTab Light excellent text editor and system automation tool; on-line at: http://www.notetab.com/

Peale, V. (1994), La pensée positive – ed. 12, Les Editions Un monde différent, Itée, Quebec, CA.

Random House, (1996), Webster's Unabridged Dictionary, Novell, Inc., New York, NY, USA. CD version.

Rodgers, J.E. (1982), The malleable memory of eyewitnesses, Science 82, 3(5), pp.32-35.

Scarr, S. & Zanden, J.V. (1987), Understanding psychology, Random House, New York, NY, USA

Schacter & Worling, (1985), Attribute information and the feeling-of-knowing. Canadian Journal of Psychology, 10, pp.379-399.

Statistica statistical package with very good graphing capability (web address)

SyStat the most complete statistical package I know of; on-line at: http://www.statsoft.com/

WinAmp, a widely used MP3 player; on-line at: http://www.winamp.com/

ZDNet, (2000), Is Microsoft's Clippy a security Nightmare? on-line at:
www.zdnet.com/zdhelp/stories/main/0,5594,2570727,00.html

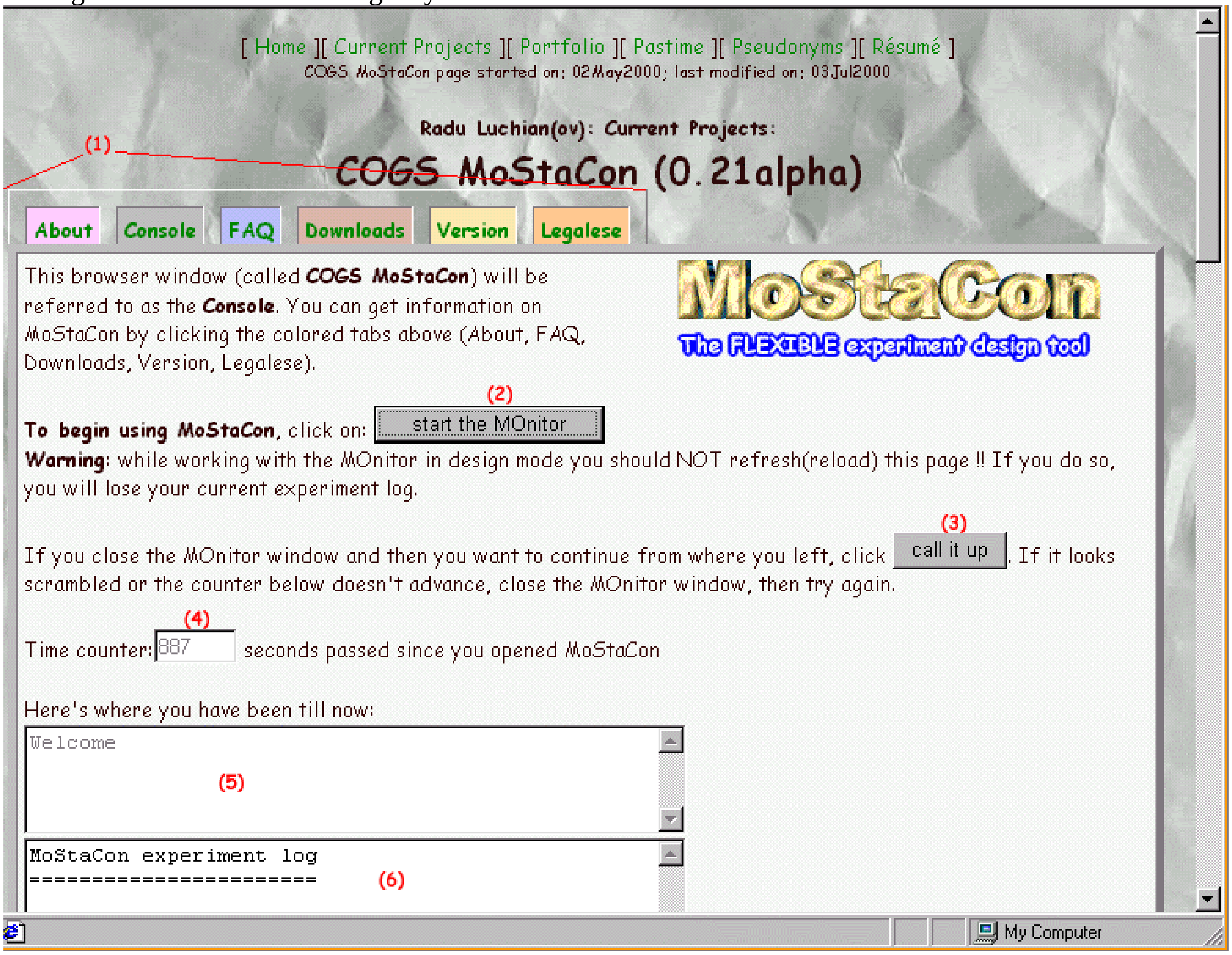
Zhang, Z. (1998), Usability engineering: usability evaluation methods, on-line at:
http://www.cs.umd.edu/~zzj/UsabilityHome.html

Zlate, M. (1999), Psihologia mecanismelor cognitive, Editura Polirom, Iasi, Romania. ISBN 973-683-278-3

Appendix 1

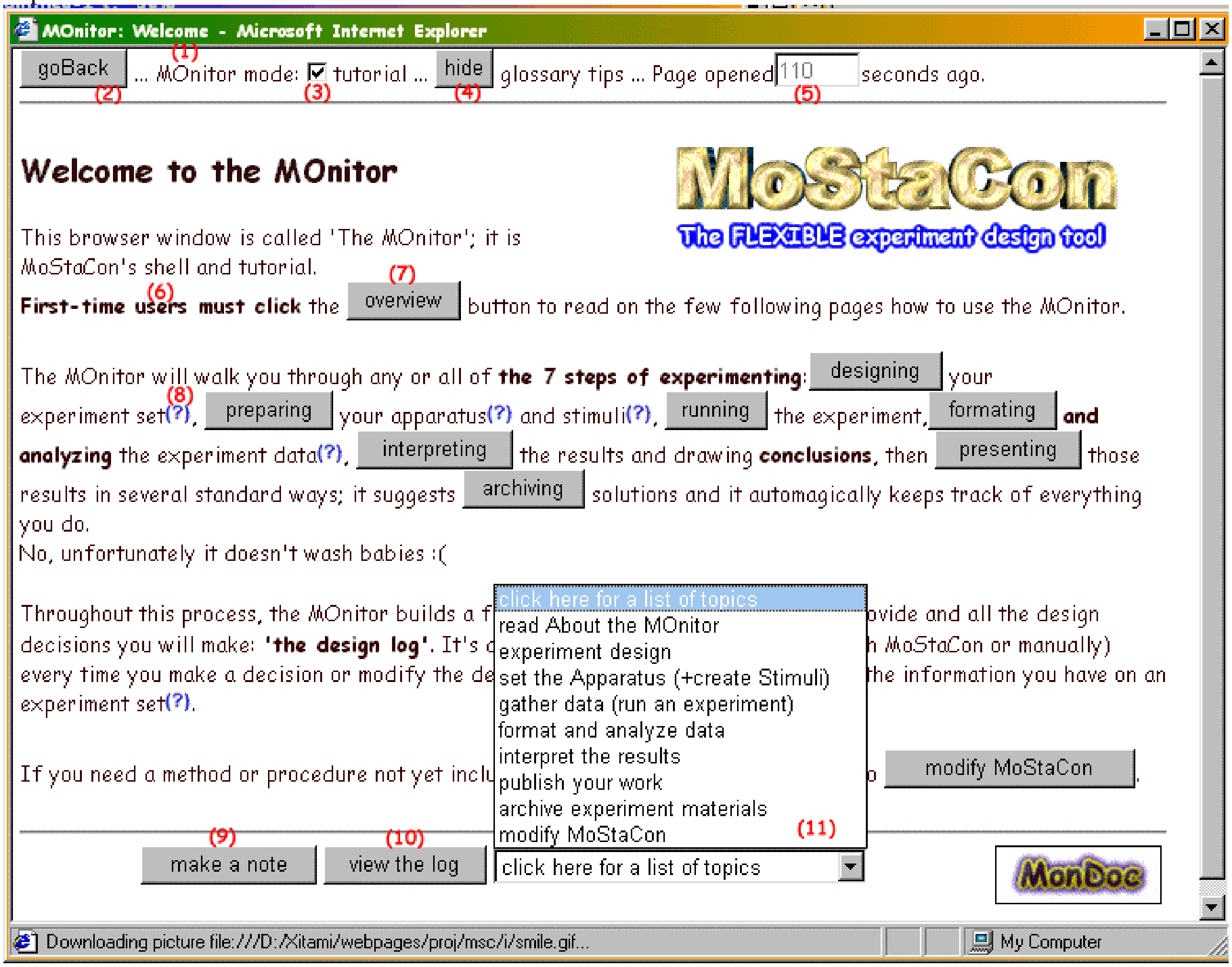
Description of the **prototype** used in this study.

MoStaCon starts up as a local or remote web page with the Console. This is the engine of the system, may contain the database of text parts, or it may use an external file containing one. The current experiment log and the path of the current user are volatile items stored in this window. That means a browser refresh, or closing the browser before saving may cause data loss.



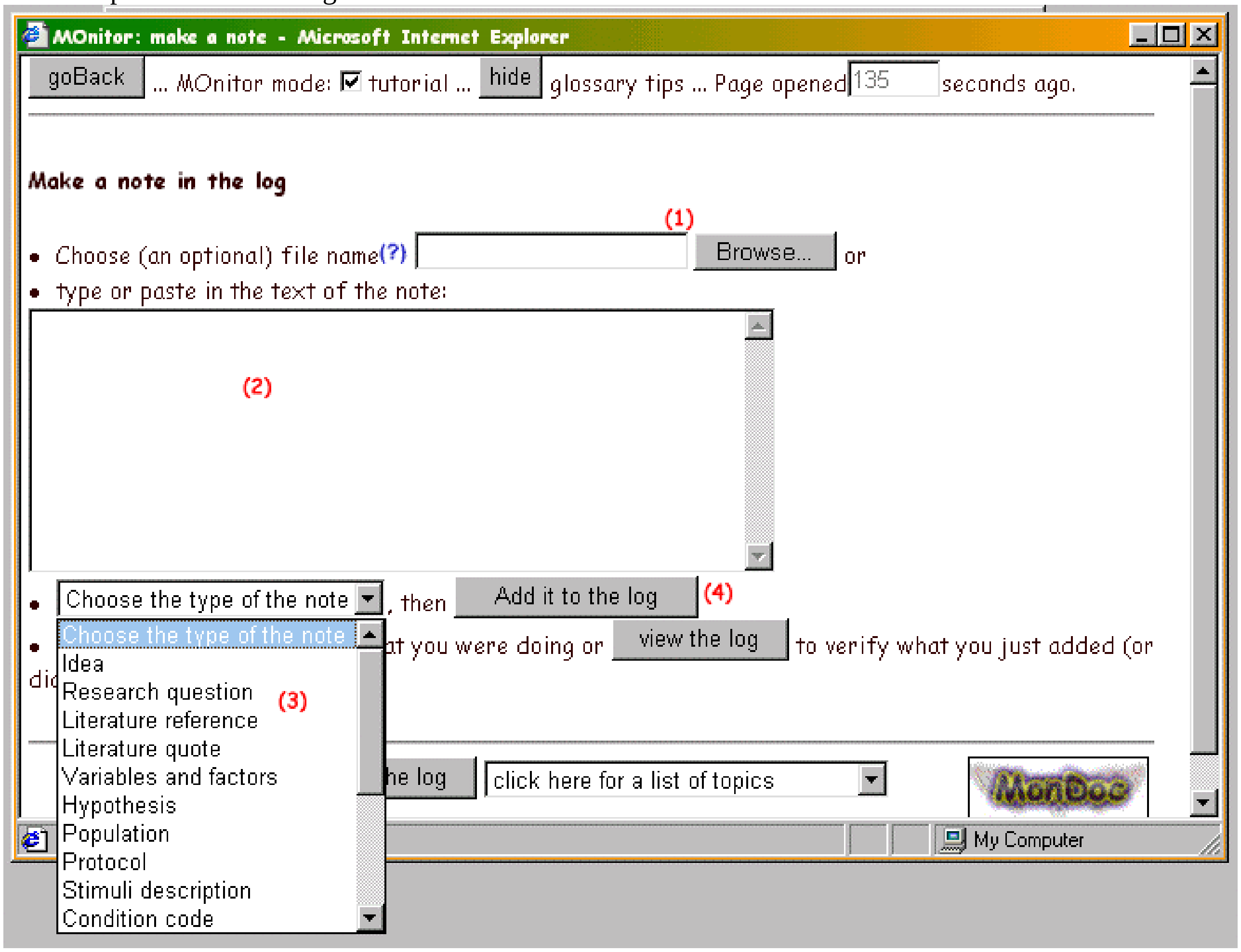
(1) tabs which can be used to access different types of information on MoStaCon without leaving the page
(2) the button used to start the MOnitor
(3) the button used to bring the MOnitor in focus in case it gets covered by other windows
(4) a seconds counter for the current session
(5) the list of locations in the order the current user has visited them (history list) – not editable
(6) the experiment log for the current experiment set – editable

The MOnitor is the main working place for the learning about experiment design and for keeping track of experiment notes.



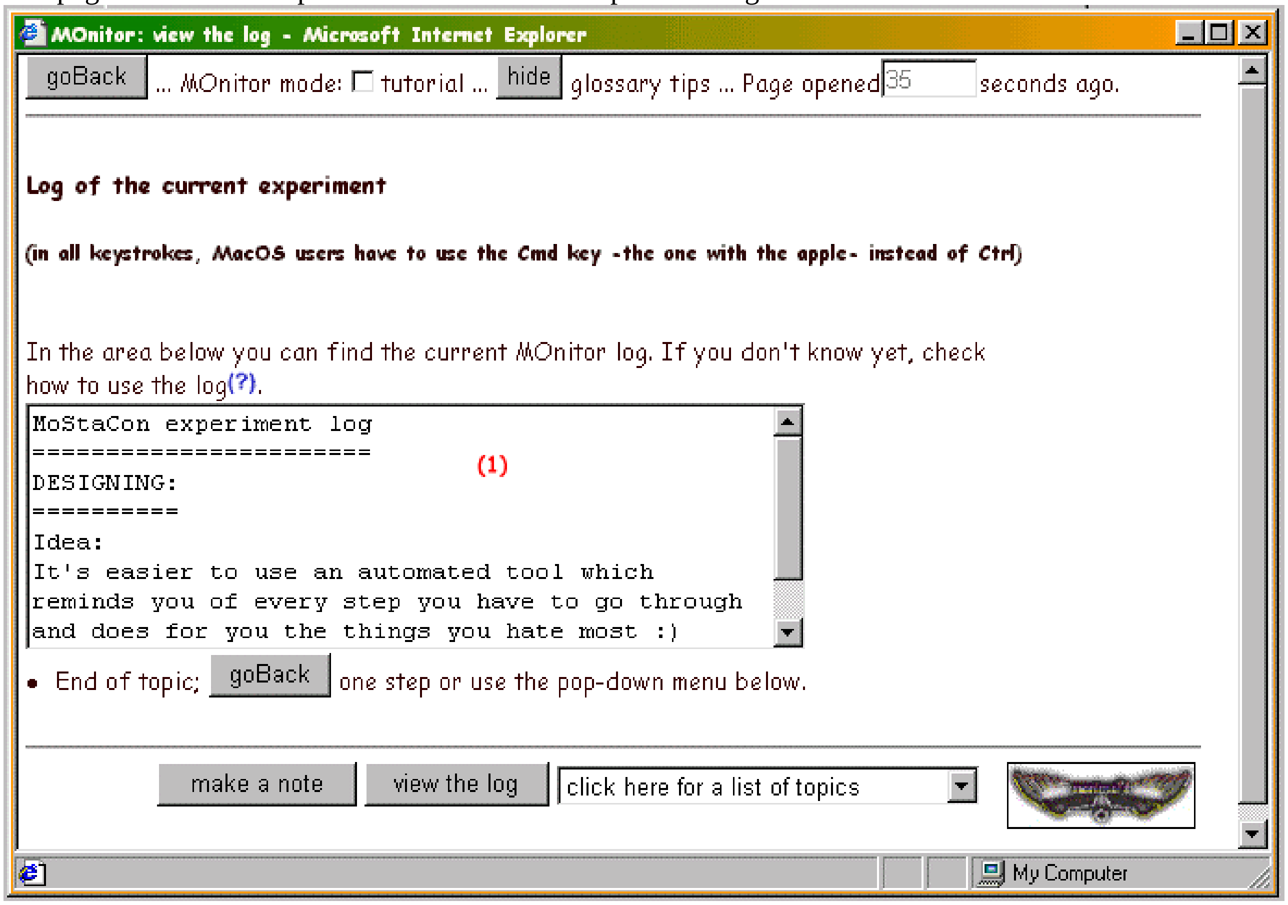
(1) title bar – it always displays the name of the current page;
(2) the 'Back' button, the only element preserved from the set of toolbars normally employed by the browser;
(3) the tutorial check-box. Used to switch between Design mode and Tutorial mode;
(4) the Glossary button – if the blue marks annoy you, and you're an expert, click this button to hide all notes and references;
(5) the local seconds counter for the page;
(6) emphasized text;
(7) path through the information space, marked by a button;
(8) the blue question mark – current note marker;
(9) opens the expert's interface, the notes page;
(10) opens the experiment log for viewing, editing and file activities (load and save);
(11) navigation menu, gives access to all 7 top-level tutorial topics;
Note: the header and the footer are the same on every MOnitor page.

The expert's interface, allows to browse the local computer for files, to add notes, and it provides an editable list of empirical research stages and research-related information.

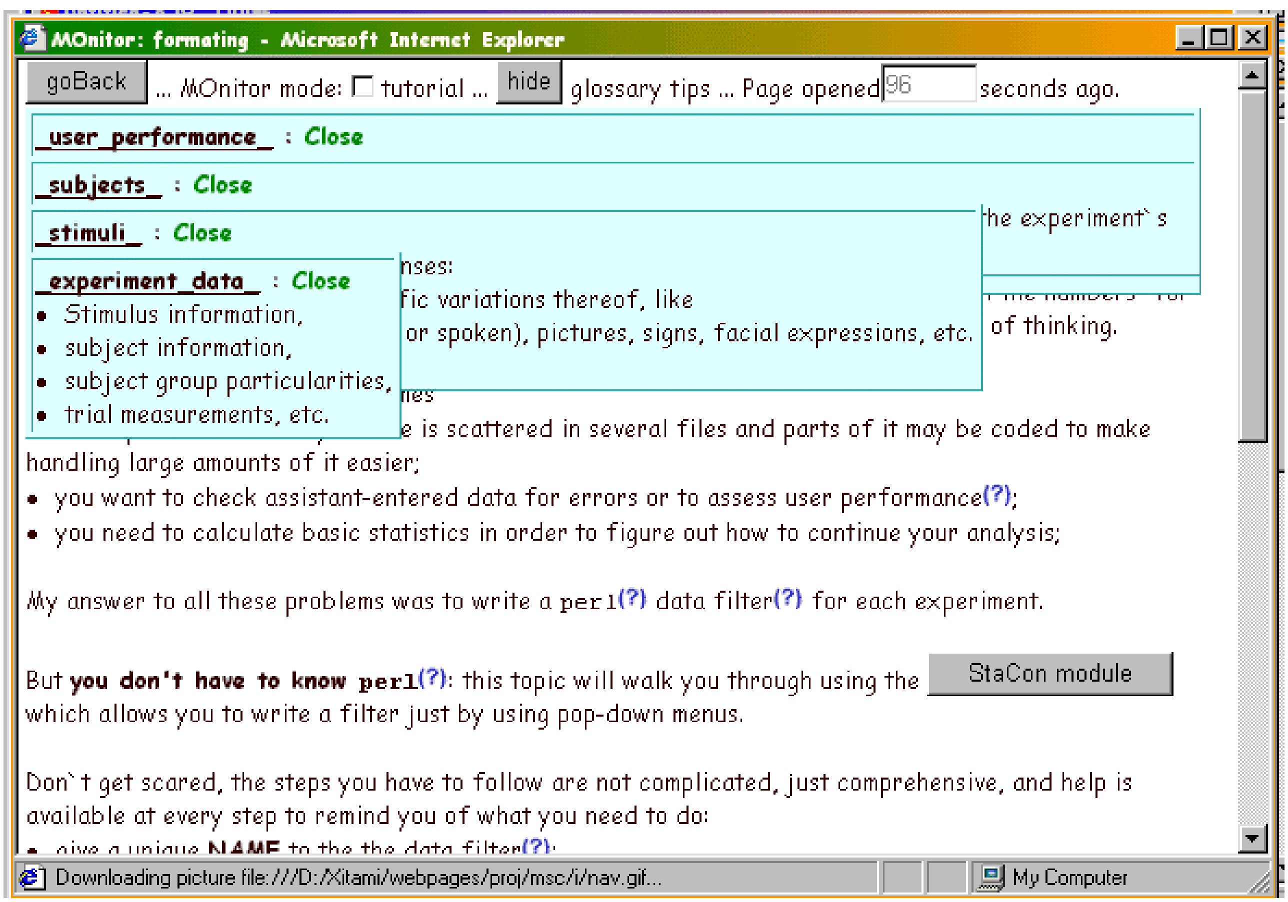


(1) file locator; use it so you don't have to type manually the names and locations of files;
(2) note area, where the user types or pastes in the text of the note;
(3) editable notes menu;
(4) button for adding the current note to the current experiment log.

The page which allows operations on the current experiment log.

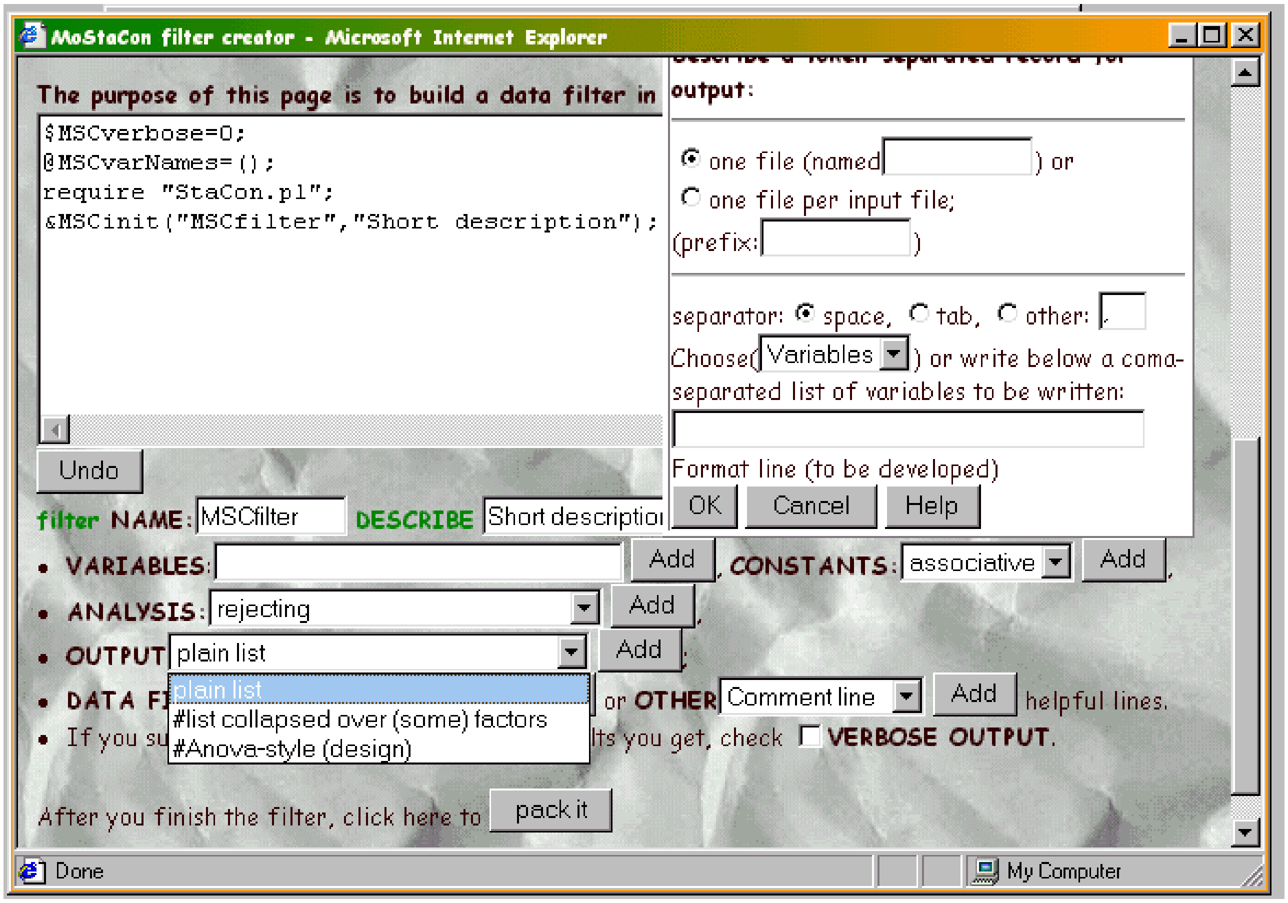


(1) when in design mode, all decisions you make are recorded in this plain text file.



An example of page (data formatting), with some opened glossary entries.

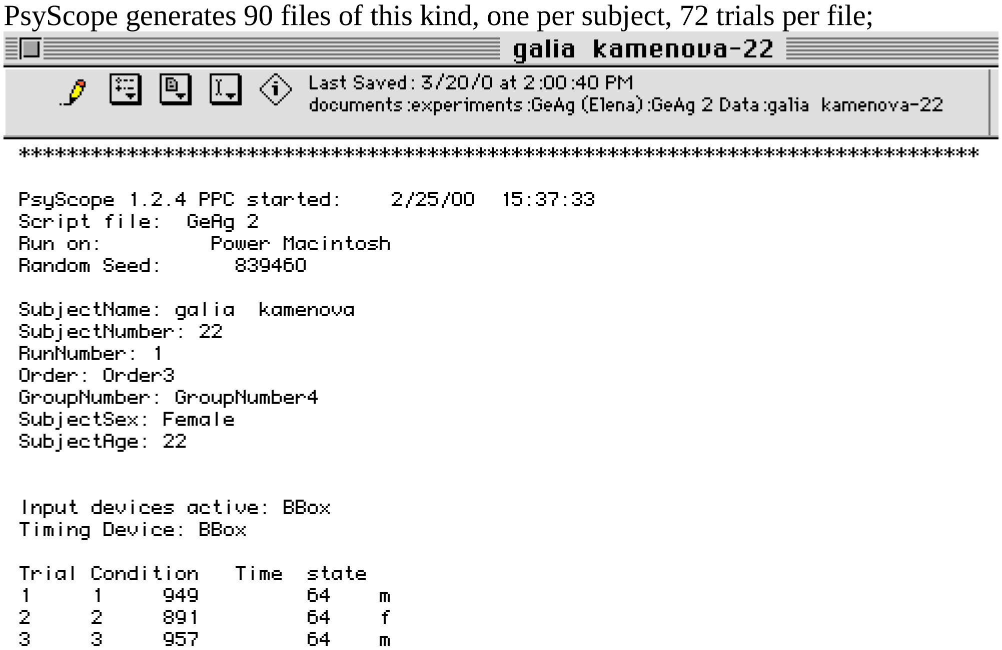
The interface of StaCon



Allows visual programming of `perl`, then 'packs' the script for Dos/Windows, MacOS or UNIX.

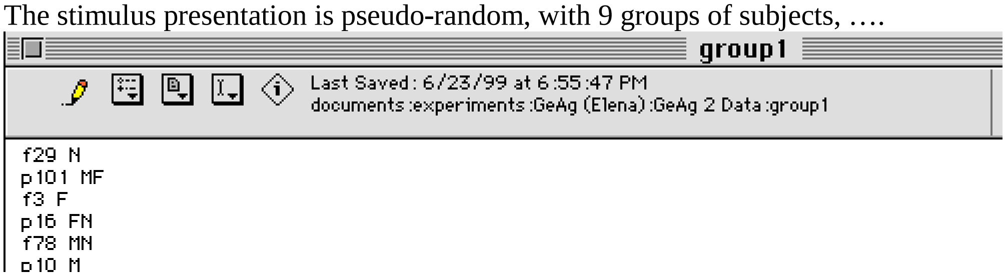
Take the following experiment example.

PsyScope generates 90 files of this kind, one per subject, 72 trials per file;



```
galia kamenova-22
Last Saved: 3/20/0 at 2:00:40 PM
documents:experiments:GeAg (Elena):GeAg 2 Data:galia kamenova-22
********************************************************************************

PsyScope 1.2.4 PPC started:    2/25/00   15:37:33
Script file:   GeAg 2
Run on:          Power Macintosh
Random Seed:       839460

SubjectName: galia  kamenova
SubjectNumber: 22
RunNumber: 1
Order: Order3
GroupNumber: GroupNumber4
SubjectSex: Female
SubjectAge: 22


Input devices active: BBox
Timing Device: BBox

Trial Condition    Time    state
1      1      949          64      m
2      2      891          64      f
3      3      957          64      m
```

The stimulus presentation is pseudo-random, with 9 groups of subjects, ….



```
group1
Last Saved: 6/23/99 at 6:55:47 PM
documents:experiments:GeAg (Elena):GeAg 2 Data:group1

f29 N
p101 MF
f3 F
p16 FN
f78 MN
p10 M
```

…. and 10 pre-randomized orders of stimuli.

order1

Last Saved: 6/23/99 at 6:55:50 PM
documents:experiments:GeAg (Elena):GeAg 2 Data:order1

```
106 f3  автомобил   кола     комби
107 p16 писмо   плик     телеграма
108 f78 елка    калкулатор  машина
109 p10 завеса  перде    сцена
110 f64 накит   колие    огърлица
111 p1  легло   спалня   креват
112 f41 торта   парче    сладкиш
```

Until now we have 109 files to be processed. Add one more in order to simplify the process of reading subject files (a list of the 90 subject file-names)

subjects GeAg 2 names

Last Saved: 4/10/0 at 5:57:56 PM
documents:experiments:GeAg (Elena):GeAg 2 Data:subjects GeAg 2 names

```
anita  dobreva-84
anna  krasteva-61
anna-marija peeva-65
antonija  geneva-33
asan  molla-20
atanas  popov-77
bogidar  hinkov-38
boriana  rashkova-51
```

Imagine the number of hours necessary for assistants to arrange this data set into usable tables. And the errors involved. And the fact that several times during analysis, we remember to add some additional variable.

With StaCon, after 1h visual programming and 5-6 h debugging the logic of coding and decoding (say one day of work), we get a script which is run in under one minute and generates the complete table:

GeAg all.txt

Last Saved: 4/10/0 at 7:21:30 PM
documents:experiments:GeAg (Elena):GeAg 2 Data:GeAg all.txt

Subj	Sex	Age	Trial	Pict	PT	VerbG	AdjG	Prod	RT	VAgr	AAgr	SType	ERR
1	F	18	1	p58	p		M	M	910	0	1	single	
1	F	18	2	f62	f	F	M	F	1486	1	2	dbl-dif	
1	F	18	3	p83	p	M	F	F	1000	2	1	dbl-dif	
1	F	18	4	f104	f		N	F	902	0	2	single	
1	F	18	5	p92	p		M	F	1092	0	2	single	
1	F	18	6	f110	f	F	M	F	1009	1	2	dbl-dif	
1	F	18	7	p4	p		N	N	836	0	1	single	
1	F	18	8	f32	f	M	N	M	1396	1	2	dbl-dif	
1	F	18	9	p6	p		N	N	837	0	1	single	

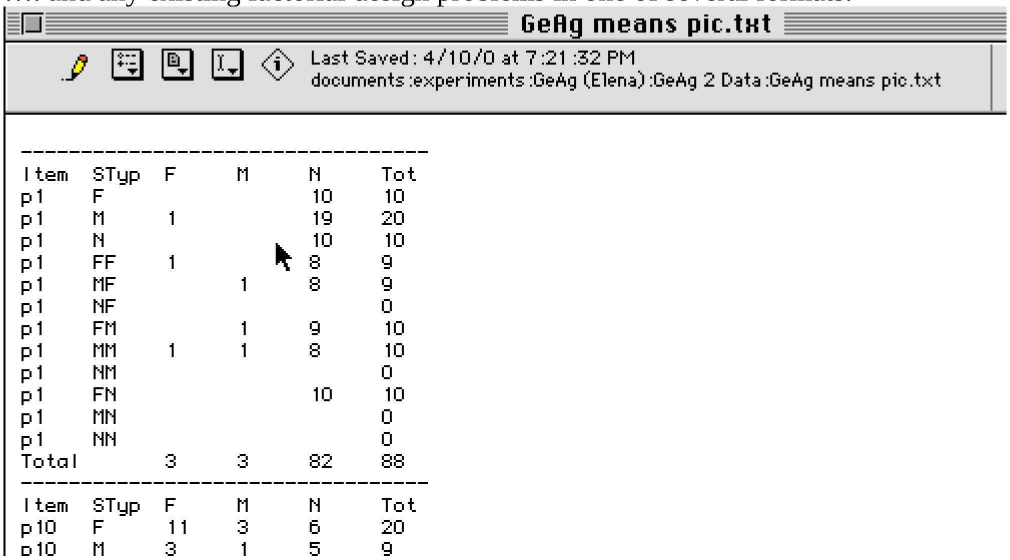
And we get a report file with technical or coding errors

GeAg errors.txt

Last Saved: 7/23/0 at 9:31:46 AM
documents:experiments:GeAg (Elena):GeAg 2 Data:GeAg errors.txt

```
Subj:82; trial:33; error:ami
Subj:82; trial:39; error:ami
Subj:83; trial:71; error:-
Subj:87; trial:9; error:ami
Subj:88; trial:23; error: no production!! (AMI)

    Summary: total 6480, from which:
 3103 valid cases (47.89%);
 3240 fillers: rejected (50.00%);
 109 cases with errors: rejected (1.68%);
 28 cases with no production: rejected (0.43%).

<End Of File>
comp-GeAg <part of StaCon v0.01> Monicsoft(c)1999
```

…. and any existing factorial design problems in one of several formats.

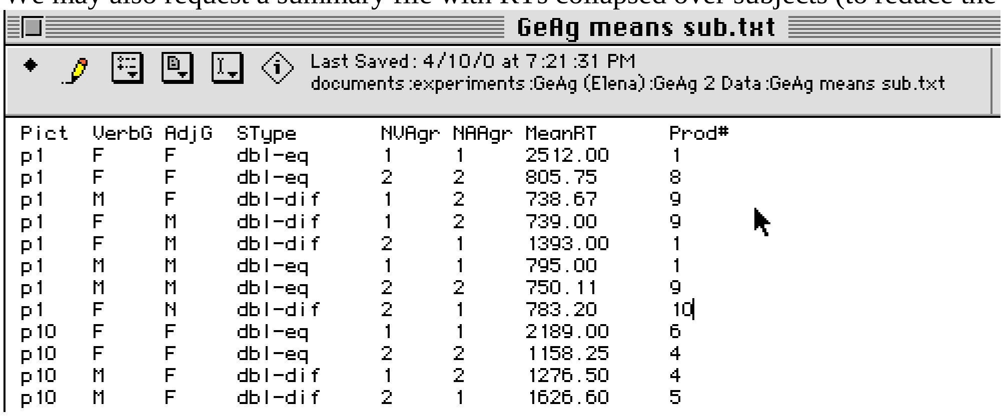

GeAg means pic.txt

Last Saved : 4/10/0 at 7:21:32 PM
documents :experiments :GeAg (Elena) :GeAg 2 Data :GeAg means pic.txt

Item	STyp	F	M	N	Tot
p1	F			10	10
p1	M	1		19	20
p1	N			10	10
p1	FF	1		8	9
p1	MF		1	8	9
p1	NF				0
p1	FM		1	9	10
p1	MM	1	1	8	10
p1	NM				0
p1	FN			10	10
p1	MN				0
p1	NN				0
Total		3	3	82	88

Item	STyp	F	M	N	Tot
p10	F	11	3	6	20
p10	M	3	1	5	9

We may also request a summary file with RTs collapsed over subjects (to reduce the degrees of freedom)


GeAg means sub.txt

Last Saved : 4/10/0 at 7:21:31 PM
documents :experiments :GeAg (Elena) :GeAg 2 Data :GeAg means sub.txt

Pict	VerbG	AdjG	SType	NVAgr	NAAgr	MeanRT	Prod#
p1	F	F	dbl-eq	1	1	2512.00	1
p1	F	F	dbl-eq	2	2	805.75	8
p1	M	F	dbl-dif	1	2	738.67	9
p1	F	M	dbl-dif	1	2	739.00	9
p1	F	M	dbl-dif	2	1	1393.00	1
p1	M	M	dbl-eq	1	1	795.00	1
p1	M	M	dbl-eq	2	2	750.11	9
p1	F	N	dbl-dif	2	1	783.20	10
p10	F	F	dbl-eq	1	1	2189.00	6
p10	F	F	dbl-eq	2	2	1158.25	4
p10	M	F	dbl-dif	1	2	1276.50	4
p10	M	F	dbl-dif	2	1	1626.60	5

Any time a new type of procedure has to be implemented in StaCon, expect a couple of days – up to a week – of programming time for someone who knows `perl` and JavaScript.

Appendix 2

Examples of the pages used in the experiment on emphasis.

Page 1, version A:

About MoStaCon

MoStaCon
The FLEXIBLE experiment design tool

Have you ever run an experiment? Any sort of experiment, something in which you **gather data** on the behavior of some system in order to **build a theory** or **check the validity** of an existing one. If you have, you must have reached a moment when you had to do something to the data which involved tedious and repetitive operations: coding, recoding, sorting, sampling, filtering, swapping variables, modifying values, etc. Most of the time some assistant (student and/or employee) winds up doing the 'dirty work'.

That's when **MoStaCon** can help. It **MOnitors** your empirical work, and when necessary, it helps you to do **STAtistics** and **data CONversion**. Imagine it as a notebook that (1) **suggests** and (2) allows you **to summarize** in a formal way **the operations** you have to perform while researching, then it (3) takes your raw data and does those operations quickly and (4) **without the error** involved in human handling. If the paradigm under which you're researching is already programmed into MoStaCon, the system (5) **walks you** through the process of designing your experiment, formatting the data, calculating statistics and it (6) **warns you** of data entry or design errors or of attempts to use wrong data types; it may even (7) answer your final experimental questions.

- Before you read any more details, you may want to start the MOnitor from the **MoStaCon** [Console] to see it at work;
- or you may want to check the feature list of the system.

Page 1, version B

About MoStaCon

MoStaCon
The FLEXIBLE experiment design tool

Have you ever run an experiment? Any sort of experiment, something in which you gather data on the behavior of some system in order to build a theory or check the validity of an existing one. If you have, you must have reached a moment when you had to do something to the data which involved tedious and repetitive operations: coding, recoding, sorting, sampling, filtering, swapping variables, modifying values, etc. Most of the time some assistant (student and/or employee) winds up doing the 'dirty work'.

That's when MoStaCon can help. It MOnitors your empirical work, and when necessary, it helps you to do STAtistics and data CONversion. Imagine it as a notebook that (1) suggests and (2) allows you to summarize in a formal way the operations you have to perform while researching, then it (3) takes your raw data and does those operations quickly and (4) without the error involved in human handling. If the paradigm under which you're researching is already programmed into MoStaCon, the system (5) walks you through the process of designing your experiment, formatting the data, calculating statistics and it (6) warns you of data entry or design errors or of attempts to use wrong data types; it may even (7) answer your final experimental questions.

- Before you read any more details, you may want to start the MOnitor from the MoStaCon Console to see it at work;
- or you may want to check the feature list of the system.

Note the (presence and lack of) emphasis and the path markers (buttons vs. links)

Page 2, version A

Designing an experiment set(?)

No experiment design happens in a vacuum. Research starts with reading literature and generates more literature in the form of the reports you write for the scientific community. You may:

- read books, journals or articles on the Web;
- participate in scientific events;
- observe yourself (introspection), the people on the street or your pet, and get an idea about which you can't find any work already published (pretty rare case... It may be that you simply didn't search well enough).

Once you get your research question (and make a note of it), you have to shape it into one or more testable hypotheses. This means you have to read some more, to determine what is already known about your hypotheses; concentrate on articles at this time, even unpublished ones, and try to contact the people you know are working in the same field, so that you can:

- reduce the amount of redundant work;
- verify each other's results and conclusions;
- coordinate or even split a larger topic;
- feel better and not so much alone in a cruel world

That exercise in reading leaves you with a documented opinion. It's time to plan the way you're going to study or verify that opinion, or you may have already adopted a paradigm as suggested by the reading.

Page 2, version B

Designing an experiment set(?)

No experiment design happens in a vacuum. Research starts with **reading literature** and generates more literature in the form of the reports you write for the scientific community. You may:

- read books, journals or articles on the Web;
- participate in scientific events;
- observe yourself (introspection), the people on the street or your pet, and get an idea about which you can't find any work already published (pretty rare case... It may be that you simply didn't search well enough).

Once you get your **research question** (and make a note of it), you have to shape it into one or more testable hypotheses. This means you have to read some more, to determine what is already known about your hypotheses; **concentrate on articles** at this time, even unpublished ones, and try to **contact the people** you know are working in the same field, so that you can:

- reduce the amount of redundant work;
- verify each other's results and conclusions;
- coordinate or even split a larger topic;
- feel better and not so much alone in a cruel world

That exercise in reading leaves you with a **documented opinion**. It's time to plan the way you're going to study or verify that opinion, or you may have already adopted a paradigm as suggested by the reading.

Appendix 3

Instructions used in the survey

(Since this experiment in Bulgaria, I used instructions in English for the participants who are fluent in this language and in Bulgarian for the others.)

[pdf files]

Instructions used in the usability study

{In order to control for the order of presentation of the test condition and the control condition, there are two versions of the instructions:

- first using MoStaCon, then Excel;
- the second using Excel and then MoStaCon}

[doc files]

Appendix 4

This is the list of empirical research steps, used in the global experiment (MoStaCon vs. standard research) Emphasized steps have been used in the Recall Time analysis. First number is the number of participants who mentioned the step in the control condition and the second number in the test condition (using MoStaCon).

8	**8**	**Read specialized literature**
8	**8**	**Find research idea**
2	4	Discuss with experts, colleagues, and/or the author of the problematic research
8	**8**	**Draft hypothesis**
8	**8**	**Isolate dependent variables**
8	**8**	**Select independent variables**
5	8	Determine control variables
2	4	If there's no available data on the amount of influence of certain control variable, run a test for it
4	6	Specify type and procedure of measurement for used variables
3	2	Check literature for similar measures and procedures (maximize construct validity)
8	**8**	**Select stimulus material**
5	8	Prepare apparatus
3	2	Draft design of analysis to be performed
4	7	Run pilot study
2	5	Record subject data for small number of [fake] subjects
2	1	Check feasibility (running time, accurate measures, etc.)
0	0	Maximize internal and external validity
8	**8**	**Perform planned analysis**
4	6	Look for confounding variables
4	6	If such are found, refine design and repeat pilot study
4	8	If not, select subjects according to hypothesis constraints
8	**8**	**Perform experiment and record subject data**
0	3	Refine subject data (filter away technical errors, reject unable subjects,)
0	1	If data set is incomplete after filtering, perform experiment on more subjects
4	4	Compare results with literature
8	**8**	**Draw conclusions**
2	2	Discuss results and conclusions with peers and/or experts
8	**8**	**Write paper**