This is the published version of a paper presented at Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Ronneby, April 23-25, 2001.

Citation for the original published paper:

Pettersson, J. S. (2001)
Simulating interactive graphic user interfaces.
In: Janis Bubenko jr (ed.), IT-lyftet 2001: Part III (pp. 55-61). The Knowledge Foundation

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:kau:diva-38769
Simulating interactive graphic user interfaces

John Sören Pettersson
Information Systems
Karlstad University
john_soren.pettersson@kau.se

Abstract. This is a report within category 4: reports with a point of departure in earlier disciplinary affiliation, which in the author's case is General Linguistics. The earlier studies on how writing was invented and how it relates to other means of expression are here extended to a characterisation of the means of expression within human-computer interaction. Special interest is paid to people with limited linguistic competence as well as the relation between linguistic and graphic means of expression. An on-going attempt to make a GUI experimental station for non-programmers, initially targeting on teachers and language therapists, is reported. The idea is to provide for manual simulations of the interactivity of proposed products, thereby simplifying testing and making possible the development of ideas during test runs.

Two forms of graphics: writing and pictures

I have dealt with Human-Computer Interaction (HCI) even before I came to Karlstad University and its department of Information Systems, teaching as I did civil engineering students in Analysis and Synthesis of Speech, that is speaking and listening computers. My main focus during my years as a research student in General Linguistics was however on writing systems (alphabetic, syllabic, logographic), especially how the historical development of writing has contributed to the development of metalinguistic concepts, concepts about language. At a deeper, philosophical level, these two areas touch on each other, because the notion of language will be crucial to the work carried out within speech technology research as well as for expectations on that work.

Also when it comes to the graphic user interface, the writing-theoretical perspective is interesting (Pettersson, 1997, 1998). As a stable or durable communicative sign, a written sign is always placed somewhere. It is convenient to speak of a 'writing-surface'. There is furthermore always a writing-tool (pen, brush, stylus, type-writer, finger) involved in the act of writing. However, as a writing-surface, the computer screen is very special. It demands constant electronic support to sustain, really re-write, the characters written on it. Of course, electronic devices can perform this tiring exercise. The writing-tool (or a part of it) is automated and integrated with the writing-surface while the text is stored electronically and is thus accessible to the integrated writing-tool. This solution defines a new kind of writing-tool, because never before did a pen have access to what was written with it, neither was the writing-tool part of a self-inscribing surface.

This integration has several implications. Here, let us just note that it allows for animation. Animation furthermore is an example of what writing-theorists have defined as not writing; the sign is not permanent but more like the ephemeral, transient signs of speech and gesture. However, in combination with the interactivity which computers allow for, the animated sign achieves some permanency when user evokes the animation again at whatever occasion they want to read it (see it).

Now, my intention is not to press the whole concept of interactive media within the frames of writing, and perhaps it is even not possible to do so. The user's eliciting of the animation could be of various sorts including unconscious eliciting of computer responses (very often so in computer games). In any case, as a General Linguist I find different forms of communication, even highly automated forms, interesting. And within
Interacting expressions

Linguistic expressions emanating from an automaton, even when it is an interactive user interface, are rather monologous. This is apparent for anyone studying young children using computers. For a somewhat older child, a figure on the screen who says something unintelligible, will remain unintelligible even if it is possible to evoke the utterance again, because computer figures simply repeat their phrases like tape-recorders. Younger children may not regard the loudspeaker voice as directed towards them at all. Obviously, the computer lacks the senses (in every respect of that word) to detect the child’s needs.

Naturally, lack of empathy holds also for the graphic part of the user interface. But while a voice interface or a text-based interface furthermore suffers from the lack of an understanding listening- or reading-module in the computer, there is for the graphics a possibility for greater automatic ‘understanding’. This is so partly because the mouse cursor has a limited range of possible physical expressions, and partly because the expressiveness is determined of what else appears on the screen.\(^1\) Because the possible expressions are limited by what appears on the screen, the computer has access to the user’s expressions. This is simply the base for direct manipulation where, for instance, one drags a document icon to the wastebasket icon (Shneiderman, 1998, Ch. 6). In the graphic user interface texts, pictures, animations and video clips can co-occur. They occupy a space on the screen and is thereby easily located and referred. They can easily as it were partake in what happens on the screen, including mouse-cursor activities and keyboard-driven text input.

Presently I am investigating how to build up a research environment which could be used for improving, actually even improvising, graphic design including the design of interactivity (this is a project within HumanIT — a research platform partly supported by the KK-Foundation). A special target group for such design consists of non-readers (or semi-literate). I have been interested in the development of reading and writing abilities in young children. At Karlstad University there is furthermore a Disability & Language group (now formally the Dept. of Special Education) working with other kinds of non-readers, even non-speakers. Lack of written language will be a common ‘dysfunction’ here. Before reporting on the ideas behind this attempt, I would like to emphasise a limitation in the concept of direct manipulation when it comes to educational software for young children but I would also like to mention a couple of problems related to written child-computer interaction.

Usability or intelligibility?

When making the human-computer interaction easier, or more user-friendly, one has to be careful not to replace what was to be taught.

To begin with, let’s look at Apple Computer’s KidSim (Smith & al., 1994). I will not say that it was a bad product, but it does not quite solve the perceived problem. It was developed as a reaction against all programming languages for non-programmers — languages which never fulfilled the expectations: “We have come to the conclusion that since all previous languages have been unsuccessful by the criterion described here, language itself is the problem. It does not matter what the syntax is. Learning another language is difficult for most people.” (Ibid.: 56) And so they made an editing program in which users can – by very concrete drag&drop demonstrations which are automatically generalised – define simulation applications and games, which then will function as stand-alone products.

In a way they solved the programming language problem although the range of applicability of KidSim is quite limited. But when the users want to go on further, making more elaborate simulations or games, they will have no programming experience at all to build on. Normally, when it comes to educational software for children, the graphic design and the interaction design are aimed at supporting the child’s acquisition of various linguistic skills. It is not bad to customise products, but in many cases, and especially when it comes to children, the goal has to be to teach what is already in existence in the ambient society, for example the syntax of English language.

Verbally or visually oriented interfaces?

Still I want to defend a rather de-verbalised interface. And I will make my points in relation to an article appearing in the Swedish daily Computer Sweden (November 8, 2000) where Agneta Guiz takes the offensive

---

\(^1\) Or at least it is easy to have the user accept this; an elaborate gesticulation of the mouse cursor simply doesn’t count. However, in draw&paint programs children can discover the effects of an exaggerated waving of the cursor and indeed, then they often lend themselves to such drawing.
against tendencies to over-emphasise visualisation in to-days teaching aids. She complains on the lack of verbalisation especially in what she calls 'mathematics multimedia' ('matematikmultimedia') mentioning a specific program as an example (Matador from Liber, 1999; produced with financial support from the KK-Foundation which might be of some relevance to this conference). She points out that the program was developed for pupils with reading difficulties, but in the very favourable reviews of this program, this limitation of the target group is not mentioned but instead it appears as if the program was aimed at every child in its target age.

A first reaction to Gulz's article would be that not all producers can be aiming at all pupils. Gulz actually says that 'verbalisers' have been favoured earlier. Then it seems to me to be arguable that we should make teaching aids for also other kinds of pupils rather than making products for every kind of pupil. But I have to admit that Gulz has a point when she claims that children's growing linguistic categorising could be restrained by totally picture-based programs. In the example she gives, linguistic categories are almost totally replaced by images. Now, this seems unnecessary monomedia in a multimedia product! However, there are two circumstances where suppression of linguistic expression should be considered.

In a program for children with reading difficulties I can see a reason not to emphasise written expressions. These pupils should have the right to feel they have a good command of the game (mathematics, in the example). They might have a low self-esteem already and it may be better to allow them to experience an unstrained success with one particular subject even if it would be desirable they had an ability to learn linguistic expressions pertaining to the subject in question (Keates, 2000: vi-vii, 7, 17, 67). Attached written designations can destroy this experience if some well-meaning classmate or teacher happens to refer to something being named on the screen. Such a reference is obvious for the literate, but can for the pupil with reading difficulties again provoke the feeling of being a failure if he does not understand. (In Gulz's example, Matador, there are actually many texts, several for each exercise; they are, however, read aloud by the program. In a reply, Pettersson 2001, I do not evaluate Matador because at that time I had not tested it myself.)

Another circumstance has more to do with systems development: I would suggest that verbally oriented user interfaces could be seductive for the software designers. Because we grown-ups already know the words, both their forms and their meaning, it is an easy mistake to do underestimate what problems children may experience with help files and instructions (even when these are read aloud by the computer). Of course, this pertains also to other teaching aids than computer programs. A de-verbalised user interface would on the other hand have to demonstrate concepts to make them intelligible. I do not speak of individual icons, since they do not partake in the screen picture — in the sense discussed earlier — except when clicked on. No, the icons may be good memory-aids when the concepts have been learnt, but as we all probably have experienced, the fact that icons constitute of small pictures does not explain their meaning, their function, if we do not have any notion of such a function beforehand.

How to construct sensible demonstrations in a visually oriented interface — it might be for language training as well as for maths — is, admittedly, no simple task. It depends very much on for what and for whom the demonstration is intended. It must also take care of children's' own attempts; it should not simply be a video film.

**Manual simulations of the graphic user interface**

As mentioned earlier, I am pursuing a project on how to make a GUI experimental station for non-programmers, targeting on teachers and language therapists initially. I am loosely collaborating with the Disability & Language group of our university. They are used to making special educational material and are in general positive to the idea of developing IT-based material. While I am interested in the pedagogical aims of my partners I think that a GUI experimental station as described below might be useful for other naive designers too. Anyhow, the question now is how to develop research facilities for the study of interactive graphic expressions as teaching aids. This is a system development that does not focus so much on end-user products as on the means for developing such products.

Lack of written language will be a common 'dysfunction' whether we speak of young children or the disabled persons in focus of my partners. The intention is that we should be able to define very graphic products even if the products are used in language training. Unfortunately, even a mock-up multimedia product will take a lot of time to set up using a prototyping tool like Visual Basic or Director (and I suppose KidSim and similar products are too simple). Testing and iterative design will be costly, especially when dealing with pupils with very special needs. And these educationalists and therapists are not skilled in programming. Rather, testing new ideas should be the ordinary run of the mill, not 4GL programming. Ease of testing will benefit the product development.
Admittedly, everything in the development process will not be solved by importing the Wizard of Oz technique into the field of graphic interaction design. Let us quickly review some obstacles.

**The Wizard of Oz technique**

There is an experimental technique often employed in language technology called Wizard of Oz experiments. In Wizard of Oz experiments a test person thinks he writes or speaks to the computer in front of him when in actual fact the test leader sits in the next room interpreting the user’s commands and providing the right responses. Inspired by this technique I see that a plain GUI with only ‘icons’ and background and drag&drop functionality could be useful if the test leader (educational researcher) could *mimic the behaviour* of the planned multimedia product. That is, the functionality apparent in the test room is a false one, no system with that functionality has been implemented. It is the test leader in the control room that controls the output of the test room computer. Thus, the control room has to be equipped with some computers connected to the computer in the test room. Mice, joysticks, keyboards, microphone and other equipment is allowed in the control room, but it must be easy for the test leader to couple them to the things appearing on the screen of the test room computer. Perhaps some functionality that is easy for non-programmers to store, like a video clip, could be within reach from the test person’s computer without being directly started by the test leader every time the test person clicks on the relevant symbol. But otherwise the functionality should be ‘demonstrated’ (enacted!) by the hidden test leader in every test run.

This would make it easy to change experimental set-up and also to start testing at all, since not much programming would be needed. I presume the pedagogues like the graphic design work which will be necessary for the things appearing on the test persons screen, at least when this graphic design is fairly simple and results in products that can be used in teaching. Ease of change of the experimental set-up clearly has great advantages in the initial development process. A development process which is simply progressing by refinements is not recommended by many experts. For instance, the interaction designer Bill Verplank says in an interview:

> What happened was that they set out with some very fixed notions early on, and simply kept refining them, so there was no real comparison of alternatives. There was really only one idea and they refined it and refined it and there were maybe two or three little alternatives. Then one would die and they would pursue the main direction again. No-one was ever very satisfied with the design and what I attribute that to was that they had a working prototype even before they decided what the product was going to be. (Preece et al., 1994: 467; cp. ibid.: 539ff.)

Compare also what Faulkner says about programming environments which supports ‘rapid prototyping’: “often causes the systems to be a mishmash of hacked together code” (1998:104). (The same phenomenon is visible at web sites – they are easily built by linking, but fortuitous ideas implemented in individual web pages lead to inconsistencies.)

Note that in the above-sketched laboratory testing, no “working prototypes” are used, but instead more or less *manual* prototypes are tested. A further step is needed to really implement the ideas tested with a Wizard of Oz setting. Such an implementation could either consist in making a real product or making a working prototype for further tests. Why then a manual prototype that looks real – could we not sketch with paper and pens? The children are not able to discuss sketches on a piece of paper – or, anyhow, if it is their *reactions* we want to test, then the test has to look real. I could add that in these tests, beside one or several test leaders in the control room, their will be a pedagogue or therapist sitting beside the test person in the test room, because that would be the normal use of any real product.

Admittedly, everything in the development process will not be solved by importing the Wizard of Oz-technique into the field of graphic interaction design. Let us quickly review some obstacles.

- Lots of animation would not be possible, because that would exhaust the test leader in every test run (instead of exhausting her already in prototype design time as in the 4GL case). But keeping the pedagogical aim in focus all the time, we could probably make decent test set-ups even if we exclude avatarish animation frills like walking legs or 3D. Still, the physical requirements of the laboratory, especially the input devices for the test leader, could be demanding. This is the case even if the test person would use only an ordinary system (or, more correctly, what looks like an ordinary system), which probably is what an implementation would run on outside our lab. The Oz-lab could be equipped on the test leader's side with whatever fancy equipment.
• Another problem could arise if the person in the control room does not follow the interaction schema defined in advance. Then no test of the product idea is actually performed. This is not to speak against improvising in the Oz-lab but to state the importance of the difference between an improvising session and a session meant to test a product idea.

• If an idea appears as successful and a professional multimedia designer starts to work on it, already the graphic changes that the designer may introduce (to make it look professionally made), may remove the product so much from what has been tested that the final product is untested. On the other hand, for a right-minded designer, the material produced by the naive wizards would provide a very precise specification. Graphic figures made, interaction schemas developed, video recording in test room as well as in control room should provide an ample ground for software development.

• Lots of ideas of interactive training aids are nurtured by my partners in the Department of Special Education – what if all these ideas turn out to work? They would not find funding for real implementation of these ideas... But perhaps an intermediate step through some kind of multimedia design tool would be possible, and perhaps student projects in subjects like multimedia design could be implementing special ideas wanted by some individual therapist for some occasions.

So that was four problems with this approach to software specification but also some comments about possible (part-) solutions. I have not dealt with possibilities to use such a mock-up system directly for therapeutic purposes, and, in fact, I see a rather limited usefulness in this although I will not expound on the reasons here. Instead I would like to mention another 'therapeutic' or pedagogical issue.

By not requiring the pedagogues and therapists to have a clear notion of a program before they enter the test lab, but rather letting them start doing graphic design work for an Oz mock-up, I think it will be easier for them to start specifying their needs, as would of course also be the case if they had a reason to make paper mock-ups. What is more, if we manage to set up a decent graphic Wizard of Oz lab environment, there will be the possibility for them to get used to the limitations of an interface-mediated communication. This would pave the ground for further thoughts on what a computer program could and should do. That is, the lab will be a pedagogical aid to lend the pedagogues some insights concerning computerised interactivity. Perhaps that would be a rewarding experience also for beginning students in the Information Systems subject; some of them have no idea of programming and do not seem to get the point that their programs know only the input given during runtime. I think also multimedia students could benefit from sometimes meeting the user via the interface. As would probably I too.

I have to admit there are easily used multimedia production tools (like MultiMediaLab; Hägström, 1999) aiming not the least at teachers and their pupils. Such production tools would of course also teach limitations and possibilities, but rather of the tool in question than of interactive software in general.

One could ask if I intend to base the work on some dialogue theories (possibly extending them) or semantic or semiotic models (there is even a "1st International Conference" coming up on Computational Semiotics in Games and New Media; see COSIGNS 2001 in ref. sec.). However, I think the focus for the moment has to be on the development of the lab. What is feasible to use in an experimental setting of this kind? How could it aid the initial target group, the educationalists / therapists, and what is needed for other designers, naive or not, to appreciate the set-up?

Thus, a few words on the development of the lab could round off my presentation.

Interviews with the initial target group

In the spirit of the Scandinavian tradition for system development, the special educationalists are involved in various ways (Flensburg & Friis, 1999; Bedker & al., 2000). According to the above-referred text-book on Human-Computer Interaction, "Participative design (sometimes known as the Scandinavian approach) accepts the importance of involving users in the design process and, indeed, argues that they have the right to be involved in the design of the systems which they will subsequently use." (Preece & al., 1994: 375)

2 More generally speaking, it would indeed be interesting to see how people communicate with graphic means only (like Healey & al., 2000), but that would be outside the scope of defining software. Over the Internet we have similar phenomena in netgames and in webcam software where not only written chat but also a common drawing pad can be used by the interlocutors.
For the planned research environment this would imply focusing only on the people of the Department of Special Education. While I hope the research facilities would be possible to utilise in more varied circumstances, I have invited the special educationalists to join a reference group. Presently, five persons have joined. My interest in the Disability-&-Language researchers, except for my general linguistic interests, is that they have a laboratory and that in their turn have a target group which is very heterogeneous. They deal with all kinds of handicapped people, from Down’s syndrome children to motorcycle casualties. This means that when they are defining methods for linguistic training, they are paying respect to each individual and his constraints. In fact, they are always developing new methods including material stuff to help in the linguistic training. And they are very positive to develop IT-based material to aid pedagogues, speech therapists and others, but as yet they have not had the capability to do so.

Together with one of them, I have interviewed all of them to learn more about their views of teaching aids and how they go about to make such and what kind of evaluation procedures they have. We also included questions about their ideas of computer-supported teaching or training. (The professor, Irene Johansson, was not interviewed because I have already been discussing digital versions of her material with her.) To give a brief summary of the responses, ‘teaching aid’, beside not being a self-evident concept, could include everything from ‘myself’, over manual signs for spoken words and cut-out paper figures, to “computer”. I did not want to limit their interpretation beforehand, because my aim was to have a little discussion about the development process. Their thinking was centred around methods, and so whatever aids used were just part of the method being used or developed. Evaluation was carried out in actual use, which is something one would expect with such a varying set of clients. There is no standard population to draw a representative sample from. Everyone had not made her own teaching aids, but those who had were of the opinion that it was possible to use also things that they were not quite satisfied with. The degree of acceptability seemed to depend on who would use the material. Sometimes the material developed was not always used by the therapist herself but by other adults who had received instruction. Evaluation was then through feedback from these adults although initial testing together with adult and child was common.

Finally, their answers to the questions relating to their ideas of computer-supported teaching or training were quite interesting and not only so because these answers will provide a starting point for the requirement specification of the test lab. Everyone had some opinion on possible utility of such support although no-one had developed any such material and although some claimed their computer knowledge too small to contribute. They have been talking about digitising existing material at the department, which probably accounts for the fact that everyone had an opinion. Therefore it is interesting that the answers were very varied, ranging from underlying principles for computerised aids to concrete things one wants to be able to do in a simpler way. And one person mentioned a social aspect: "Prejudice: men will be more available in the training process."

References

COSIGN 2001 Computational Semiotics in Games and New Media. Call for Participation.


Addison-Wesley, Harlow, England.
