Survey Results:

Usenet Questionnaire on User Interfaces
June 1992

by
Valerie Hall

val@marsh.cs.curtin.edu.au

Introduction

This survey was undertaken as background research for a talk and essay on user interfaces. Improving user interfaces will help to solve the software crisis, how much it will help is unknown. The software crisis resulted directly from the introduction of third generation hardware. The problem is that software is not keeping up with advances in hardware. It is often late, costs more than predicted, is unreliable, difficult to maintain and performs poorly.

I already had references on user interfaces for my talk but felt that some raw input would be of interest. Theoretic techniques are fine, but are they being used? Is the quality of user interfaces more, or less, important to users than we would expect it to be? I wanted to find out what real people thought about user interfaces, and get some up to date information.

The Surveys

In an effort to get quick responses, I decided to make use of the Usenet to distribute my questionnaire. The choice of questions and the choice of news groups had to be done in a way that ensured that the answers I received would be worthwhile. I wanted people who knew about user interfaces. I chose comp.software-eng and comp.graphics.

The software engineering group was a fairly obvious choice as my research was to come from a software engineering viewpoint. The people reading this group would know about the software crisis. They would also have a overall perspective on the creation of software. This perspective would give a good indication of the importance of user interfaces with respect to other components of software. Hopefully I would get replies from people who have had experience with large systems and the good and bad reactions to the user interfaces involved.

The other group, comp.graphics, was chosen for three reasons. The first is that they are likely to be involved in the creation of user interfaces, especially graphical user interfaces. Secondly, being involved in creating user interfaces would give them a more critical view of the interfaces they use. I wanted examples of good and bad interfaces and the reasoning behind their choices. The third reason was to be able to find out what graphics people thought the future for user interfaces would be. I feel that the interfaces are becoming more and more graphical and by asking computer graphics people I would get a good idea of where things were heading.

The Questions

To try to get the best possible responses, I kept the questions short and to the point. The questions were changed slightly for the different audiences. Both questionnaires had five questions. The topics covered were :

A - influence of user interfaces.
B - approach to/priority of user interfaces.
C - user expectations for interfaces.
D - predictions for the future of user interfaces.
E - examples of good and bad interfaces.

Topic A - Influence.

These questions aim to find out the respondent's opinion on how much influence the user interface has on the user's acceptance of software. I am looking at user interfaces in regard to what they can do to help ease the software crisis. Sommerville (1989, p5) states that, "...software should provide an appropriate user interface. It is now clear that much software is not used to it's full potential simply because the interface that it offers makes it difficult to use. The user interface design should take into account the capabilities and background of the intended system users and should be tailored accordingly." Thus, user interfaces do have their part to play in combatting the software crisis.

Two questions in the software engineering group's questionnaire and one question in the graphics questionnaire were related to the influence of user interfaces.

Software Engineering

1) What effect do you think work done in the area of user interfaces will have on the Software Crisis?
2) Do you think it will have much influence?

Graphics

1) Do you think improvements in user interfaces will help with user acceptance of software?

Topic B - Priority/Importance.

My questions in this area were aimed at finding out how important the respondents thought user interfaces were in the production of software. To make sure I got a response that was their true feelings, I worded the questions in a way that would get a reaction. "Should we worry about user interfaces if we have time to spare??", is bound to get a response, especially from people who create user interfaces for a living.

Each questionnaire has two questions on the priority and importance of user interfaces.

Software Engineering

3) How important do you think user interface development is?
4) Should we concentrate on user interfaces, or are they something we should look at if we have time to spare?

Graphics

2) What sort of priority do you think user interfaces should be given?
3) Should we leave it until last and work on it if we have time to spare?

Topic C - Expectations

This question was only given to the software engineering group. It aims to find out what users want from a user interface. There may be some golden rules on what to do, and what not to do, when creating user interfaces. It is more than likely that users will want a say in the design of their interface. Is this common? Users may want a very elaborate system without realising how difficult and resource consuming they are. Do users ask for too much?

Software Engineering

5) What do clients expect from us as far as user interfaces are concerned?

Topic D - Future

I decided that the graphics respondents might know more about the future of user interfaces. The trendy interface right now is Virtual Reality. Do people really think it's going to get anywhere, or is it just a fantasy toy? There may be standards in the pipeline for user interfaces which would make it easier for users. Should we expect some huge advances in interfaces over the next few years, or will we be getting more of the same?

Graphics

4) What do you feel the future of user interfaces is going to be? (VR?)

Topic E - Examples

Being users with a background in computers makes the graphics group a good source of critical overviews of different systems. What features do they think are important in a user interface? My thoughts are that the match between the user and the interface, along with consistency and a good help facility are crucial for user satisfaction.

The results of this question will be interesting to compare with the responses to the other questions as it will give an evaluation of user interfaces. A sort-of indicator of how well we're doing.

Graphics

5) Do you have any examples of really good or really bad interfaces? (which ones and why?)

The Respondents

There was a good mixture of respondents. Out of the 19 replies, five were students, ten were software developers and four were interested parties from other fields. The software developers included systems analysts and designers along with people involved in developing applications. The replies from people in other fields were particularly interesting as they came from the point of view of users with limited backgrounds in computing. The originals of the replies are in Appendix C.

The Answers

One the whole, the answers I received were very informative. I did get one person who asked if I was a Mac representative looking for quotes for an advertising campaign, which was probably a fair comment. An overview of the answers to each section follows.

Topic A - Influence

Almost every reply stated that user interfaces were important when trying to sell a software product. The graphics group were agreed that the quality of the user interface helped with user acceptance of software. Many respondents felt that the user interface was the component that puts your product ahead of the competition. A few people mentioned the early days of window based systems when they were very slow and used up a lot of memory. Even though the systems had problems, users still chose them ahead of command line interfaces. One of the responses stated that a good user interface could reduce training time and let users with little experience with computers do work that would otherwise require skilled labour. Another response said that user acceptance was fairly hit and miss. Quite often, the users would have a bias to a certain type of system (eg. window based) and that could make all the difference when making decisions about software.

The software engineering group were mixed in their responses. They all had something good to say about user interfaces, but could see problems with putting too much emphasis on them. User interfaces are an essential, key component of the system. "On the US DoD projects for new systems, as much as 50% of the code is concerned with the user interface, according to the comments of the government labs we work with." Another respondent said, "Read Myers and Rossons paper "Survey on User Interface Programming" in this years ACM CHI 92 proceedings. In their survey an average of 48% of systems code is devoted to the user interface portion." With so much code devoted to the user interface, it makes sense for people interested in improving code to try look at improving methods for creating user interfaces. User interfaces can increase productivity by making computers easier to use.

One possible problem with concentrating on user interfaces was that over-emphasising the user interface could leave a shell of a program. The program would be very pretty, but not very functional. Another problem is that users are expecting interfaces that are the state-of-the-art. "The work done so far will make the crisis worse. Why? Because the current work shows that developers have to work harder to produce applications of more than acceptable *usability* - where usability is some measure or value judgement of; how easy an interface is to learn, how effectively it supports the actual computerised task and how satisfied users are by the interface (as measured by questionnaires)." It is possible that developers will have to neglect other areas of their programs to give more time for creating a satisfactory user interface.

On the whole, the replies were agreed that user interfaces are very important when trying to improve software. Developers should be wary of putting too much time into the user interface, however. The product still needs to have substance to be worthwhile.

Topic B - Priority/Importance

When asked to rate the importance of user interfaces to a system, all of the replies gave them top, or next to top, priority. A few people listed their priorities:

> "The three priorities of software development (for customers other than yourself) are features, interface, and performance. Interface comes at least as high as the feature set."
>
> "As a rule of thumb, I'd say user interface comes after functionality and before "raw" performance for the general user community."

"The user interface should be given second priority to having no errors."

"Second [priority]. First, solve the technical issues your program is designed to solve."

"user interface development is at least as important as verification/validation/correctness and management."

"About the same priority that design concepts are given to automobile construction. If its an ugly/impracticable car would you by it?"

Most people considered the correct functioning of the product to be more important than the user interface.

When asked whether the user interface should be looked at only if there was time to spare, most replies thought that sort of approach would be ludicrous. One reaction was, "[Expletive deleted] No!" Another respondent said, "the best interfaces (to me) are those that are designed concurrently with the function and are customized to be most effective/ intuitive when in use with that function." Many said that it was best to start from the user interface and work back through the system. Another response to whether or not to concentrate on user interfaces stated, "Definitely yes. If you want to sell your products yes. If you want to meet international standards currently being drafted (ISO etc) yes. All your competitors will be making their products easier to use. If your products get a bad reputation for ease of user you will lose market share." It seems that the user interface should be created in parallel with the rest of the system if the package is to be cohesive.

Topic C - Expectations

User expectations were said to be very high, often impossibly so. Some users wanted perfect software. Software developers are expected to, "meet their requirements even if we don't fully understand them." The most important thing is to find out from the user what they want. "Clients expect different things depending on the end user and the complexity of the product. It is my opinion that the user be asked right off the bat what kind of interface he is expecting since it could seriously affect production costs."

One respondent said that, "my client expects a user interface that, (prioritized list),

1) Works
2) Works according to the documentation
3) Has some additional bells and whistles to make things "easier" for the user"

Clients are often very hard to please. They are "rarely tolerant of less useful (and flashy) user interfaces than the current state-of-the-practise." The interface should be useful for accomplishing the user's objectives, and should not have any nasty suprises. "Our customers generally expect to be able to use complex instruments without a manual. Many expect to be able to use most functions of the instrument within 10 minutes of opening the shipping carton. As you might guess, that's a difficult job."

To create the right system, "we need systematic ways of capturing requirements. Then we need to rapidly prototype systems and do incremental, iterative development, involving users in product development. Then we need to provide adequate documentation, training and support."

Topic D - Future

The consensus for the future of user interfaces is for them to become friendlier and more intuitive. "Virtual Reality is the future--but its going to take a few years. The bulk of the computer systems today are under-powered and could not begin to support VR. Consequently, we will slide towards more and more graphical user

interfaces." One response was, " [the future for user interfaces is] Whatever the next stupid thing they can pawn off onto an unsuspecting batch of users."

Many people mentioned multimedia and the use of audio and speech I/O. "VR is really an extended instance of multimedia in my opinion." Gadgets that could come to the fore include:

> "Pointers controlled by eye movements, more touch screens. How about real-time productivity graphs for management?", "Virtual Environments and graphical interfaces such as the BBN Simnet (Simulation Network), head-mounted tracking displays, camera-mounted vehicles, excellent simulations including dynamic terrain (with full physical modeling and capability), data gloves/paws, etc."
>
> Looks to me like things are going to get very exciting!

Topic E - Examples

The reasons for classing a product as having a good or bad interface were mainly concerned with ease of use. If a product was full of inconsistencies, or the user had to remember lots of different commands, it was given the thumbs down. Most command line interfaces fell into this class. In most cases, people thought that window-based systems were the nicest ones available. The correspondence between icons and their meanings and pull-down menus were very popular. On the other hand, Brett Bensley states that " [iconic] interfaces have reduced users to the 'wordless comic strip' level... [one] problem is misrepresentation of icons...what has one meaning for one person may have another meaning for someone else." The screen can get crowded by icons and having to weave through menus can be frustrating for users. Often a combination of windows & menus along with hot-keys can make life a lot easier for the user.

Other reasons for liking an interface include consistency and the ability to set up the system for the user's needs. "I like unix and Xwindows because you can change configurations."

The Good List

Mac philosophy
Unix and Xwindows
Windows (abstraction)
Graphical User Interfaces
The latest word processors
Mailtool
The new SGI Operating System

The Bad List

dBase
Windows and Icons
rn
mail
Word Perfect 5.0
vi
Non-mouse products

Extra Comments

Some of the people who replied to my message gave information that wasn't related to a specific question, but is still worthy of mention. Anecdotes about their experiences with users and user interfaces illustrate the points they made when answering the questions. Here are a few of the "extra comments".

David Bainbridge

David makes some general comments about user interfaces, "The user interface is the first thing that a client uses, and since most clients are not too intelligent even choose a product wholly on the user interface and not the guts. Thus quite often if you can wow the client with bells and whistles they will overlook the actual usefulness of an application. (At least until they have used it for a month.) One further comment ... UIs need to be customizable meaning one UI does not fit all. Each individual needs to be able to personalize an applications UI so it fits 'just right'."

Brett Bensley

Brett's comments were the only ones that had anything bad to say about windows. He has seen people hide bad programs behind flashy interfaces, "The users have been taught to expect a fancy interface (Thanks Apple and Microsoft). If they don't get the "wiz-bang" interface, they don't seem to want the program or package! I have seen a really discusting program out sell a good program just because of interface. The fancy user interfaced one didn't even come up with correct answers 20% of the time, but the users didn't care it was "easy to use!" (And much slower, less accurate, more time consuming, and frequently confusing to the user)"

The novelty of windows and icons can wear off, "Frequently, I have seen and heard people say, 'why don't they get rid of these little pictures and give me a character or a key to press. I have to memorize what these pictures stand for, so the picture could be anything, even a letter....and it will take less time, and less hardware to do it!'" In Brett's experience, users often forego a snazzy interface for speed and functionality, "I frequently get calls about user interfaces asking if we would write our programs for GUI's and other interfaces. When we do, most prefer to return to the command level programs instead, even though there are very little differences in functionality. Comments have included 'requires less memory,' click, I don't have to move my hands from the keyboard,' etc." If this is indicative of most users, maybe we have to rethink our user interfaces.

Paul Black

Paul Black compared software to other saleable items, "Can you imagine if automobile makers had "look and feel" lawsuits? When was the last time you needed to look in the users' manual to drive a different car? In contrast, how about operating a different VCR, graphics (paint) program, or even a multi-function wrist watch? A 5-year old can use a telephone to call half way around the world, but how easy is it to use email?...users are surprising tolerant of (don't notice) misleading prompts, useless clutter, and program failures. They'll develop the feeling that they don't like to use such a program, but they'll rarely be articulate enough to say this or that is bad. (Contrast: what can and do people say about an automobile's looks, handling, and features?)"

David Hasan

David comments on teaching students about software engineering, user interfaces in particular, "From a SW engineering point of view, I believe that focusing (for a while) on user interface could have a great benefit if done with the following emphasis: Task the class to design the software with a GUI. Then _change_ the requirements to add that the software should also have a command line interface. This captures, I believe, the kind of evolution of requirments that is typical of real world SW development/ maintenance, but it does so with a very narrow focus and so is "doable" in a semester. As I see it, the final "product" of a class like this could be a _paper_ (!) discussing the way in which the design "evolved" when the rqmts. changed (i.e., gracefully or in catastrophic rewrites) and perhaps some general discussion of interfaces and how they relate to SW engineering in this context.

My concern about "user interface"-based CS curricula is that it encourages the students to dabble with the visual appearance of the software for the entire semester and end up with code that has the interface sprinkled throughout. In fact, that's my concern about GUI in general: I'm not convinced that it will result in better code -- it might become horrendously ossified because of the plethora of (for example) X calls spread everywhere you look ... "GUI considered harmful"... (!)"

David Medberry

David looks at the commercial side of software, "User Interfaces are either the first or second most important aspect of a computer program. In a commercial/competitive/office automation program they are the most important aspect of the product. This is largely because the feature sets of most commercial products overlap virtually 100%. Their distinguishing characteristics become their user interface. In addition to the look and feel of a user interface, the simplicity and how well an application implements a user interface are important. In a vertical application market, the user interface is the second most important part. Naturally, if you have the only multi-dimensional universal whatzit program, and there is a need for multi-dimensional universal whatzit programs, you are going to sell some. However you had better start improving your UI because a market never stays exclusive for long and the next developer to make a m-d univ. whatzit program will make it easier to use (simpler, or faster, or smaller, or prettier, etc.)"

David also mentions a common approach to porting systems to different machines, "Most real developers these days are developing for multiple platforms. In doing so they develop an engine, write to a common API, then glue on the specific UI for a given platform.

To whit: To develop a program that will do ray tracing and present the final image on screen.

1) Write a core Ray Tracer
2) Ask the API to provide you with the parameters of the scene
3) Return to the API the finished bitmap.
4) Port 1: To port to Mac, write the glue so that the parameters of the scene are user settable via a menu with one or more dialog boxes. E.G. A menu of objects. A menu of Colors. A dialog of positions and sizes."

Bradford Needham

Designers and users often see the same piece of software in completely different ways, "I'm constantly amazed at the ways in which our initial thoughts of an 'easy to use' interface cause customers grief. For example, one product I recently worked on has a cryptic line of 'helpful' text on-screen. As the user moves around the menu system, the 'helpful' text changes. The trouble is, every user who tried out prototypes of the system found this 'helpful' text invisible. Because of the size of the font and the activity in other parts of the screen, this text is literally invisible when you visually attend to the menu items...Testing user interfaces with real users doing typical tasks is essential, yet many developers omit this step."

David Saranen

David had a lot to say about user interfaces after years of working with them, and the people who try to use them. He gives these guidelines for creating a user interface:

1) Can my parents use it? I saw a newspaper article a couple years ago about a coffee maker given as a gift - it sat unused because the recipient couldn't figure out how to program it. The same applies for VCR's and tape decks. A computer would be an outright disaster.

2) Can the kids use it? If they push the wrong key - what can they destroy. Related to that - usually something is incorporated in as a message 'Are you sure ...' - the messages can be so cryptic that the user has no idea what it is about. I work on a computer using Windows in a security application - the users had a three hour training session.

About once a week, someone would close the program manager - shutting down our security. Then you would have to reboot - and the data file changes made since the last reboot would be lost. The vendor made it difficult to save data file changes - and the operators weren't supposed to save the files.

3)Is the whole thing configurable? I'm starting to work in Russian. It has been a headache - I can do word processing - but my dictionary uses the latin alphabet - so they don't make sense.

Murphy's Law often applies to computing, David gave two corollaries:

"One of the corollaries of Murphy's Law: If builders built buildings like programmers wrote programs, the first woodpecker to come along would destroy civilization."

"Another corollary of Murpy's Law - it's impossible to make a system foolproof because fools are so darn ingenious."

Conclusions

Commenting about user interfaces is a very good way to spark up a very interesting (and often heated) discussion amongst computing people. As with everything in computing, people have their favourites, their biases. Most people will agree on some general points when pressed.

1) User interfaces are important for user acceptance.
2) User interfaces must be built to suit the user's needs and abilities.
3) Users judge a system by its interface.
4) The user interface should come first or second in priority during the creation of a software product.
5) User interfaces should be designed in parallel with the rest of the system, not left until last when there is no time left for them.
6) Consistency within an interface is imperative if the user is to make any sense of it.
7) A good help facility is a necessity.
8) Systems will become friendlier and more graphical in the future.

I've learnt quite a bit about how people really feel about user interfaces through this survey. Many of the people who sent in answers to my questions had been working in the software development industry for many years. They would have had to learn the importance of user interfaces the hard way, by trial and error. Hopefully we can all use their experiences to help us when creating systems to meet user requirements.

Suggested References

Coats R.B. and I. Vlaeminke (1987) Man Computer Interfaces: An Introduction to Software Design and Implementation. Blackwell Scientific Publications Ltd: Oxford.

Cordy J.R., N. Elliot and M. Robertson (1990) Turing tool: a user interface to aid in the software maintenance task. IEEE Transactions on Software Engineering. Volume 16 (3): 294-301.

Cox K. and D. Walker (1990) User Interface Design. Advanced Education Software: Australia.

Davis F.D. (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. In: MIS Quarterly, September 1989: p319-340.

Foley J., W. Kim, S. Kovacevic and K. Murray (1989) Defining interfaces at a high level of abstraction. IEEE Software. Volume 6(1): 25-32.

Gasper E. (1988) Getting a head with HyperAnimation. Dr Dobb's Journal of Software Tools 13 (7): 18.

Gross d. (1991) Man and Machine. In: Computer Graphics World, May 1991, pp 47-50.

Hayes F. and N. Baran (1989) A guide to GUI's. In: Byte, July 1989: pp 250-257.

Hurley W.D. and J. Silbert (1989) Modelling user interface application interactions. IEEE Software. Volume 6 (1): 71-77.

Keough L. (1989) The new face of computing. Computer Decisions. Volume 21(2): 36-41.

Millman H. (1990) The GUI in your future. Compute! Volume 12 (6): 27-34.

Morishima S. and H. Harashima (1991) A natural human-machine interface with model-based image synthesis scheme. In: Proc. Picture Coding Symposium 1991, Tokyo: Japan, pp 319-322.

Myers B.A. (1989) User interface tools: introduction and survey. IEEE Software. Volume 6(1): 15-23.

Newman W.M. and R.F. Sproull (1979) Principles of Interactive Computer Graphics. McGraw-Hill International Editions: London.

Sommerville I. (1989) Software Engineering. 3rd Ed. Addison-Wesley Publishers Ltd.: Wokingham.

Steward D.V. (1987) Software Engineering With Systems Analysis and Design. Brooks/Cole Publishing Company: Wokingham.

Appendix A

Software Engineering Questionnaire

Hello,

I'm currently doing an advanced software engineering unit and would like some opinions on user interfaces from news reading people for a major assignment I'm doing.

I'm interested in what effect people think work done in the area of user interfaces will have on the Software Crisis.

Do you think it will have much influence?

How important do you think user interface development is?

Should we concentrate on user interfaces, or are they something we should look at if and when we've got time to spare?

What do clients expect from us as far as user interfaces are concerned?

Any input appreciated,

Valerie Hall

Curtin University of Technology

Perth, Western Australia

val@marsh.cs.curtin.edu.au

Appendix A cont...

Graphics Questionnaire

Hi,

I'm doing a report on user interfaces and their importance in software acceptance by clients. I'd like to know how you rate their impact on how well software is received and what priority the development of user interfaces should be given. There has been problems, and they still exist, with user satisfication as far as the final software product is concerned, will UI development change this?

Here's some leading questions.....

Do you think improvements in user interfaces will help with user acceptance of software?

What sort of priority do you think the user interface should be given?

Should we leave it till last and work on it if we've got time to spare?

What do you feel the future of user interfaces is going to be? (VR?)

Do you have any examples of really good or really bad interfaces? (which ones and why)

Thanks for any information and opinions, I really appreciate it. I know that graphics people will think that user interfaces are all important so I thought I'd ask the experts!

Later,

Valerie Hall

Curtin University of Technology

val@marsh.cs.curtin.edu.au

Appendix B

Respondents

zippy@binkley.cs.mcgill.ca	(Fotis Xipolitakis)
bradn@tekig1.pen.tek.com	(Bradford H Needham)
hammondr@sungod.crd.ge.com	(Richard A Hammond)
paul@delfin.com	(Paul E. Black)
de@dcs.glasgow.ac.uk	(Dr David England)
hasan@emx.utexas.edu	(David A. Hasan)
shanks@saifr00.cfsat.honeywell.com	(Mark Shanks)
dkb@calvin.nrtc.northrop.com	(David Bainbridge)
cfbdb@ux1.cts.eiu.edu	(Brett Bensley)
CEPURNEEKC@sturt.flinders.edu.au	(Carl Cepuneek)
d65y@crux1.cit.cornell.edu	(Samuel Paik)
saranen@epx.hscs.umn.edu	(David Saranen)
brian@epx.hscs.umn.edu	(Mr. Big Stuff)
tom@vexcel.com	(Tom Baltzer)
medberry@logdis1.oo.aflc.af.mil	(David W. Medberry)
debora%ernie@ac.dal.ca	(Debora Dunphy)
tct!deep6!amaroney@myrddin.sybus.com	(Allen Maroney)
marrou@ist.ucf.edu	(Lance Marrou)
watters@cs.dal.ca	(Carolyn Watters)

Appendix C

The following pages contain the original replies to my posts about user interfaces.

Software Engineering Group

Graphics Group