

USER INTERFACE DIRECTIONS FOR THE WEB

*If you think it's crowded now, just wait!
The Web is heading toward its own year-end calamity
unless some skillful maneuvering is applied—quickly.*

■ Jakob Nielsen ■

The Web is about to face its own Y2K crisis—one that has a great deal in common with the problems facing the mainframe industry. We know it's coming, the solution is easy in principle, but difficult in practice due to sheer mass. And, we can safely predict that much of the problem will remain unsolved by the time it hits the fan.

The Web's Y2K crisis is due to the number of Web sites that will go online in the next few years. Figure 1 shows the growth of the Internet and the Web during the present decade. Since the diagram has a logarithmic y-axis, the curves represent exponential growth. If the growth rate does not slow down, the Web will reach 200 million sites sometime during 2003. Since the Web will have about 4 million sites by the time this issue of *Communications* reaches subscribers, we can conclude that about 196 million new sites will go online during the next five years.

The world has about 20,000 user interface professionals. If all sites were to be professionally designed, and if a site design did not require more than a single UI professional, we can conclude that every UI professional in the world would need to

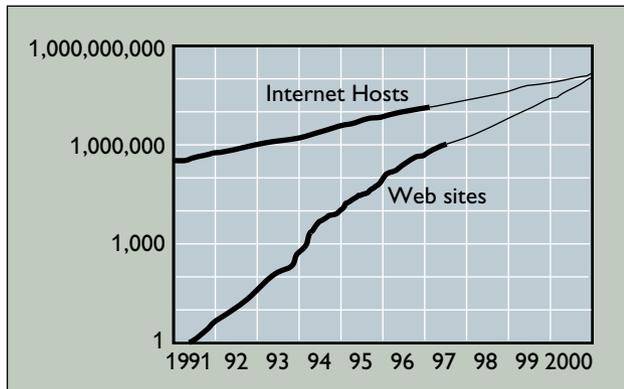


Figure 1. Growth of the Internet and the Web. Heavy lines indicate empirical data and thin lines indicate projected growth.

design one Web site every working hour from now on to meet demand. This is obviously not going to happen; even a small site takes more than an hour to design, designing a large site requires collaboration between a team of UI professionals, and some UI will prefer to stay with their traditional role of software design rather than moving into the wild world of Web design.

There are three possible solutions to the problem:

- Make it possible to design reasonably usable sites without having UI expertise;
- Train more people in good Web design; and
- Live with poorly designed sites that are hard to use.

The third option is not acceptable in my opinion. Unless the vast majority of Web sites are improved considerably, we will suffer a usability meltdown of the Web no later than the Year 2000, and most people will refer to the Web as “oh, yes, we tried that last year, but it was no good.” Thus, we have to strive for a combination of the first two options: making it easier to design acceptable sites, and increasing the availability of staff who know how to do so.

Making it easier to design usable sites will likely involve a combination of templates and design conventions. Several Web authoring tools have already started providing templates for the most common types of pages as well as sets of templates for common subsite structures. By simply pouring content into such templates, even a novice author will often get an acceptable result.

Templates can never cover all design needs, and there is also a risk that templates from different vendors will be different and, thus, result in proprietary UI standards. Users should not have to change their

interaction expectations depending on what authoring tools were used to build the sites they visit. To ensure interaction consistency across all sites it will be necessary to promote a single set of design conventions. For example, one design convention should be to put a logo or other site identifier in the upper-left corner of every page and link it to the home page. Other design conventions would cover how and where to activate search, how to communicate the search scope to users (are you searching the entire site or a subsite?), and how to change the search scope (through links to differently scoped pages or through a widget right on the page?). This short article cannot enumerate all the necessary conventions, but considering the time it normally takes to reach international consensus about Web issues, efforts toward establishing such conventions must be initiated without delay. It would be unfortunate for the cross-platform potential of the Web if interaction standards were set by individual vendors rather than by the community at large.

Luckily, the most basic issues in Web design are fairly well resolved. Certainly, ever-changing Internet technologies do raise new usability issues such as the appropriate balance between “push” and “pull”

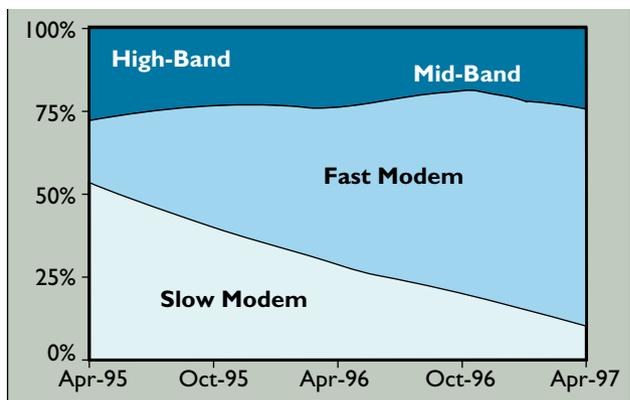


Figure 2. Proportion of users accessing the Web at various speeds according to the Georgia Tech user surveys [1]. “Slow modem” is 14.4 or slower, “fast modem” is 28.8 or 33.6, “mid-band” is mainly ISDN, and “high-band” is T-1 or faster.

content delivery, but many other design issues relate to basic principles of hypertext and human-computer interaction [6] and are, thus, relatively constant. For example, a journalist recently interviewed me for a story on Web animation, and I was able to refer him to an essay I had written in December 1995 [3]. Upon reviewing my 1995 writing I found that all the recommendations for appropriate use of animation still held. In fact, the only problem in my essay was the examples, several of which had suffered link rot or were implemented in animation technol-

ogy that was no longer supported. Human factors principles for animation live much longer than specific implementations since usability is based on people and their tasks, both of which are slow to change.

Having studied many usability factors from a variety of Web sites since 1994, I have found many design principles that are remarkably constant through changing browsers and Internet technology:

- Download speeds are the single-most important design criterion on the Web [4]. Figure 2 shows the proportion of users accessing the Web at the high speeds it was designed for has been declining over the last few years as more and more low-end

- Scrolling must be avoided on navigation pages. Users need to be able to see all their options at the same time and links below the window border when the page comes up (“below the fold” in Jared Spool’s newspaper analogy) are much less likely to be chosen than the top links.
- Content is king. Even though I was often testing sites to evaluate their interaction design, the test users were far more focused on the actual page content. Most users don’t go to the Web to “have an experience” or to enjoy the site designs: the UI is the barrier through which they reach for the content they want. In particular, gratuitous animation and scrolling text fields (so-called mar-quees) are universally despised by users because

Unless the vast majority of Web sites are improved considerably, we will suffer a usability meltdown of the Web no later than the Year 2000, and most people will refer to the Web as

“OH, YES, WE TRIED THAT LAST YEAR, BUT IT WAS NO GOOD.”

users come online. The basic response time recommendation has been the same for about 30 years [2]. Moving from one page to the next requires sub-second response for users to navigate freely and anything slower than a second is hurting users.

- Search mechanisms are essential for any site with more than about 200 pages. Many users reach for the search button immediately upon entering a Web site and the rest do so when they get lost.
- Despite the need for search, it is also necessary to have a strong sense of structure and navigation support in a site so that users know where they are, where they have been, and where they can go. Again, a very old principle [7]. Also, 20 years of hypertext research indicates that site maps will be useful because they give users an overview of the navigation space. Unfortunately, most current site maps are very primitive and lack, for example, the “you are here” indication that every mall shopper knows to be essential. Still, there is hope that future Web client software will have better support for visualizing navigation structures.

they distract from the content and slow down use of the Web.

These rules and many more are well established and have been found repeatedly in many studies. The main problem lies in getting Web sites to actually obey any usability rules. Unfortunately, it is common for sites to aim at being “cool,” “sizzling,” or even “killers” rather than trying to do anything for their users. We can only hope this contempt for users’ needs and the value of their time will be a temporary stage in the evolution of the Web. Design Darwinism will tend to drive out the most flamboyant sites and concentrate traffic at sites that follow the usability principles. Many users have said in our interviews that they go to sites like Yahoo! because they get the info they need faster there.

Going beyond site design, it is important to write content for the Web in ways that are optimized for the way users access online information. The word “repurposing” ought to be banned from the dictionary of new-media executives since the use of online information is very different from the use of printed

information, movies, or other old media. For example, we have repeatedly found that users do not read online. Instead, they scan the text, picking out highlights and hypertext links and only read selected paragraphs. Also, users are very impatient while using the Web; the medium tends to pace users ever onward. This leads to a new writing style for the Web that is based on writing multiple short segments interlinked with hypertext, designed for skimming, and structured according to the inverted pyramid style taught in journalism school [5].

Most authors of Web content are not aware of these rules for online writing, and even people who do know find it very difficult to change their writing habits to the needs of the new medium. Most of us have been through more years than we care to remember in the educational system, with all instruction from the first years in elementary school to the last years of graduate school focused on the production of ever-longer linear documents. Breaking this habit is tough, but we are not in dissertation land anymore, Toto.

To prepare for the future, I encourage schools to teach Web authoring and proper ways to structure online content. Building project Web sites is already the dominant way of exchanging information in many organizations, and Web authoring will surely be a much more important skill than memo-writing when today's students graduate. Even small children can create their own Web sites and larger projects can result in collaborative authoring of shared information structures across multiple schools. It is much more motivating for students to create content that others will read on the Web than when they are simply writing for the teacher's red pencil. Encouraging self-expression and content authoring among students is a much more productive use of the Internet in schools than the unrealistic approach advocated by some politicians where "every school class can have direct access to leading scientists over the Internet." Let me tell you, as one of the people on the receiving end of this idea, I get far too much email to have time to reply to the ever-increasing flood of students who want me to do their homework for them. Do-it-yourself is the best way to learn.

Even though much is known about basic issues in Web design and content creation, Web usability has many unresolved issues. There are major problems with current Web technology in deciding on the best way to support applications over the Internet. Because Web browsers can double as GUI builders across multiple platforms, many people have created quite complex applications that run within a browser window. Unfortunately, browsers are not suited for

complex interactions with underlying data objects, as exemplified by the infamous back-button problem: Consider a user who goes into a shopping application and adds a purchase to his or her shopping basket and then decides not to buy that product. Most users will click the browser's back button and return to a screen that shows a shopping basket without the last purchase. Since the server doesn't know what the browser is doing, it retains a state where the user's action has not been undone.

My current thinking is that it may be best to reserve the Web for what it is good at: browsing information as well as lightweight interactions like entering a search query or authenticating authorized users through a log-in screen. More heavy-duty applications are probably better treated as client/server applications and implemented as traditional GUI in applets written in a cross-platform Internet-aware language. I am not completely confident in this assessment, however, and it certainly requires additional research to partition the space correctly and determine the boundaries between lightweight and heavy-duty interactivity.

The Web is a cross-platform design environment, and the ability to project a single design onto a wide variety of platforms presents new UI challenges. Traditional UI design has had to deal with maybe a factor of 10 in performance between high-end computers and the lowest-end machine on which a given application would run and a factor of six in display area, ranging from a small 480x640 laptop screen to a large 1600x1200 workstation screen. These differences pale against the factor of 1000 in bandwidth between modem users and T-3 users and the differences in interaction capabilities between a car phone, a palm-sized PDA, a WebTV, a traditional computer, a computerized meeting room with a wall-sized display, and a full-immersion virtual reality environment, all of which will be used for Web browsing.

Currently, the recommended way of dealing with device diversity on the Web is to separate presentation and content and encode the presentation-specific instructions in stylesheets that can be optimized for each platform. This approach requires authors to structure their content appropriately, but current authoring tools are poor at facilitating structure editing. Many Web authoring tools employ WYSIWYG editing, which has an unfortunate tendency to make authors believe the readers will see pages that look the same as what the author is seeing on his or her screen. Obviously, this is not true: the user may not even have a screen but may be using a voice-only access device. Future Web authoring tools will have

to be based on structure editing and must have ways of making authors design for a multiplicity of display devices. Since current approaches to stylesheet-based editing are notoriously hard to use, we will need considerable research progress to realize this goal.

In the long term, simple style sheets will not be sufficient to accommodate Web use across widely varying devices and bandwidths. Consider, for example, delivery of news over the Web: if accessed over a large display a multicolumn layout much like a newspaper may be used, with a variety of illustrations and long story segments. If accessed on a small display, the same layout would be much less usable. A better design would have smaller illustrations, only the most important stories, and use fewer words for each story (with hypertext links to the full story). Figure 3 shows how an otherwise nicely designed page from www.news.com leaves only a paragraph of the main story visible when shown on WebTV. To generate optimal designs for different devices, the content will need to be encoded with much-enhanced structure and metainformation beyond current HTML. For example, news stories might be encoded with information about the importance of each story as well as what paragraphs can be dropped for what category of readers, and images may be encoded with information facilitating the use of relevance-enhanced image reduction (a technique to reduce the image while focusing on the most salient parts).

Web navigation is a challenge because of the need to manage billions of information objects. Right now, the Web *only* has a few hundred million pages, but before the end of the decade, there will probably be 10 billion pages online that can be reached from any Internet-connected device. Current UIs are simply not well suited to dealing with such huge amounts of information. Virtually every current UI is more or less a clone of the Macintosh UI from 1984. The Mac was optimized to handle the few documents that an individual user would create and store on his or her disk. Even Xerox PARC research, from which much of the Mac design was derived, was mostly aimed at office automation to support a workgroup and a few thousand documents. The Web, in contrast, is a shared information environment for millions of users (soon to be hundreds of

millions of users) with—incredibly—many more documents.

Web browsers are applications in the style of the currently dominant UI paradigm, so they are inherently ill suited for the task of browsing the Web. Consider, for example, how a pull-down menu (even with pull-right submenus) is an extraordinarily weak way of organizing a user's bookmarks. How to design better bookmark support is an open research problem, but it is clear a richer representation will be necessary. For example, bookmarks could reflect the structure of the information space (What bookmarks refer to pages within the same site?), and the history of the user's navigation behavior (What bookmarks

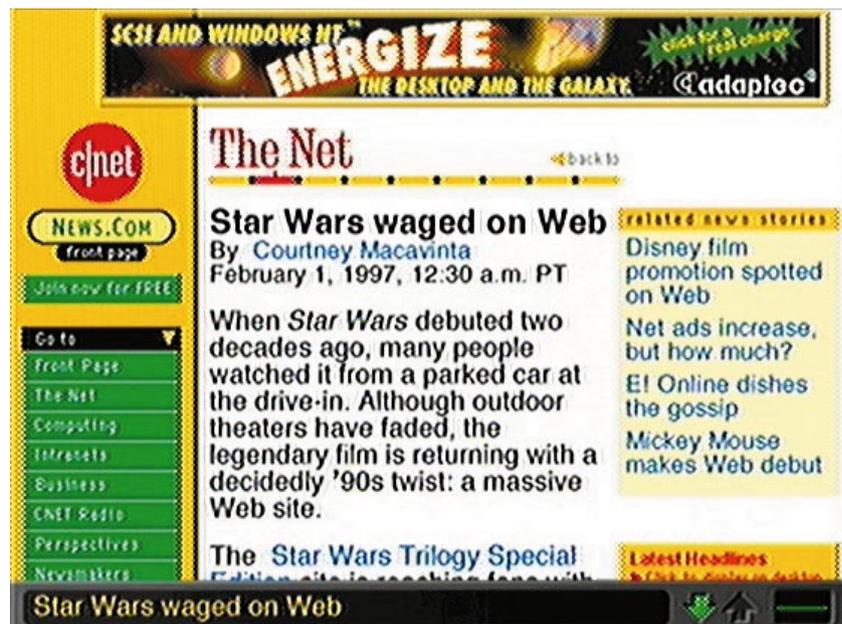


Figure 3. The three-column layout from www.news.com does not work well on the small WebTV screen: navigational overhead dominates and the user can only read a single paragraph of the main story.

are used often and what bookmarks are never used and might be pruned?).

We obviously cannot represent every single Web object in a navigation UI (given that there are so many). Thus, we will need a variety of methods to reduce the clutter. Some useful methods are:

Aggregation (showing a single unit that represents a collection of smaller ones). This can be done quite easily within a site (indeed, the very notion of a site is one useful level of aggregation, as are various levels of subsites), but it may be harder to aggregate across sites.



Figure 4. Example of adding a chat room with a reputation manager to a UI that integrates television and the Web. The illustration is a concept design from SunSoft's World Without Windows project.

Summarization (ways of representing a large amount of data by a smaller amount). Examples include use of smaller images to represent larger ones and use of abstracts to represent full documents. We need ways of summarizing large collections of information objects.

Filtering (eliminating whole wads of stuff the user doesn't care about). I am a firm believer in collaborative filtering and in quality-based filters (for example, only show stuff that other people have found to be valuable).

Elision and example-based representations. Instead of showing everything, show some examples and say something like "3 million more objects."

The Web needs to embrace the notion of quality as a pervasive attribute of objects. The Web has traditionally assumed that everybody was equal: every

site could link to every other site, and every page would be displayed, bookmarked, and otherwise treated the same. This policy is acceptable as long as it only needs to handle a small amount of information. Once the Web gets to be a hundred times larger than it is today (that is, in two years), users will need ways to zero in on the most valuable information. For example, if I search for "widgets" in a search engine, I would not want to find all 20 thousand pages that include the term, but only the best pages about widgets. The traditional information retrieval concepts of precision and recall are not well suited for the Web because they implicitly assume that users want a complete set of relevant documents. On the Web, nobody will ever have the time to read all the relevant documents, and it is more important to guide the user to a small number of high-quality documents than to achieve completeness.

One interesting approach to guiding users to good documents is the PHOAKS project at AT&T Research [8]. The PHOAKS server reads Usenet newsgroups and extracts the URLs that various posters recommend. The resulting database helps users find Web documents and Web sites that have been deemed the most valuable for the various topics discussed on Usenet. The point is that quality is determined by human judgment. I don't believe we will be able to have computers generate quality ratings for the foreseeable future.

Once humans have rated quality, however, it should be possible to automate the processing of these ratings. For example, my colleagues and I have studied a *reputation manager* for the Internet that would work the same as a reputation works in the real world. Everybody has a reputation: for example, Sue may be known as a great Perl hacker who can whip out a cgi-script for anything you need on your site; Bob may be known to always be late for meetings, but still be able to reconceptualize problems in a revealing way. If you ask around, people will tell you to go to Sue if you need a script and to Bob if you are stuck with a problem. Computers can automate this process, such that you can aggregate a reputation from a large number of other people's assessments. For example, when browsing individual Web pages there could be two buttons available for the user to say either "really great" or "really bad." There would certainly be some satisfaction to being able to, every time you've been cheated or you've seen something bad, hit a button saying "this is bad, I really warn anyone else from ever wasting time or money on this page." Or, if something was really valuable, maybe you would get a little satisfaction out of helping other users or giving the author some added business by saying "this was

good.” All of these ratings would be accumulated by the reputation manager and would help subsequent users judge the quality of the pages. Reputations would be built up for individual pages, for entire sites, for individual users (using a “person object” to represent people, possibly under pseudonyms), and for companies. No doubt, the value of a particular user’s rating of an information object would come to depend on that user’s own status in the reputation manager: a highly respected user’s opinions should be given added weight.

Figure 4 shows an example of the reputation manager applied to chat rooms. The problem with current Internet chat is it is dominated by the users who are the most aggressive in typing in their opinions and who spend the most time in the chat rooms.

Unfortunately, people who don’t have a life are probably the ones who are least interesting to listen to. In contrast, a reputation manager would include an enhanced bozo-filter to eliminate boring or irrelevant contributors and enhance the presence of valuable contributors in the interface. In Figure 4, we used the reputation manager’s ratings of the quality of the various contributors to determine their head size. Users who are rendered with big heads are the ones that have proven most interesting in the past and who, therefore, have the best ratings in the reputation manager’s database.¹

¹The World Without Windows project shown in Figure 4 was a collaboration between the author and Bruce Browne, Bob Glass, Bruce Tognazzini, and Elizabeth Waymire.

■ Hypertext history ■

The Web is a hypertext system, as indicated by the letters “H” and “T” in its basic protocol, HTTP, and its basic data format, HTML. Now that the Web has taken over the world, it may seem to be the only hypertext system that matters, but much can be learned from the hypertext systems that went before the Web and the extensive research that was done with these systems.

I did my first hypertext project in 1984 when I worked on ways of adding reader annotations to online information. Other researchers started even earlier, with the first hypertext project being Vannevar Bush’s Memex vision from 1945. Even though Memex was never implemented, there are many aspects of the Memex specifications that are more advanced than current Web features. For example, Bush envisioned the possibility for users to construct an alternative view on an information space to share with other users. Currently, the two main

ways of approaching this ideal are to email a friend a hot link and to construct your own page of recommended other sites. There is no way to recommend a trail through a larger space.

Hypertext research took off in earnest after Ted Nelson coined the term in 1965 and the first research systems were built in the late 1960s and early 1970s. The 1980s was the golden era of hypertext research and early commercial products, culminating in 1987 with the release of HyperCard and the first major conference on hypertext—Hypertext’87. If Apple had played its Cards better, HyperCard might have grown into the universal status now owned by the Web. It had a wide following, a great GUI, reasonable navigation support, and its own easy-to-learn programming language. HyperCard even had a UI design standard—something we are still missing for the Web.

Many of the designs done in

HyperCard and other early hypertext systems go far beyond the current state of the art on the Web, and much inspiration for Web designers and Web software architects can be had from surveying these projects and their visions. For example, Nelson’s early work included detailed discussions of ways of paying authors royalties for their works as these works were accessed through hyperlinking from other works. Current work on micropayment schemes should obviously be implemented with open Internet protocols and not rely on the details of a 30-year-old design, but it would be unfortunate if the rush to enhance the Web overlooked the lessons from earlier work. When working on “Internet time,” you can get to market faster by knowing what has already been done, and by taking inspiration from the more advanced designs that were laboriously constructed in the closed hypertext systems of the past.

Since content is king on the Web, the only way to increase the ultimate value of the Web to users is to enhance the quality of the content. This, again, requires that content creators get paid since there is only so much that people can do on a volunteer basis. Even though microtransaction technology has already been invented, I expect a few years of further delay due to the need to build infrastructure. Therefore, I predict that micropayment systems will be used sometime this year to provide a revenue stream to authors of value-added content. The reputation manager might be used to increase the usability of a micropayment system. For example, a user might set up a preference to automatically pay for any Web pages costing less than five cents as long as they get a good quality rating from the reputation manager.

In conclusion, the Web's early years demonstrated the compelling attraction of the "everyone, everywhere; connected" vision of the Web's founders. Unfortunately, as "everyone" grows to half a billion or more people (most of whom are not even remotely geeky) and "everywhere" expands to tiny PDAs with low-band cell modems, it is clear the current Web UIs are insufficient. Major advances are necessary in browsers, navigation, and information management as well as in content authoring. I am confident the Web will succeed and will grow into

the dominant medium of the next decade, but the eventual success may be delayed for several years unless less emphasis is placed on dazzle and coolness and more emphasis is placed on quality content and software that augment users as they go about their tasks. ■

REFERENCES

1. Kehoe, C., and Pitkow, J.E. GVU's WWW user surveys, (1997); www.cc.gatech.edu/gvu/user_surveys/papers/
2. Miller, R. B. Response time in man-computer conversational transactions. In *Proceedings for AFIPS Spring Joint Computer Conference*. (1968), 267-277.
3. Nielsen, J. Guidelines for multimedia on the Web. (1995); www.useit.com/alertbox/9512.html
4. Nielsen, J. The need for speed. (1997); www.useit.com/alertbox/9703a.html
5. Nielsen, J. Be succinct: How to write for the Web. (1997); www.useit.com/alertbox/9703b.html
6. Nielsen, J. *Designing Exceptional Websites: Secrets of an Information Architect*. (1999) New Riders, Indianapolis, Ind.; www.excellentsites.com.
7. Nievergelt, J., and Weydert, J. Sites, modes and trails: Telling the user of an interactive system where he is, what he can do, and how to get to places. *Methodology of Interaction*. R.A. Guedj, P.J.W. ten Hagen, F.R.A. Hopgood, H.A. Tucker, and D.A. Duce, (Eds.). North Holland, (1980), 327-338.
8. Terveen, L., Hil, W., Amento, B., McDonald, D., and Creter, J. PHOAKS: A system for sharing recommendations. *Commun. ACM* 40, 3 (Mar. 1997), 59-62; www.phoaks.com

JAKOB NIELSEN (www.useit.com) is a principal consultant at Nielsen Norman Group, Atherton, Calif.

COMMUNICATIONS of the acm

March 1999

Special Sections:

Multiagents Systems on the Net and Agents in E-Commerce

agents, networking, simulation, databases, e-commerce, Java, XML, WWW, smart appliances, virtual communities, human-computer interaction, interface designs, commercial applications

Display Advertising Closes: January 25, 1999

For more information contact:

ACM Advertising 212-626-0685 acm-advertising@acm