Getting It Across

Layout Issues for Kiosk Systems

Jan Borchers, Oliver Deussen and Clemens Knörzer

A clear and appealing screen layout is crucial to the success of on-line kiosk systems, public terminals that are connected to a network. This paper addresses the problem of developing such a layout, and provides several guidelines, drawn from traditional typography and Gestalt psychology as well as from hypertext authoring, and human-computer interaction. To identify how a kiosk system's primary task influences optimal layout, kiosk systems are classified into four basic types. The usability of HTML (Hypertext Markup Language) 2.0 and 3.0 to write documents for these systems is discussed, and some alternative existing environments are presented.

Introduction

Kiosk systems are computer terminals which are open to the public. They are typically installed to give information about their environment, or to offer other electronic services. [Holfelder94] defines a kiosk system as a "... computer-based information system in a publicly accessible place, offering access to information or transactions for an anonymous, constantly varying group of users, with typically short dialogue times and a simple user interface."

The success of such systems depends largely on the attractiveness of their user interface, how easily they allow access to information and how clearly they present it. This is, of course, true for most computer applications, especially with a graphical user interface (GUI), but the fact that a kiosk system may encounter an especially varied and usually untrained user community should make these aspects primary design goals.

We will try to present some approaches to layout issues for kiosk systems, showing other scientific fields whose results can be used for our task, but also pointing out that interactive hypertext systems, and especially kiosks, have many aspects that require additional rules and guidelines to be defined.

To simplify maintenance, even kiosks that do not have to exchange data with some central system to fulfil their purpose are being connected to networks, usually via telephone lines. With the development of the World-Wide Web [Berners94a], writing pages for such systems using the Hypertext Markup Language, HTML [Berners94b], has become feasible. We will show in what respects HTML in its present version is insufficient for such a task, how the emerging standard, HTML3 [Raggett95], improves this situation, and how other existing standards relate to these solutions.

An Interesting Example

To show how design can influence usability and thus acceptance of a kiosk system, let us have a look at Figure 1, a screenshot of an existing library kiosk system that runs on publicly accessible PCs.

Even though this screen looks quite usable at first sight, several shortcomings become clear when looking at it more closely, or using the system:

- The menu bar contains too many entries, making it wrap around to a second line. This looks confusing, as users expect a menu bar to really be a *bar* with some entries in a single row.
- There are too many alternative ways to achieve the same result using different interface elements. For example, to change password, a user could do any of the following:
  - Choose the entry from the menu using the mouse,
  - access the menu entry using the keyboard mnemonic Alt-P,
  - activate the menu entry through keyboard traversal,

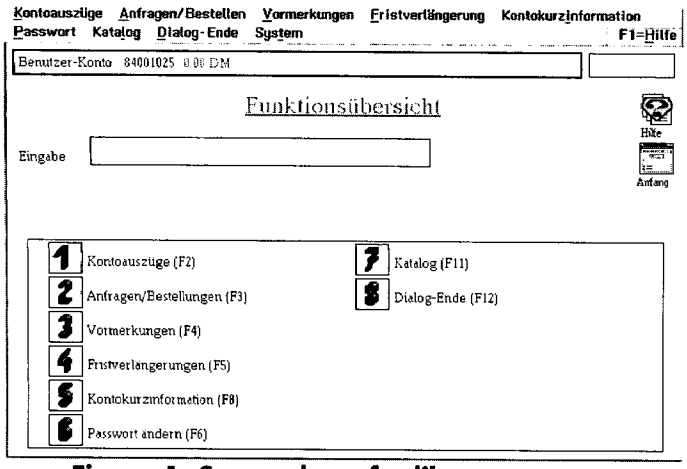


Figure 1: Screenshot of a library system

– press the function key F6,

– type a "6", followed by RETURN, into the text entry field,

– select the box with the "6" in it using the mouse.

Even though it is good practice to offer users an interface with several ways to accomplish the same task, this has clearly been overdone here. The interface would probably be easier to grasp if it did not use a menu bar at all.

- The interface is inconsistent. To see the catalog, the user is supposed to enter the number "7", or press function key "F11"(!). Such design flaws typically emerge when existing, text-based interfaces are revamped into a GUI system without proper redesign.
- The 3-D look of the numbers is unnecessary, and impairs readability.
- Something that cannot be seen on the screen: the mouse attached to the system uses an accelerating driver that makes it practically impossible for the (usually untrained) user to locate objects on the screen with it.
- Another problem not visible on this opening screen: information is displayed mostly in additional windows popping up. This tends to clutter the screen, and makes the user feel insecure about "what to do next".
- The system does not keep the user from opening an arbitrary amount of windows. Users have been observed who had pressed a key, producing an error message, but who kept entering the same information again and again: They had not noticed the error dialog, maybe because of the many windows that were open by that time, or maybe because it appeared at the wrong position.

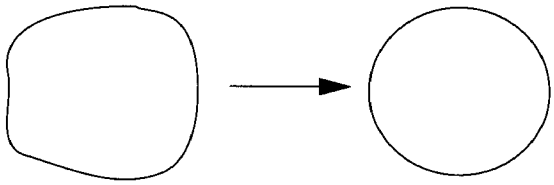
Many of the above shortcomings have a single, simple reason: the system was designed by computer experts who had not realized that not they, but computer novices would be its principal users. This is a design flaw encountered to a certain degree in many systems, but with kiosk systems it is fatal.

Gestalt Psychology: How People Perceive Information

To avoid design errors as those listed in the example, we will first have a look at a very basic question: how do we perceive objects on screen, on paper, or in the real world; and according to which principles does our perception combine objects into groups?

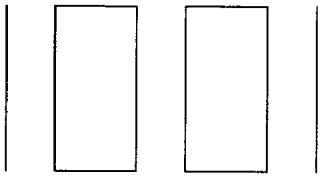
The scientific field of Gestalt psychology, founded by Köhler, Wertheimer, and Koffka in the 1920's [Koehler29], deals with these questions. The applicable essence of their research can be summarized in a number of "Gestalt laws" that explain why certain patterns are considered as belonging together, thus forming an object or appearing to be related to each other. We will cite some of these laws here (see also [Schmitt94]); many more of them have been identified.

- **Law of Succinctness** This rule, also called *Law of good shape*, states that perception tends to see objects as having some perfect or simple shape because it is easier to remember. For example, a polygon with many edges which is "almost round" is often perceived as being a circle when looked at for a short period. In a way, this transformation is essentially a built-in lossy compression algorithm of our memory: associating the already existing notion of a circle takes up less storage capacity than remembering the complex shape we actually saw.



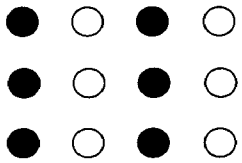
- **Law of Proximity** Another effect can be seen in the next figure: objects that are closer to each other seem to form a group. This rule is important to screen design; it gives us an easy way to indicate that certain pieces of information belong together.

- **Law of Unity** Objects that form closed shapes are also perceived as a group. This effect can change the grouping of the former example:

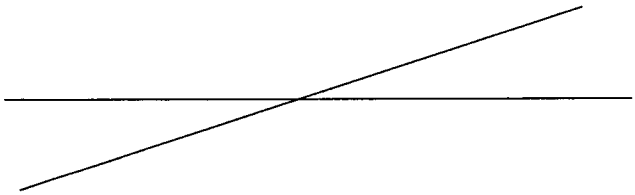


The boxes and frames used frequently in graphical user interfaces, e.g., to combine a row of buttons into a more complex user interface component with certain semantics, make use of this law.

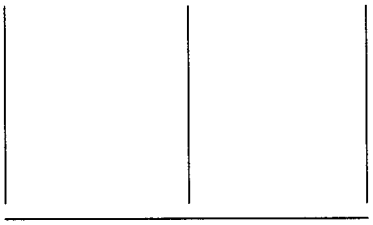
- **Law of Equality** Similar objects are another candidate for grouping by our perceptional system. This is the reason why, for example, push buttons arranged in a row should all be the same size.



- **Law of Continuity** Furthermore, there is a tendency of perception to assume continuity in objects: we do not consider the following example to consist of two lines that both have a bend in the middle:



- **Law of Experience** Finally, we tend to try to match objects perceived to things we already know. This is why user interfaces that rely on well-known real-world metaphors are more successful: they minimize memory load.



Most people will not map this figure to the letter "E" because their experience does not expect it in this position.

Going into Gestalt psychology in its whole depth is beyond the scope of this article, but it should be clear that these laws are fundamental for the way people see and understand displays. An "ideal" user interface could then be defined as one whose image in the user's long-term memory does not change anymore because the interface already obeys the Gestalt laws: the perception system does not misinterpret it when applying its rules.

Layout Rules for Conventional Documents

To come to design guidelines for kiosk documents, we will start with those rules that traditional typography offers us. In contrast to widespread belief, typography deals with much more than just "font design": it gives guidelines to the layout of pages and whole documents as well.

There are innumerable books on typography, and we cannot give a comprehensive summary of the guidelines that exist. Instead, we will present the basic schedule that every typographer follows when creating a new document, giving some examples of rules for the different steps, and later identify how those steps have to be adapted to cater for kiosk systems. This change from printed documents to public on-line systems will be divided into several steps: the first important change is to interactive hypertext documents. Next, networked documents will add another dimension to layout, and finally we will arrive at kiosk systems which we will divide into four basic types, and give separate design rules for each of those types.

The basic design schedule in conventional document layout comprises the following steps (see also [Gulbins93]):

1. *Page format:* Width and height of the document pages have to be determined. The most widely used formats (width:height) range from the DIN A series ($1:\sqrt{2}$) to $1:\sqrt{5}$ as a very slim format.

2. *Page layout:* In this step, the area of the page that is to be "used" is defined, i.e., its margins are specified. Classical ratios are 2:3:4:5 for inner:outer:head:foot margins, with the margins taking up $^1/_3$ of the page width for a spacious layout.

3. *Page pattern:* The arrangement of information inside the page layout framework has to be defined next: how many columns will be used? This directly influences line length which should be in the range from 45 to 65 characters per line.

   Since the human eye moves over text in chunks of several words, shorter lines would mean that the reader could grasp less text per "read cycle" than possible, and that more line changes are necessary, slowing down reading speed. Justification should be avoided with very short lines; ragged right formatting should be used in those cases.

   Longer lines, on the other hand, mean that too many read cycles per line are needed, and they make it harder for the eye to find the beginning of the next line, again impairing reading speed.

   For vertical spacing, professional document layouts actually use a very rigid grid. The vertical line spacing of the body text is the grid distance, and the bottom line of all elements, even headings and pictures, has to lie on this grid.

4. *Fonts and Styles:* The fonts and sizes to use for body text, headings, tables, etc. have to be determined and used consistently throughout the whole document. Classical values are: 24pt bold for chapter headings, 18pt bold for section headings, 14pt bold for subsection headings, 12pt bold for paragraph headings, 10pt for body text, and 8pt for annotations and footnotes.

   We do not want to repeat all the rules that probably have made their way to most DTP users by now, and that basically have the same essence: *don't overdo it.* A maximum of two different font families should be used, and they should be compatible to each other concerning line thickness, capital letter height, etc. Emphasizing elements (italics, bold face, small capitals) should never be used in combination on the same word.

   The overall task is to create a page appearance with an even "grey value", without particularly dark or light areas.

5. *Contents:* Only after finishing this layout, the actual text is entered into the document frame, and corrected from the contents point of view.

6. *Aesthetic correction:* Professional documents now go through a final "fine tuning" stage. Here, minor changes to kerning etc. are made. For example, a right margin does not look properly justified when adjusted mathematically correctly; lines that end with, e.g., a colon have to be extended slightly to the right to look "in line" with the rest of the margin.

Layout Rules for Hypertext Documents

Hypertexts are electronically accessible documents with cross-references that can be followed interactively. The following problems arise when trying to apply the conventional layout steps to those documents.

1. *Page format:* Most screens have landscape rather than portrait format, and are not really suitable to display information in pages. Page layout can reduce the negative effects of this, but the best solution would be to use portrait monitors for kiosk systems that are to be designed from scratch.

2. *Page layout:* To come close to traditional media sizes even with landscape monitors, there is a choice between leaving side parts of the screen unused, or using several columns in the layout. Both alternatives are not entirely satisfactory, however. Using standard screens as-is with a single-column layout results in text lines that are much too long to be readable.

   Margins, on the other hand, do not usually have to be as spacious as with printed documents, as the monitor often offers the necessary frame area around the text.

3. *Page pattern:* Hypertexts can be accessed in more ways than ordinary documents. Because of this, a consistent page pattern with an informative heading showing where in the document the user currently is located is essential to avoid the dreaded "lost in hyperspace" syndrome.

4. *Fonts and Styles:* Electronic documents can use colour much easier than printed media. However, colour is the most striking way to emphasize text parts, so only a few different colours should be used, and they should be applied conservatively. They can replace the use of different font sizes, e.g., for headings, but it must be remembered that, for example, 8 percent of males in Europe and North America have some form of colour blindness. Colour should be used consistently, and not in pairings like blue-on-red (spectral opposites lead to focus problems of the eye). Also, colour displays will always have a much lower resolution than monochrome or gray-level displays.

5. *Contents:* Here probably lies the most important difference to conventional documents. Hypertexts have to be split into small chunks that can be displayed individually, and organized in a graph-like or a tree-like structure. This usually means that conventional documents have to be restructured completely. Unfortunately, many authors instead choose to just provide an "index layer" of links that point to various sections in their still linear document. This is especially inadequate for systems where scrolling text is not possible.

6. *Aesthetic correction:* The significantly lower resolution of today's computer screens in comparison to printed media means that the fine correction described earlier is hardly possible for documents that are viewed on a screen. On the other hand, the comparably low resolution often results in rounding errors and jagged letters, effects that should be reduced through techniques like anti-aliasing.

In addition to this, hypertext documents are read interactively using a computer, which means that all the rules from the area of human-computer interaction and design of graphical user interfaces have to be kept in mind (see [Shneiderman92] for a comprehensive treatment of user interface design).

Layout Rules for WWW Documents

When writing documents to be put onto the World-Wide Web, the major problem is that most layout decisions are by definition left to the browser which is in general unknown to the author. On the one hand, this is exactly HTML's advantage. Ideally, HTML documents should contain *logical markup* only, and transforming this information into different rendering styles is done by the browser according to the resources available. On the other hand, this has unsatisfying consequences for document layout:

- *Page format, layout, pattern, styles:* The screen size of the target system, its layout policies, fonts used and colours available are all unknown, so hardly any assumptions can be made about the final appearance of the document. It might even be displayed on a text-based browser!
- *Contents:* The world-wide availability of documents should have influence on their contents. They should be available in English and, if applicable, in the local language; the author, a contact address, and dates of creation and last modification should be included or clearly referenced through a link (you will hardly ever find a book without its authors's name in it!).
- *Aesthetic correction:* Fine correction of the document is impossible for the author as the actual layout is not known. Browsers would have to fulfil that task, but the aesthetically correct layout of a document is a problem that sometimes even systems like T_EX do not solve correctly.

Layout Rules for Kiosk Systems

Fortunately, when writing pages for a Kiosk system, the final appearance of the document is usually predictable even when using a logical markup language like HTML, because the systems that will receive and display the document are well defined and in practice all of the same type (screen size, etc.). Applying our layout steps yields the following:

1. *Page format:* In contrast to WWW pages, it is usually not acceptable for kiosk systems to require the user to scroll through pages. Firstly, a possibly inexperienced user would be distracted from the task, and secondly, many kiosk systems only have primitive input devices that do not allow for "meta actions" like scrolling which have nothing to do with the actual page contents.

2. *Page layout:* As the target browser is usually known, it becomes possible again to define margins, etc., so the rules for hypertext documents apply.

3. *Page pattern:* Headings are as important in kiosk pages as

they are in WWW documents. The task of making the document structure clear to the user is simplified through the limited domain that a kiosk system usually covers.

4. *Fonts and Styles:* The display of the target browser can be influenced. The style sheets of HTML3 offer some additional functionality in this respect.

5. *Contents:* As with WWW pages, the user diversity (though usually not world-wide) plays an important role. The contents have to be understandable and accessible by a broad variety of users. In contrast to WWW pages that are often visited primarily by "net surfers" (i.e., by experts, and for fun), many customers of public kiosk systems will be inexperienced computer users, but have some more or less urgent task to fulfil.

6. *Aesthetic correction:* As far as the target display allows, this is possible, and also necessary. An aesthetically appealing layout is crucial to attracting users, and to keeping them motivated throughout the session.

A Classification of Kiosk Systems

Further rules can only be identified if the task of a kiosk system is defined first. We propose a classification of kiosk systems, according to their major task.

- **Information Kiosks** These kiosks have the primary goal to provide information in a usually limited subject field. Examples are kiosks at railway stations where a user can find a train connection for a chosen destination. Users that come to those systems are motivated already because they need some information. However, they will usually give up if they do not succeed in finding it quickly.

  In those systems, the pages should be laid out clearly, with constant parts and a stringent user guidance, so that customers do not have to use the system longer than necessary. Their design should try to *minimize* average session duration. Text elements may be dominating the appearance of pages if necessary to provide the information requested.

- **Advertising Kiosks** Advertising kiosks are installed by companies or institutions to present themselves or their products to the public in an attractive and innovative way. The missing initial motivation of potential users has to be compensated for by a visually attractive design, especially of the start page. Furthermore, the longer the user stays in the system, the higher its success will be in the eyes of the provider.

  For that reason, it makes sense for subsequent pages to present new visual stimuli, e.g., by changing to a different overall layout and appearance. The contents should be interesting, entertaining, and keep up suspense so that the user is motivated to explore it further. User guidance should be less stringent. In short, design should try to *maximize* average session duration. Graphical elements should dominate the layout, with only as much text as necessary (people prefer "pushing" a nicely drawn start button to just clicking on a highlighted word saying "Start".)

- **Service Kiosks** These systems are similar to information kiosks, with additional emphasis on information flow from the customer to the system. Examples are hotel reservation systems, where the user may have to enter certain data (like name and address) to book a room, and this information is passed on to the hotel. The user interface contains more complex physical or virtual input devices (like a keyboard) that have to be integrated into the system without affecting stability, security and ease of use. User guidance always has to make clear what the customer is supposed to enter in a dialog, to avoid frustration.

  Motivation is comparable with information kiosks, although success expectations will usually be even higher as the customer has a clear task in mind ("book hotel room") and hopes to complete using the system. Design should again try to *minimize* session time.

- **Entertainment Kiosks** These kiosks finally do not have a specific task apart from entertaining the user. Installing such kiosks might not appear feasible at first, but it may increase consumption in bars or similar places. They might also prove a useful new concept to help people pass the time, e.g., in waiting rooms.

  Layout has to follow the same rules as for advertising kiosk systems or those encountered in the area of video games. Initial motivation is curiosity which the system has to create first. Session duration should be maximized (within reasonable limits). The interface has to be visually appealing *and* simple at the same time, which often leads to design conflicts.

Of course, most systems will belong to two or more of the above classes. Many kiosks that supply information or services also have the goal to advertise the provider. There will, however, usually be a primary task which should be identified, if not for the whole kiosk system, then at least for its individual pages.

HTML for Kiosk Systems: Status and Trends

HTML 2.0

HTML (Hypertext Markup Language) was initially designed to facilitate cooperative working at technical documents [Berners94a], and has since become the standard document description language of the World-Wide Web. With the idea of creating on-line, networked kiosk systems, using HTTP as protocol and HTML as document language suggests itself.

However, HTML is based on SGML, the Standard Graphics Markup Language [Goldfarb94], and is not really designed to specify layout, but rather a logical document markup (structuring) language. As mentioned above, transforming those structures into a layout is the task of the Browser. For example, the exact positioning of pictures on a page cannot be specified in HTML unless the browser is known in advance, and even then arbitrary positions of the picture are not possible.

But apart from this general problem, the current standard of HTML, version 2.0, has several other major drawbacks:

- Tables are not part of the standard,
- colors for arbitrary elements cannot be specified,
- underlaying pages with background images is not possible,
- page layout (two-column text, etc.) cannot be defined.

HTML 3.0

HTML development continues, however, and after the intermediary definition of HTML+, the specification of HTML 3.0 is now in its last stages [Raggett95]. For the design of kiosk systems, it offers the following improvements:

- Tables and justification as therefore page layout (two-column text, etc.),
- Background images,
- Inline images with many different, arbitrarily shaped sensitive areas,
- Concrete layout specifications via cascading style sheets.

Those improvements abolish the need, for example, to create HTML pages consisting of a single sensitive image creating a server-client round trip for each event. Special layout wishes can be implemented using HTML itself, or custom style sheets. At the same time, HTML itself remains largely a logical markup language, since style sheets are not part of the language itself but special instructions that can be obeyed by certain browsers, but ignored by others.

Nevertheless, even HTML 3.0 still lacks some functions that are desirable for kiosk pages:

- Even with style sheets, it is not yet possible to assign colours to arbitrary parts of the document.
- Inline videos cannot be displayed yet (showing videos in an external viewer is not suitable for kiosk systems).
- Time-based control is not supported yet.

It has to be said, however, that the specification of HTML 3.0 is not complete yet, and some of the above shortcomings might still be eliminated. For example, partial support for *HyTime* in HTML 3.0 is currently being discussed.

Alternative Approaches

Hyper-G

Hyper-G [Kappe91] is a system to manage networked information, much as the World-Wide Web with HTTP and HTML, and compatible with those standards. The main difference is the way links are stored. While HTML documents contain them as part of the markup, Hyper-G stores the original document and its links separately in its document database. This means that different views on the same document can be created without copying the underlying document, and it is easy to add your own links to a document you find somewhere in the web.

However, most of the above criticism of HTML 2.0 applies to Hyper-G, too. Tables, formulae, inline videos and time-based control are not supported, and the use of inline images is limited to the functionality found in HTML 2.0. Thus, even though Hyper-G contains several very promising approaches, it currently is not more suitable for the design of kiosk documents than HTML. In addition to this, Hyper-G is less widespread, so third-party development might neglect it. Nevertheless, the system is used by some publishers for the creation of electronic documents, and it is surely worth keeping an eye on its development.

HyTime

HyTime (for "Hypermedia/Time-based Document Structuring Language"), like HTML, is a standardized infrastructure for the representation of "hyperdocuments" for integrated open hypermedia applications [Goldfarb91]. It has been published as ISO 10744.

Like HTML, HyTime is based on SGML but, as the name suggests, it tries to solve two problems together: structuring hypertext documents, and managing the presentation and synchronization of multimedia information. This is accomplished by seeing them both as just two different applications of the same principle: addressing objects. Using this approach, HyTime offers concepts for the following tasks:

- associating objects with links,
- placing and interrelating objects within space and time,
- structuring documents logically,
- including non-textual data in a document.

HyTime-compliant systems do not have to support all of the above features. To make partial implementation easier, HyTime's features are distributed into six modules, e.g., for hyperlinks, time-space scheduling, rendition, etc.

Although HyTime is specified in the form of an SGML DTD, it is not meant to be a complete architecture for hypermedia systems. It is rather an enabling "meta standard" that should be applied when designing concrete hypermedia architectures and applications.

For kiosk systems, HyTime offers several benefits: support for time-based control, and a refined model for the spatial arrangement (layout) of objects. However, *the* HyTime system does not exist, although support for some parts of HyTime is being discussed for HTML 3.0, as mentioned above.

Conventional Authoring Tools

Most of the systems presented have shown some shortcomings in the area of layout control and animation. Traditional authoring tools, like MediaStyler, or ToolBook, are a good "reference" for those features. By examining those systems, several advantages become clear. They offer fading effects, inline videos, and a free graphical layout with many dialog components already available. On the other hand, they are not usually capable of dealing with networked documents, and they are mostly available for MS-Windows systems only.

Future Requirements

With these observations in mind, we can put up a list of requirements against which future authoring systems or languages for online kiosk systems will have to be checked, whether they are extensions of HTML, or new systems designed from scratch:

- Full layout control: Positioning, fonts, styles
- Arbitrary colour assignment
- Inline images and videos
- Object animation and effects (fading, etc.)
- Time-based control
- Real-time response through local processing where necessary.

Summary

Layout has shown to be crucial to attractive, information-conveying kiosk systems. Guidelines for good layout have to be drawn from many different areas, particularly Gestalt psychology, conventional typography, human-computer interaction, and graphical user interface design. These fields are complex enough to justify building a team of experts from each of these fields when creating professional kiosk documents.

However, hypertext as a new medium requires adapting many rules, especially those taken from conventional typography, to the fact that information has to be presented in small chunks.

For networked, on-line kiosks, HTML can be used. Current versions have shortcomings regarding layout control, but this will improve with HTML 3.0. Alternative systems exist, but the ideal system for the authoring of documents for on-line kiosk systems has yet to be designed.

References

[Berners94a] Berners-Lee, Tim: *The World-Wide Web*, CACM, Vol. 37, No. 8, August 1994

[Berners94b] Berners-Lee, Tim: *Hypertext Markup Language Specification, Version 2.0*, Internet Draft, 1994 (http://www.ics.uci.edu/pub/ietf/html/html2/)

[Goldfarb91] Goldfarb, Charles F.: *HyTime: A standard for structured hypermedia interchange*, IEEE Computer, August 1991

[Goldfarb94] Goldfarb, Charles F.: *The SGML Handbook*, Oxford University Press, Oxford 1994

[Gulbins93] Gulbins, Jürgen: *Mut zur Typographie*, Springer-Verlag, Berlin 1992

[Holfelder94] Holfelder, W., and Hehmann, D.: *A Networked Multimedia Retrieval Management System for Distributed Kiosk Applications*, in: Proceedings of the 1994 IEEE International Conference on Multimedia Computing and Systems, May 1994

[Kappe91] Kappe, Frank M.: *Aspects of a Modern Multi-Media Information System*, dissertation, Graz University of Technology, Graz 1991 (http://hyperg.tu-graz.ac.at/)

[Koehler29] Köhler, Wolfgang: *Gestalt Psychology*, Liversight, New York 1929

[Raggett95] Raggett, Dave: *HyperText Markup Language Specification, Version 3.0*, Internet Draft, 1995 (http://www.hpl.hp.co.uk/people/dsr/html/CoverPage.html)

[Sassoon94] Sassoon, Rosemary (ed.): *Computers and Typography*, Intellect, Oxford 1993

[Schmitt94] Schmitt, Alfred: *Mensch-Maschine-Dialog I*, lecture notes, University of Karlsruhe, 1994

[Shneiderman92] Shneiderman, Ben: *Designing the User Interface*, Second edition, Addison-Wesley, Reading, MA, 1992

About the Authors

Jan Borchers received his master's degree in computer science at the University of Karlsruhe this year. His diploma thesis dealt with hypermedia documentation of program code. Now he is working at the Institute for Computer Science of the University of Linz.

His interests are dialog systems, multimedial dialogs and graphical user interfaces.

Institute for Computer Science
Telecooperation Group
Altenbergerstr. 69
4040 Linz, Austria
email: jan@tk.uni-linz.ac.at
http://www.tk.uni-linz.ac.at/

Oliver Deussen graduated in computer science at the University of Karlsruhe in 1991 and is now working on his doctoral thesis at the Institute for Operation and Dialog Systems of the University of Karlsruhe.

He is interested in simulation and modelling aspects of computer graphics and the optimized design of graphical user interfaces. He also likes efficient and easy to use algorithms of computational geometry.

Institute for Operating and Dialog Systems
University of Karlsruhe
76128 Karlsruhe, Germany
email: oliver@informatik.uni-karlsruhe.de
http://i31www.informatik.uni-karlsruhe.de/

Clemens Knörzer studied at the University of Karlsruhe and received his master's degree in 1992. Now he is working at the "Zentrum für Kunst und Medientechnologie" Karlsruhe (ZKM) in a project called "Forschungsverbund Medientechnik Südwest", where he is designing a multimedial video database. He is also interested in data compression and multimedial user interfaces.

Institute for Operating and Dialog Systems
University of Karlsruhe
76128 Karlsruhe, Germany
email: knoerzer@informatik.uni-karlsruhe.de
http://i31www.informatik.uni-karlsruhe.de/