VISUAL ALARMS: THE ALARM CONTROL DESKTOP - A COMPONENT OF A BUILDING AUTOMATION SYSTEM

Walter-Alexander Jungmeister
Visiting Research Scientist/Human Factors Project Manager
Human-Computer Interaction Laboratory, University of Maryland,
College Park, MD 20742 USA and
Union Bank of Switzerland, Dept. BERG, P.O.B, CH-8021 Zürich, Switzerland
Email: Alexander_Jungmeister@ZH007.ubs.ubs.arcom.ch

ABSTRACT

Task analysis for alarm monitoring tasks indicate that new ideas for more efficient user interfaces are desirable. Novel interface features like the active data point metaphor combined with advanced hierarchical screen layouts and translucent windows have lead to promising prototypes. Their evaluation with usability testing has shown, that improved tools for large and complex information displays in building automation systems are a clear necessity. This article discusses details of five experimental prototypes.

KEYWORDS: Buiding automation systems, specific desktops, active point metaphor, translucent windows.

1 INTRODUCTION

In most large companies, e.g. banks or warehouses, there is an ever increasing need to monitor a huge array of security and alarm devices. These alarm devices have sensors, which send status information in case of emergency or upon request - hundreds or thousands of sensors may be centrally monitored. These sensors ("data points") are hierarchically organized in many levels, corresponding to groups, sections, departments, floors, buildings and zones etc.

Usually these alarm devices have a central processing unit, connected with a widearea network, which allows immediate alert transmission to a computer display (e.g. line printer, terminal) or a conventional button panel. The computer screens show mostly characters on an 80x25 lines alphanumeric terminal display, and are focusing on the task of the maintenance specialist, not the security guard.

Only some newer systems offer semigraphical maps (a few advanced systems show them as real graphics in windows[1]), which allow exploration or give graphical feedback. However, these graphical systems still show slow accessible menu techniques, which are then clicked with a mouse instead of pressing an F-Key.

The power of graphics and visualization is still untouched. Typical ergonomic problems of those alphanumerical and graphical representations are:

- Screen space not used efficiently.
- Operators lack overviews and become lost in detail.
- Disorientation arises, because of too many page turns (in window systems page turns are replaced by overlapping windows, causing “window clutter”).
- There are media breaks (manual lists, additional drawers with maps etc.), which cause unnecessary time delays.
- Users often don't use/like graphics because access is not quick enough and seems to be unprecise.
- If multiple alarms of the same type come up, there is no visual clue indicating which alarm is to be handled first.

2 PROBLEMS OF SECURITY GUARDS AT WORK

Current designs produce a cluttered desk with many different systems to use for the security guards. So they are very often they unable - especially in emergency situations - to manage the rarely used systems and perform the required responses.

When an alarm comes up, the security guard has to quickly determine what kind of alarm it is, what priority it has and respond quickly. Normally the alarms are short - mostly cryptic - text strings with only a brief hint to the source of the alarm or just a numbered, highlighted button on an enormous panel. The guards have to read the alarm number, consult a manual for security instructions, match up the device number to a room map and locate the origin of the alarm. Afterward they have to find the appropriate key for the room and visit the location of the alarm.

The last step is to decide whether to escalate (inform superiors or internal or external specialists) or simply reset the alarm. This is usually done by pressing a button or F-Key on a computer keyboard. All this should be done in a few minutes. Confusion will normally arise, if many alerts at different locations (perhaps of the same priority) occur at the same time. As a result very often alarms are not handled correctly, are canceled due to overload or sometimes even automatically call fire trucks or police cars, which must be paid, if called without proper reason.

Alarm systems are complex and therefore require careful evaluation of special user interfaces. Research on this topic, however, is rare and suffers from the fact, that most systems are built without connections to the user interface research community. But the interest in human factors issues for alarm systems is growing and will increase with the extension of system functionality and complexity. Research on complex systems and visualization are both relevant for current alarm system design.

3. "HIERARCHYPLANES": THE EXPERIMENTAL PROTOTYPES

These problems have lead us to the development of some experimental prototypes, to explore better solutions to the given ergonomic problems. For quick results - our focus was on user interaction design - an Apple Macintosh/ SuperCard 1.6 environment was chosen. The HierarchyPlanes prototypes were developed as standalone applications without runtime modules.

3.1 Components of an Alarm Control Desktop

Each of the five developed prototypes has the same basic set of features, except for the alarm visualization and hierarchy grouping information. These basic features consist of a direct manipulative area which allows exploration of active data points (status information of color and shape coded devices) and a control area with buttons and fields, which give additional precision text information *(figure 1)*. The buttons are large enough to be easy targets[2]. They allow to pop up either translucent or opaque windows, where detailed information such as maps or the previous history of alarms can be seen. The alarm history presents itself as manageable text strings for acknowledging, postponing or routing of alarms.

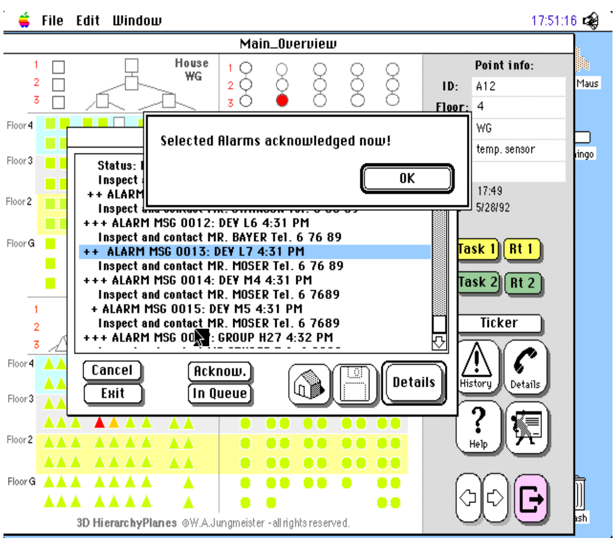

Figure 1: The Alarm Control Desktop,with the gray control area at the right, the active data point area at the left and an (here opaque) alarm history window.

3.1.1 The active data point metaphor

To ensure ease of use it is useful to rely not only on how the data are structured, but also on the users previous mental models - here this is the conventional button display wall. Backward *compatibility of mental information* representation helps to increase speed of learning while reducing cognitive load. This approach is best achieved if the data can be shown as points and are presented as groups of virtual "button" panels in allusion to the traditional button panels with lights, yet with enhanced functionality (active clickable and responding buttons). With the active data point metaphor it is also easier to achieve *consistency*: only one user interaction technique (clicking on points) is necessary for accessing all of the information of all security devices. The active data points also allow a better *overview* over the huge amount of data points and their logical structure (hierarchies).

Easy to understand color-, size- and pattern *coding techniques* (according to known ergonomic rules) are important to guide users intuitively without requiring too much mental transformation (cognitive overload). Our implementation of the active data point uses color coding for status information. Red means alarmed status, orange a serious defect, green is normal. *Shape coding* is used for device type information (door detectors are represented by triangles, while smoke detectors are squares etc.). *Pattern coding* is used for displaying results of a graphical query. Coding through *grouping* with white space and border lines allows easy recognition of higher levels or hierarchies (virtual points).

3.1.2 Active graphic exploration and search

Flexibility: Information can now be viewed and handled alternatively through maps, lists and panels of points. So personal preferences of the security guard or work group can be matched easily according to their individual work style.

Active *exploration* of the system status is as easy as moving a mouse pointer over the data points receiving immediately feedback in an text field area which gives information about the name, the status, the location etc. of the data point. Connections to neighbor points, the reality on floor and hierarchy remain intact and don't change, providing a stable "world" of view for the user instead of ever changing lists. If *active search* of some data points through specifying search parameters is required, our prototypes have the ability to do this and present the found data points through a special graphic pattern.

3.1.3 Direct manipulation

Direct manipulation (as defined by Shneiderman 1983) combined with browsers (see Chimera 1991, Shneiderman 1991) can help with the understanding of related or *connected problems* by displaying them in context. Only if the interdependency of data points, status information and time related information are transparent, is it easy to make a quick decision about which information is important if multiple alerts arise.

The prototypes have the ability to integrate all necessary information (though not all information sources are implemented yet). It is no longer required, that multiple *media breaks* (as before with the drawer, maps, lights, text lines, lists) slow down the task of information gathering and exploring. There is only one media (the computer display) with only one handling philosophy, the mouse click (of course alternative keystrokes can be assigned, if the users prefer that).

Learnability: A context sensitive help window explains the coding scheme of the data points to reduce cognitive load and to decrease media breaks. It is important, too, that all necessary information is accessible through the system. Therefore a context sensitive help system and clear labeling are a necessity for an integrated alarm handling system.

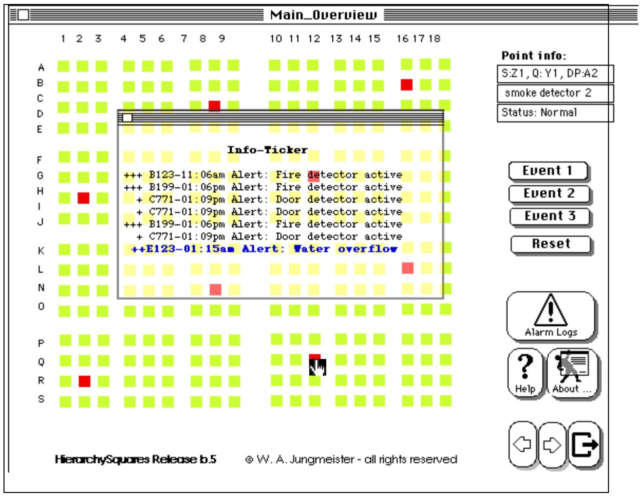

Figure 2: Translucent window over point panel

3.1.4 Translucent windows

To mimimize disorientation problems (getting lost in window clutter or hierarchies, extensive paging or deep menus) it is very important to show as much data as possible on one single screen. So we tried to organize a large array of data points in overlapping planes. Translucent data points can increase the number of points that can be shown at once. This was inspired by "Clearface", overlapping translucent video screens, Ishii invented for collaborative drawing at NTT in Japan (Ishii 1991).

The problem with this approach is that overlaid structures of the *same type* cause perception problems, - is it often not clear, which point or group belongs to which data level. This is because of some data points and grouping information like white space is hidden from others.

Translucent overlapping works well, however, if the structures are *completely different* (e.g. square panel - text). Therefore we created a scrollable alarm history window *(figure 2)*, with containing text information - text has a different texture than data points - which can be layered on top of the data point area. The window can be made translucent (optional) to increase the density of information shown at once. watched. The operator can see so the color changing status information of the alerted data points through the text information.

3.1.5 Innovative alarm routing visualization

Data points are usually divided in two groups**:** *actual data points* which have more or less direct connections to actual sensors in devices and *virtual points*, which are basically points without a direct connection to a sensor defined by an operator or system designer. Virtual points are usually set to alarm to show the increased importance of attendence of a group of defect or alarmed sensors. Efficient visualization schemes for grouping important alarm information are mostly neglected in todays building automation systems, yet are one of the most important parts of those systems.

We set up seven different prototypes with different alarm routing visualization techniques, to evaluate the optimal routing. All of our prototypes we built have translucent windows and the same active data point metaphor.

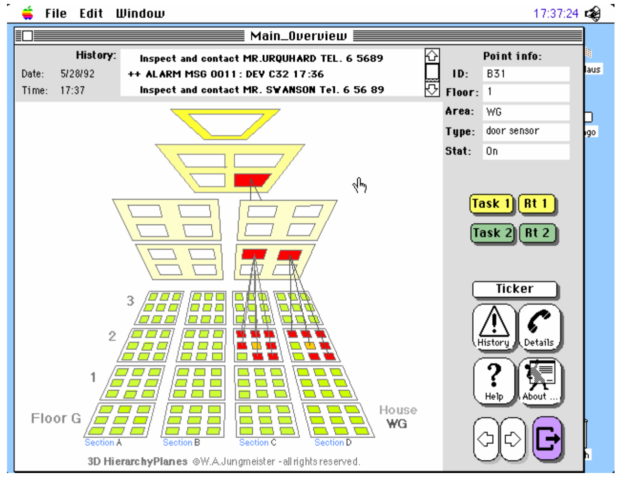

Figure 3: 3D Hierarchy Planes

3.1.5.1 3D HierarchyPlanes

To avoid information cluttering and increase the perception speed of different type of information (actual vs. virtual data points), we projected the alarm level hierarchy with the virtual point (group information) planes in a 3 D view from the side, one level over each other *(figure 3)*. Each level represents the priority of the alarms reached, the highest level means the most urgent alarms. As with the previous prototypes each actual data point can have a different status - a range from normal to alarm, represented trough color coding. The virtual data points, the groups are now presented as points too, but above the level of actual points. The heuristic rule decides, if there is a certain level of serious defects in a group of data points (for example more than 50% of a group) then the next level of points is activated and so on. A given set of rules implemented in the algorithms will be activated, if there is competing information, for example two areas are highlighted with the same defect priority, and set a visual signal (flashing and deeper color) which alarm area has to be treated first.

This prototype is especially graphically appealing and the alarm visualizing and the priority of the alarm are very easy to check out. But the presentation still has some drawbacks:

- The 3D view is limited to a certain amount of data points (ca. 150 data points on 13" Monitor, ca. 600 dp on 19"monitor, because of the 3D projection (waste of space)
- Labeling can become difficult, if it has to be projected into 3D.

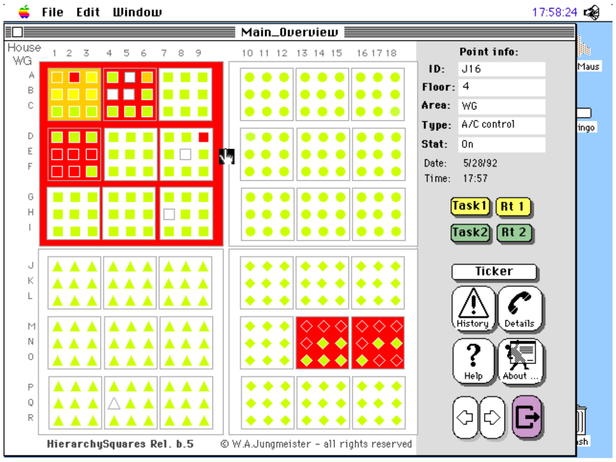

Figure 4: 2D HierarchyPlanes with highlighted areas

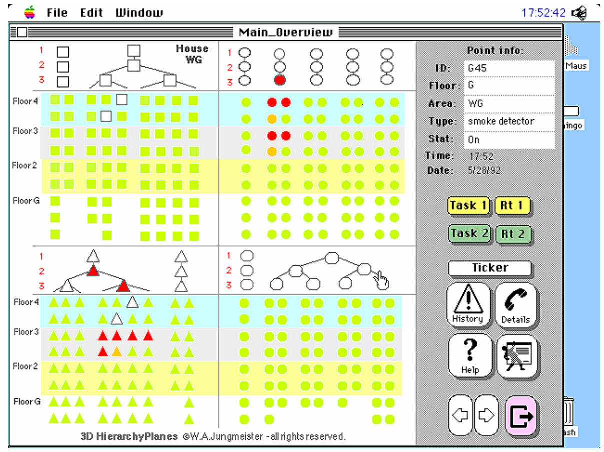

Figure 5: HierarchySumwith summary panel on top

3.1.5.2 2D HierarchyPlanes

To increase the density of data points on screen again, we set up a new plain data point panel, HierarchyPlanes *(figure 4)*. Shape coding of data points indicates as before the kind of device, while color coding shows the status. As with the previous prototypes there is no direct match between the original location of the device on the floor and their representation on screen, but the devices are grouped logically (all heating devices, all smoke detectors together).

Instead of increasing the alarm priority with area (size) coding with bigger points above it was created with group area coding below the actual data points. In case of escalating alarms these group areas (virtual data points) are popping out and show so the level of increased importance.

This prototype has a second variant, called Hierarchy Sum *(figure 5)*. For the increasing alarm status we designed a prototype with a different scheme for the alarm hierarchy routing. The basic idea behind it is a field above the data point panel, which is functioning as a "summary light" for increased alarm priority.

Grouping different sets of points are so very easy to set up and it is very easy to write algorithms to automate the point layout out of the data structure and status.
With the additional functionality, it is easy to check out with the prototype, what to do in every situation. One minor problem occurs, nevertheless.
It is required to perform a mental transformation (increasing cognitive load) between the data points - located in devices - in real world and their abstract representation on screen. The security devices, mounted on designated walls, ceilings etc., are usually carefully laid out on a map and have to be found at that exact location in case of emergency.

This information transformation could be problematic if a certain task requires an exploration of interconnected problems.

3.1.5.3 FloorPlans

This issue forced us to give up the logically layout in favor of a more map oriented design to match more mental compatibility to the outside world. This led to the FloorPlan prototype. Shape and color coding and alarm routing are done the same way as in the HierarchyPlanes, the only difference is that the layout of the planes is a condensed and stylized floor layout as it can be seen in *figure 6*.

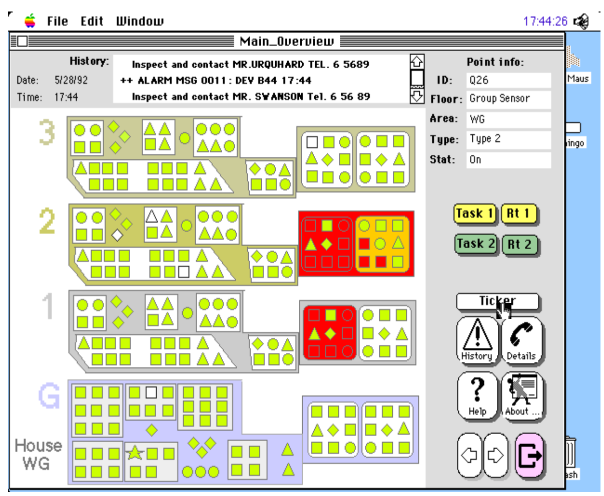

Figure 6: Floor Plans

The FloorPlan prototype has several advantages. It is providing a better help for interconnected problems between different types of devices, as they are displayed in their actual order according to the real world. So for example if three smoke detectors are alarmed around a treasury sensor, it is easy to foresee down future troubles with the treasury sensor. In the HierarchySum version, where the point layout is done logically for each type of device this task is more difficult. Another strong point is the better general orientation in the building, where the panels actually mean floors and sections of the building and are not an abstract figure. As a disadvantage it may be considered that creating individual floor layouts manually for each building is costly, time consuming and probably cannot be automated by a smart layout algorithm.

So this final approach of floor planes seems to be a good compromise between space efficient point grids and space consuming building maps with data points projected on them.

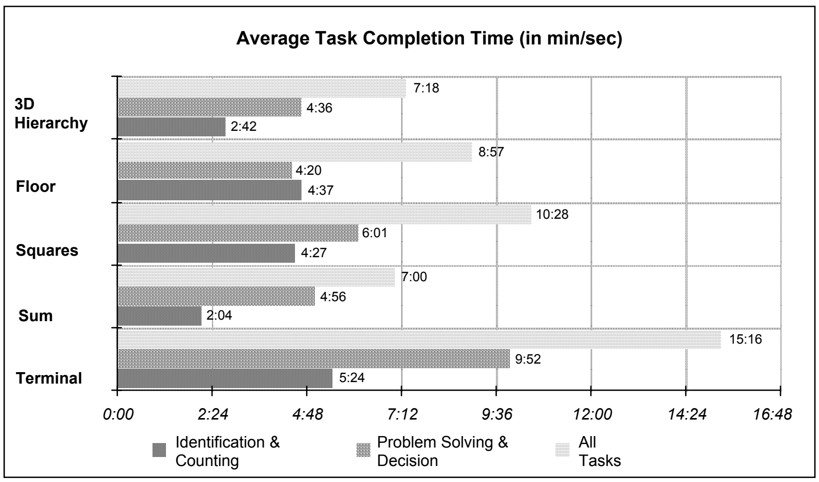

Figure 7: Usability test - average task completion time

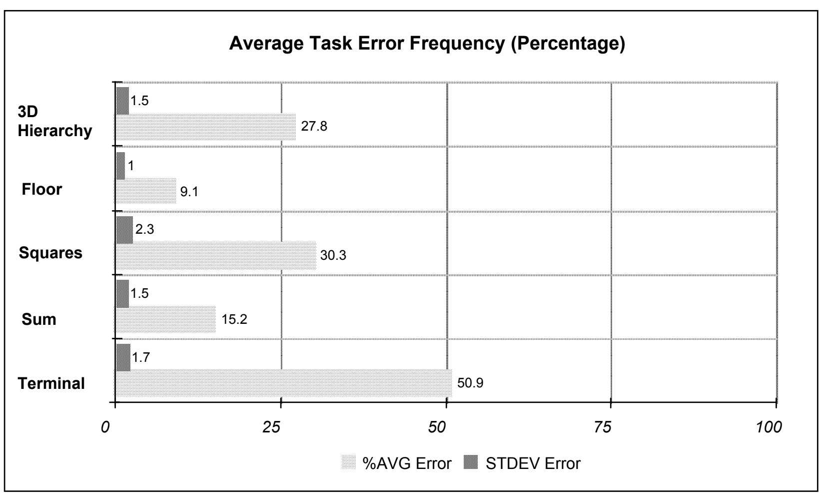

Figure 8: Usability test - average task error frequency

4 USABILITY TESTING

4.1 The usability study

To find out which of the four major prototypes (3D HierarchyPlanes, 2D HierarchyPlanes, HierarchySum, FloorPlans) fit the given task best and compare to the conventional method with media breaks and dumb host terminals, we set up a usability study. First we created another prototype mimicking the traditional character terminal solution. Next we prepared a set of standard tasks, which had to be performed by all test persons after an initial training session. We separated two distinct levels of task difficulty to observe the efficiency of our prototype designs on each level individually:

Identification and counting tasks (level 1)

Problem solving and decision tasks (level 2).

The users were students of different levels and never had any contact before to a building automation application. We rated this as important, because we know about effects from previous usability testings, that users who have used a certain system for a long time sometimes tend to rate innovations negatively and therefore the results would be probably biased. To draw a complete picture, however, a second follow up test with security guards should be performed.

4.2 Findings

It showed clearly, that the character terminal prototype needed the most time to complete all tasks. Compared to all other prototypes, nearly double the time was necessary to complete all tasks! The standard deviation of the completion time hints to the conclusion, that the terminal solution also has the biggest variation, meaning it is difficult to get stable results from that prototype. It was the one, too, which had the highest error rate, nearly five times as high as the best performing graphic prototype. This means practically, that every second task was performed wrong with this version. For the graphical user interfaces the HierarchySum prototype was the quickest, the FloorPlans a close second and the HierarchySquares was the slowest.

Concerning the quality of the results (low error rates), the HierarchyFloor prototype performed definitely best with the most constant variation. HierarchySums gained a close second, while the 3D Hierarchy and the HierarchySquares performed distinctly worse than both of the top two. FloorPlans was the prototype the users like most to work with it and judged it the most graphically appealing *(see figure 7, 8)*.

Summarizing the results, FloorPlans is the overall winner, if constant results, low error rates overall and in complex tasks are more important than raw performance over time.

FloorPlans seems best for complex problem solving and decision tasks, while HierarchySums is a close second there and has its strengths for quick identification and counting tasks.

5 CONCLUSION AND OUTLOOK

The usability study showed clearly the advantage of graphical displays - especially of the condensed floor maps - for both error rates and task completion time. This encourages us to continue our work with further expansion and refinement of the basic concepts and prototypes. Yet an array of future improvements is necessary to cover open questions of our prototypes, too:

- *Extendibility* and better *handling* of data point status information (sorting, resetting, adding and deleting of points). For example a selection marquee tool could be provided to gather data points and disable them by mouse click or move them to another panel, if this is required.
- *Translucency control:* Our translucent windows showed a fixed setup of translucency. For improving the usability of the translucent windows should have a control button for incremental changes of their translucency.
- More research on advanced data *filtering techniques* (only included criteria are displayed as data points, data groups, sections, departments, etc.) and *dynamic and static queries* (highlighting only the desired data while the rest is grayed out) seem to be a necessity, too. Queries and filters are especially useful for emergency situations, when problems should be immediately investigated.
- *Better customization:* The screen layout should be more user definable. A screen split or multi windowing for comparison purposes could be helpful well as changing the display order or choosing variations of colors or patterns.
- *Display limitations:* As our prototypes are only developed for 14" screens they are easily to expand for 19" and 21" monitors without changing the basic concepts. We believe, indeed, that the bigger display is a necessity and makes the best out of our prototypes. But there are still some upper limits for displaying a certain number of data points, because shape and color differences are difficult to distinguish if the points are too small. Here clearly an opportunity for extended research can be found.
- *Great potential:* We think that the active data point metaphor combined with translucent overlay structures has a great potential for future development in other applications areas. A promising future for that kind of graphical information systems lies ahead, but there is still the need for the creative mind to enrich and empower these concepts to optimize their performance to human factor requirements.

6 REFERENCES

1. **Chimera, R.; Wolman, K.: Mark, S., Shneiderman B.**: Evaluation of three interfaces for browsing hierarchical tables of contents. University of Maryland, Technical Report CAR-TR-539, CS-TR-2620, March 1991

2. **Ishii, H. and Kazubo A.:** Clearface: Translucent Multiuser Interface for Teamworkstation. In: SIGCHI Bulletin volume 23, number 4, Oct. 1991

3 **DELTA XO** System Guide**:** Sensor & System Development Center, Honeywell, MN 55418, Minneapolis 1991

4. **Metasys** Operator Guide, Systems and Service Devison, Johnson Controls, P.O.B. 423, Milwaukee WI 53201-0423, Milwaukee 1991

5. **Shneiderman, B**.: Direct Manipulation: A Step Beyond Programming Languages. IEEE Computer, August 1983.

6. **Shneiderman, B**.: Visual user Interfaces for information exploration. University of Maryland, Technical Report CAR-TR-577, CS-TR-2748, July 1991

6. **Tognazzini**, **B.**: TOG on Interface. Reading, MA 1992

---

[1] The METASYS System (Metasys Operator Guide, Johnson Controls, Milwaukee 1991) shows them in an PC MS-Window environment. The DELTA XO System (Delta XO, Honeywell, Minneapolis 1991) presents its information in an Unix OSF/Motif environment.

[2] According to Fitt's law (see Tognazzini 1992, pp. 201)