Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Int. J. Human-Computer Studies 63 (2005) 565–586
International Journal of Human-Computer Studies
www.elsevier.com/locate/ijhcs

Ecological interface design and computer network management: The effects of network size and fault frequency

Pierre Duez, Kim J. Vicente*

Cognitive Engineering Laboratory, Department of Mechanical & Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Canada M5S 3G8

Received 18 August 2003; received in revised form 7 April 2005; accepted 2 May 2005
Available online 5 July 2005
Communicated by S. Wiedenbeck

Abstract

This article describes an experiment investigating the impact of ecological interface design (EID) on human performance in computer network management. This work domain is more dynamic than those previously studied under EID because there is a constant potential for the addition and removal of devices, as well as changing configurations, making it important to study the generalizability of the framework. Two interfaces were created for the University of Toronto campus network consisting of 220 nodes: a P interface based on existing design practices which presented primarily physical information and a P+F interface based on EID which presented both physical and functional information identified by an abstraction hierarchy analysis. Participants used one of the two interfaces to detect and diagnose faults or disturbances in the simulated network in real-time. Network size and fault frequency were both manipulated as within-participants variables. The P+F interface led to faster detection times overall, as well as improved fault detection rate and more accurate fault diagnosis under higher fault loads. These results suggest that the EID framework may lead to more

*Corresponding author. +1 416 978 7399; fax: +1 416 978 3453.
E-mail address: vicente@mie.utoronto.ca (K.J. Vicente).
URL: http://www.mie.utoronto.ca/labs/cel/.

1071-5819/$ - see front matter
doi:10.1016/j.ijhcs.2005.05.001

robust monitoring performance in computer network management compared to existing interfaces.

Keywords: Ecological interface design; Network management; Abstraction hierarchy; Fault diagnosis

1. Introduction

Computer networks span the globe; they are present in growing numbers of workplaces, at universities, in schools and in homes. Even companies that are not connected to the Internet are likely to possess internal networks to connect their computers to each other, as well as to shared resources (such as printers, file servers, etc.). In a survey by Lucent Technologies NetCare (Blum and Kaplan, 2000), 80–90% of companies surveyed were found to use a Network Operations Centre (NOC) to administer their local area networks (LANs) or wide area networks.

Internet protocol (IP)-based networks are networks of computers (loosely defined here to include switches, hubs, printers, etc.) that communicate over IP (Leon-Garcia and Widjaja, 2000), a protocol used by most LANs. As these networks grow, it becomes important to monitor them and to remedy any problems (e.g., faulty equipment, resource exhaustion, etc.) before the effects are too widespread. As the errors propagate, it becomes more difficult for network managers to deal with the increasing demands. If these problems are not caught, they can often spread to the point where network performance suffers, causing users of the network to be unable to access remote resources, or communicate with other machines. This, in turn, can result in worker frustration, a loss of productivity and—in more extreme cases—disgruntled clients and lost revenue. For example, the New York Stock Exchange fined TD Waterhouse $225,000 (US dollars) "for problems related to Web Site failures, which led to an inability to fill online stock orders." (Toronto Star, March 1, 2001, p. C2)

As the growth of networks is a fairly recent phenomenon, the tools to assist in network management and supervision are also new. However, while management tools for networks make it easier for a network administrator to oversee the functioning of the system, these tools are increasingly being created in-house, based on commercially available platforms (Blum and Kaplan, 2000), and are assembled and configured under severe time constraints. The amount of planning and design—and attention to human–computer interaction issues—that might therefore go into them is minimal.

The objective of this research is to determine whether ecological interface design (EID) can serve as a valuable tool in the development of improved interfaces for computer network management. EID is a framework developed by Vicente and Rasmussen (1992) for the design of human–computer interfaces for complex systems, whose primary aim is to support worker adaptation to novel or unexpected situations (see below for more detail). EID was chosen as a potential solution to interface design within this domain because of promising results in other complex work domains, such as power plants, petrochemical plants, airplane cockpits and medicine (Vicente, 2002). Here, *complexity* may be defined as a measure of the number of variables in, degree of coupling within, or the rate of change of a system.

The complexity in the domain of computer networks arises from the very dynamic nature of these networks—not only in terms of state and performance, but also structure. First, faults and disturbances can happen much more quickly and frequently than in systems with slower dynamics, such as nuclear power plants. Second, whereas the equipment available to operators in other work domains, such as aviation, is relatively constant, the number of nodes, the connections between them, and the functional consequences of these can change more frequently and with a greater degree in computer networks. For example, at an institution such as the University of Toronto, computers might be added to the network by research laboratories, administrative offices, etc.—all without central guidance or approval. This changing work domain structure thereby poses relatively unique challenges, at a design level requiring a strong need for expandable or reconfigurable interfaces, and at an operational level requiring a strong need for operator adaptation to novelty and change. Therefore, it is important to investigate the generalizability and applicability of the EID framework to these characteristics.

1.1. Ecological interface design

EID grew out of the research program conducted in the Electronics Department of Risø National Laboratory beginning in the 1960s (Vicente, 2001). That research led to, among other things, two conceptual tools—the abstraction hierarchy (Rasmussen, 1985), and the skills, rules, knowledge (SRK) taxonomy (Rasmussen, 1983)—that serve as the theoretical foundations of EID. The *abstraction hierarchy* is a framework that can be used to develop models of particular work domains. It can be defined as a stratified hierarchy characterized by a structural means-ends relation between adjacent levels. Higher levels of an abstraction hierarchy model contain functional information, whereas lower levels contain physical information. Physical information describes the state of objects in a work domain (e.g., pumps, heaters, valves in a process plant). Functional information describes the state of the functions or purposes that those objects are intended to satisfy.

The SRK taxonomy describes three qualitatively different ways in which people can interact with their environment (Rasmussen, 1983). Skill-based behaviour involves parallel, automated, direct behavioural interaction with the world. Rule-based behaviour involves associating a familiar perceptual cue in the world with an action or intent, without any intervening cognitive processing. Knowledge-based behaviour involves serial, analytical problem solving based on a symbolic mental model.

The twofold objective of EID is to encourage the use of skill- and rule-based behaviour (to save on scarce cognitive resources), while providing support for otherwise more effortful and error-prone knowledge-based behaviour (to cope with unfamiliar and unanticipated situations requiring adaptive problem solving). To achieve this aim, the framework comprises three design principles, each directed at supporting a level of the SRK taxonomy.

1. *Skill-based behaviour*—workers should be able to act directly on the interface.
2. *Rule-based behaviour*—there should be a consistent one-to-one mapping between the work domain constraints and the perceptual information in the interface.
3. *Knowledge-based behaviour*—the interface should represent the work domain in the form of an abstraction hierarchy to serve as an externalized mental model for problem solving.

1.2. Computer networks and EID

Burns et al. (2003) pioneered the application of EID to computer network management. They selected a university residence network consisting of about 65 switches as a testbed, and conducted a work domain analysis of that network using the abstraction hierarchy. Using the information requirements identified by that analysis, they then built a prototype EID interface with 3-D graphics, and compared that interface to a commercially available network management tool in a static usability test. Representative network data were displayed on each interface as static screen shots, and participants were required to perform two fault detection tasks and three diagnosis tasks. The latter required participants to assess the state of a single device that was pre-identified by the experimenters. The results showed that the EID interface was slower on the fault detection times than the commercially available interface, but faster and more accurate on the fault diagnosis tasks.

The research presented here extends this important work in several ways. First, we applied EID to a different and larger testbed (our university campus network), thereby testing the scalability of the framework within this sector. Second, we investigated the impact of fault frequency, thereby testing the robustness of the framework to increasingly dynamic events. Third, we also investigated the impact of network size by varying the number of nodes. Because the number of work domain components changes across trials (e.g., 57 nodes vs. 220 nodes), the structure of the work domain model representing the components and their functions will necessarily expand or contract as the network size is increased or decreased, respectively. Therefore, this manipulation allows us to test the robustness of the EID framework to changes in work domain structure. As far as we know, this manipulation has never been conducted in previous work on EID because the number of work domain components and the interconnections between them has been held constant within experiments (e.g., Pawlak and Vicente, 1996; Xu et al., 1999). Changing the number of components within an experiment adds new challenges to the EID framework, specifically a need for expandable or reconfigurable interfaces and a stronger need for operator adaptation to novelty and change to accommodate the changes in work domain structure. Fourth, each of our participants performed 30 trials, making it more likely that the results would generalize to more experienced network managers. Finally, our experiment was conducted with a real-time, dynamic simulation using realistic data and required participants to perform a more open-ended task that had them monitor the entire network, increasing the representativeness of the evaluation, and thus, the potential generalizability of the results to operational settings.

1.3. Experimental testbed

The University of Toronto (U of T) computer network was chosen as the representative experimental testbed on which the analysis was performed. The NOC provided much useful information, including several maps of the University's computer network at different times (see Fig. 1 for an example). This enabled us to get an idea of how the network grew over time, which in turn allowed us to generate realistic models of different-sized networks. Although the largest network used in the experiments (comprising 220 nodes) was taken directly from network maps (i.e., the names of the nodes, as well as the hierarchy from the gateway to the departmental nodes, are the same), the smaller networks—subsets of this larger one—are not entirely historically accurate. Owing to load balancing, the hierarchy of the network was shuffled several times. To simplify the design of the simulations, simple subsets of the largest (most recent) network were used for the smaller ones.

The U of T Network includes a 50 Mb/s (Mbit/s) gateway to ONet (the University's connection to the Internet), 10 key routers, and several departmental routers. The network as it stood in September of 2001 (the last map that was incorporated into the experimental work domain) had 220 supervised switches or routers. A 2-day representative sample of traffic on all nodes on which the NOC monitors traffic was used as the basis for all faults used in the simulations; although not all faults come directly from this data, they are based upon trends seen in the sample (peak levels, duration of faults, etc.).

1.4. Information requirements

A work domain analysis was conducted for the U of T Network using the abstraction hierarchy framework to identify the information requirements that should be built into an ecological interface. A brief summary of the analysis follows (see Duez, 2003 for details).

Physical form: This level includes the model of the device, as well as information (provided the device has been properly configured) about the physical location of the device, and contact information for the person or office responsible for maintenance of the device.

Physical function: This level includes information about the configuration (e.g., IP address) and capacity (maximum speed) of the ports on a device.

Generalized function: This level includes summary functional statistics—the load on each port, as well as the ratio of good to bad (error) packets—in addition to more detailed information available for a selected port, including a detailed breakdown of packet sizes, destinations and errors from the last time slice.

Abstract function: This level represents the network in terms of first principles (i.e., conservation of information—the ability of a device on the network to transmit

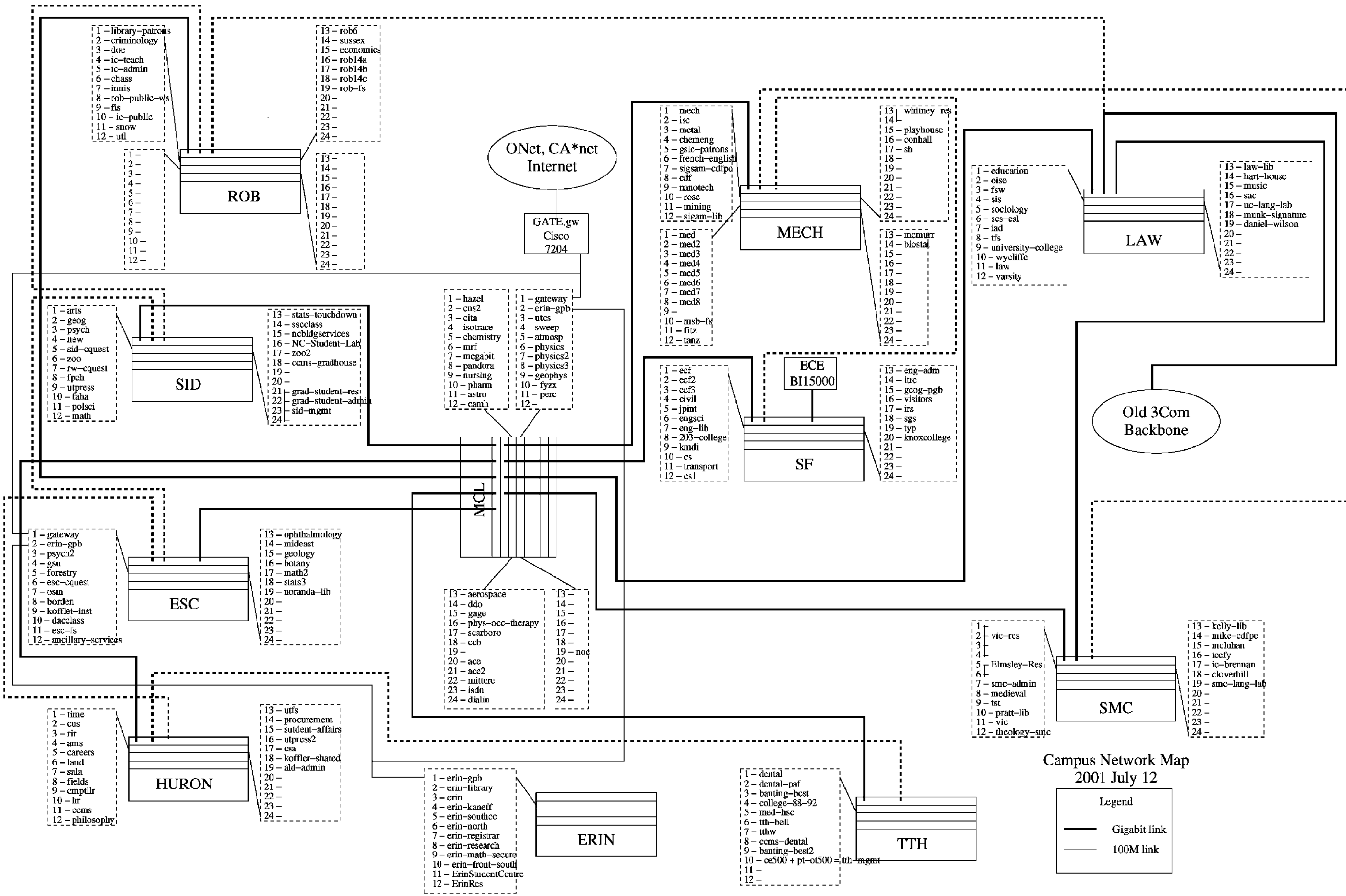

Fig. 1. The University of Toronto Campus Network Map (as of July 12, 2001) used as the experimental testbed for this research. Individual node labels are not intended to be legible.

information from its origin to its intended recipient, or recipients). As load (utilization) on a device plays a key role in determining this ability, the load through every port is coded qualitatively according to three categories (normal, high and critical), which are defined by thresholds specified by the network designers. This coding is represented by a colour scheme, described below.

Functional purpose: This final level indicates the state of the network in terms of its overall purpose: fast, reliable communication. This includes a summary of the size of the network, as well as a numeric (and percentile) indication of the fraction of nodes within the network that are operating below the first load threshold.

In summary, constraints exist at every level of this work domain, from the location and interconnectedness of the equipment, the relations governing its normal operation, the functions it is intended to achieve, the abstract information flows in the network, and finally, the purposes that the network is designed to achieve.

There is one important difference between this analysis and the approach taken by Burns et al. (2003). Our analysis follows the traditional abstraction hierarchy modeling approach, with high levels of abstraction presenting information about aggregation of nodes and low levels of abstraction present information about individual nodes. Rather than adopting a part-whole relation, Burns et al. chose instead to decompose their network using several layers of the OSI architecture, such as application, network, data link, and physical (see Kuo, 2001 for a thorough treatment of this analysis). The relative advantages and disadvantages of these two modeling approaches, and their differential impacts on human performance are not well understood at this time, and therefore remain to be investigated in future research.

1.5. P+F interface

Based on this analysis, an interface that contained physical and functional (P + F) information at all levels of the abstraction hierarchy was created. In designing this interface, it was necessary to bear in mind that some of the elements would be removed in order to create the corresponding P interface (see below).

The P + F interface, shown in Fig. 2, is divided into five viewports. Because this interface is solely for monitoring rather than modification of network performance, there is no way to change the behaviour of the devices. The controls described below with each viewport change the display, but do not affect the performance statistics for any nodes or ports.

In addition to the viewports designed as a result of the abstraction hierarchy analysis, a sixth viewport consisting of a clock and two buttons exists to facilitate experiments conducted with this interface.

Main viewport (#1): Fig. 3 illustrates the main viewport of the P + F interface. It consists of a hyperbolic tree representation of the network, an idea developed at Xerox PARC (Lamping and Rao, 1995), and used in an EID interface for the first time. An example of this construct can be found in the site map of Inxight [www.inxight.com], a spinoff company from Xerox. Each node in this hyperbolic

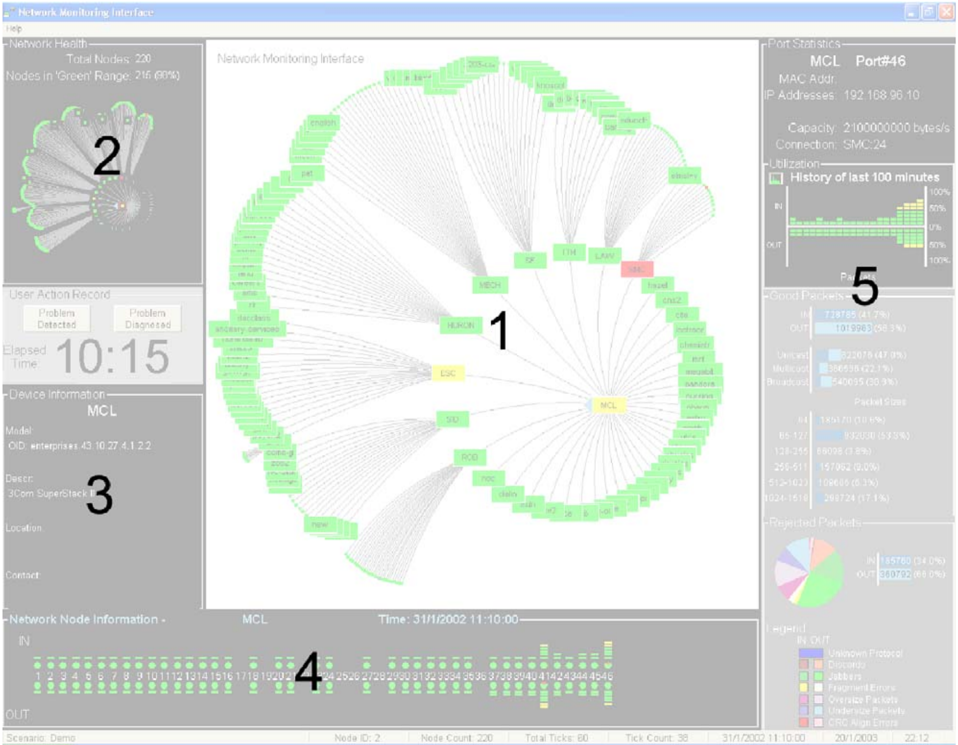

Fig. 2. The five viewports in the P + F interface for the U of T network.

tree represents a network device (e.g., switch or hub). Links represent connections between two nodes.

Each node in this tree contains the name of the device represented, as well as colour-coding for the incoming and outgoing loads: nodes corresponding to normal loads (for these experiments, utilization lower than 40%) are shown in green; those showing a high load (utilization between 40% and 75%) are displayed in yellow; and those showing a critical load (above 75%) are displayed in red. If a node exists, but no data is currently present for that node, it is deemed "unavailable," and is coloured grey.

As the nodes move towards the periphery, they must shrink in size until they become too small to contain the device name with any legibility. As this happens, the node graphic simplifies; it becomes a dot (preserving the same colour convention). The name is presented as a "mouse-over" piece of information. This makes this interface highly scalable, as it can easily scale to over 1000 nodes (Lamping and Rao, 1995), while allowing proper focus on a selected node.

This main viewport is the key source of control for the interface. The map itself can be "dragged" around by clicking on any point with a mouse, and dragging it. As the nodes nearest the centre are provided the greatest amount of focus (compared

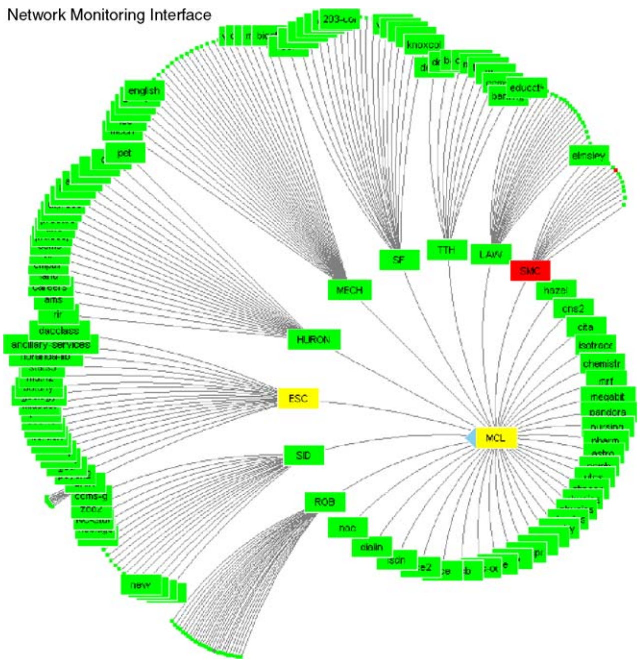

Fig. 3. The main (#1) viewport of the P + F interface.

with those near the periphery), one can change which part of the network is receiving the greatest amount of focus. Additionally, clicking on a node "selects" it. The map moves so that the selected node is at the centre; furthermore, information about that node is displayed in the left-hand "Device Information" viewport, and summary information for each port is displayed along the bottom "Network Node Information" viewport (see below).

Network health viewport (#2): The top-left viewport, shown in Fig. 4, represents the overall network health. Each device is simply represented as a small dot on a static map (a hyperbolic tree, as on the main viewport), the colour of which is a preliminary indication of the device's health. This colouring system is identical to that for node colours described above. This viewport also gives a quick numerical summary of the total number of nodes that make up the network, as well as the overall "health" of the network, measured by the number and percentage of nodes which are working within tolerance ("green" nodes).

Although this map is static, clicking on one of the nodes has the same effect as clicking on that node in the main viewport. The node is selected (with the results in other viewports as described above), and the map in the main viewport is moved so that the selected node is in the centre.

Device information viewport (#3): The third viewport, shown in Fig. 5, gives information for a selected device, and is blank if no device is selected. This includes device type, description and location, as well as contact information for the device

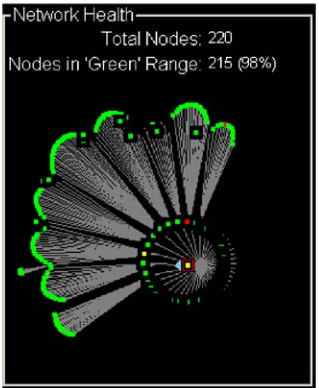

Fig. 4. The network health (#2) viewport of the P + F interface.

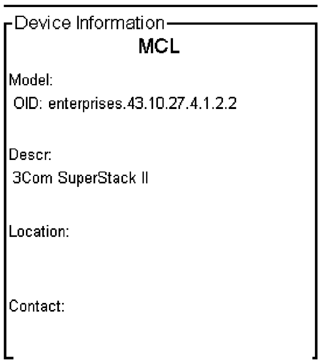

Fig. 5. The device information (#3) viewport of the P + F interface.

Fig. 6. The network node information (#4) viewport of the P + F interface.

manager. This information appears in the viewport when a device is selected, and remains as long as the node is selected. If this information is not known for the selected device, the fields are left blank.

Network node information viewport (#4): The next viewport provides information about network nodes and is shown in Fig. 6. Since there is no constant number of ports for all nodes, a large amount of space (corresponding to a theoretical maximum number of ports) is reserved for this viewport, although it is seldom completely used. This viewport lists summary information for all of the monitored ports of the selected node. (If no node is selected, then this viewport is blank.) The port numbers are listed along the centre of the graphic, with pie charts above and below each number contrasting the number of "good" packets (in green) vs. the number of error packets (in red) being received and transmitted on that port (as the labels indicate, information corresponding to incoming packets and load is displayed above the port number; outgoing information is shown below). Error packets are those that are not received or transmitted (depending on the direction) for various reasons, including fragmented packets, packets which fail a CRC check, packets which are discarded due to full buffers, etc.

The load graphics for each port represent the load as a percentage (utilization), from 0% to 100%. The colour convention for load described above applies here too: bars corresponding to normal loads are shown in green; those showing a high load are displayed in yellow; and those showing a critical load are displayed in red.

To obtain more detailed information for any port, including a detailed summary of packets in the last time slice, as well as a load history, the user can click on any of the graphics corresponding to the desired node (i.e., either load bar, either pie chart, or the number label). This will cause information for that specific port to be displayed in the "Port Information" viewport, described next.

Port information viewport (#5): When a port is selected, detailed information is shown in the right-hand "Port Information" viewport, illustrated in Fig. 7. This includes port-specific configuration information—neighbour, speed, MAC and IP addresses, and the name of neighbouring nodes—as well as performance information.

Performance information includes a utilization history (either split or combined, wherein total load is the average of incoming and outgoing load), and detailed breakdowns of both good and bad packets (number in vs. out, histogram of sizes, numbers of unicast, multicast, etc.).

1.6. P Interface

Fig. 8 illustrates the P interface, which contains physical (P) information identified by the lower levels of the abstraction hierarchy analysis. This design was chosen as a baseline because it is representative of the types of information contained in existing network management interfaces (Kuo, 2001). The P interface was created by starting with the P + F interface, and removing elements that are not typically found in network management interfaces. The red-yellow-green colour-coding (corresponding to the Abstract Function level) was removed entirely; nodes in the P interface are green regardless of the load on any of their ports. Inactive nodes, though, were still represented in grey. (Note that due to this change, the wording in the Network Health view had to be changed; the data and criteria for Network Health, however, are the same as those used in the P + F interface.) Not all generalized function information was removed, however. Although detailed information regarding both good and bad packet distributions was eliminated, the interface retains basic

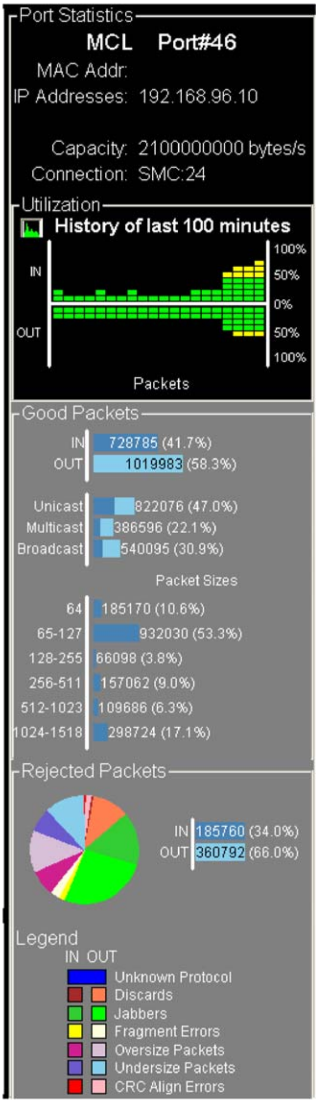

Fig. 7. The port information (#5) viewport of the P+F interface.

information about load and the ratio of good packets to bad, as this information is often included in existing network management interfaces.

1.7. Hypotheses

Because EID is intended to help operators adopt to novelty and change, we expected that fault detection and fault diagnosis performance with the P+F

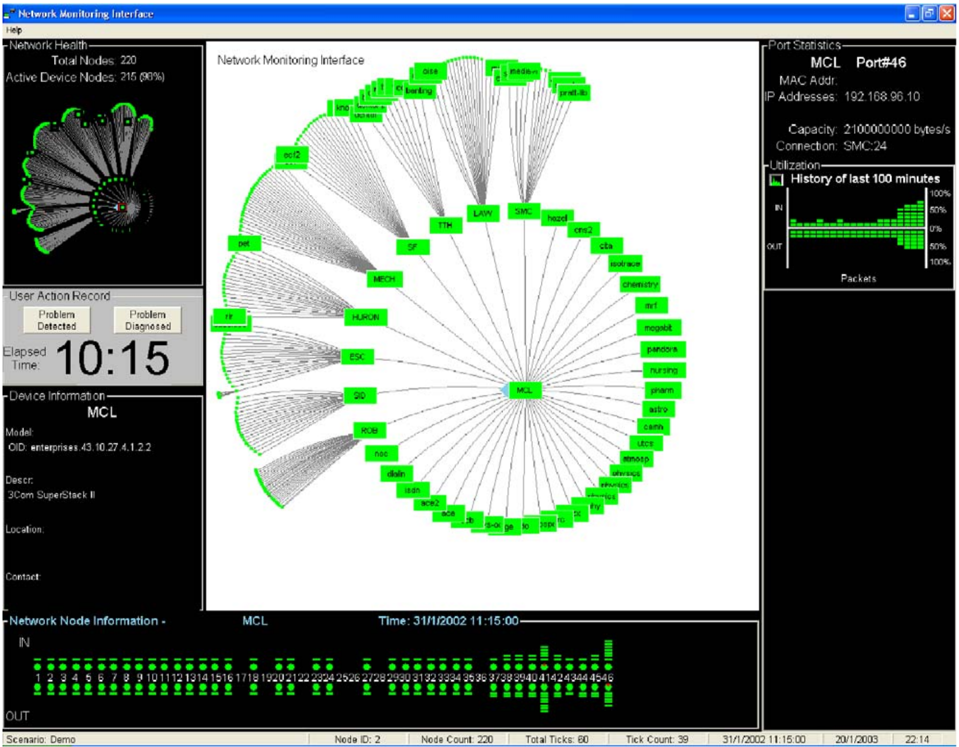

Fig. 8. The P interface for the U of T network.

interface would be less disrupted by increases in fault frequency and network size than the P. Thus, with frequent faults and large networks, performance with the P + F interface should be superior to that with the P.

2. Method

2.1. Experimental design

A mixed experimental design with two within-participants factors (network size and failure frequency) and one between-participants factor (interface) was adopted. Network size refers to the number of nodes in the work domain for a particular trial. Four levels were tested: 220, 154, 101 and 57 nodes. Each of these networks was taken as a subset of the next largest; that is, all nodes in a given network were present (with the same connections) in all of the larger networks.

Frequency of failure corresponded to the rate at which devices failed (as an average time-to-failure), and was also varied between trials. Three mean rates of failure were used: 1 fault-per-minute (fpm), 0.6 and 0.25 fpm. (The precise timing of the faults was random.) Each participant completed six practice trials, as well as two trials for each combination of network size and fault rate in a randomized order, for a total of 30 trials $[6+(2\times4\times3)=30]$.

Interface refers to the use of either the P or the P+F interface (as described above), and thus took on two conditions.

2.2. Experimental domain

The domain simulated was that of the backbone of the U of T campus network (Fig. 1). Whereas the entire network consists of several thousand PCs, hubs, switches and routers, the simulation only covered the portion administered directly by the University NOC. This network is comprised of the gateway to ONET (the Internet connection), approximately 20 key network switches and routers, and close to two hundred secondary switches served by these key devices.

2.3. Experimental task

Each 20-min trial required participants to monitor a real-time, dynamic simulation of the microworld. During every trial, few or many faults occurred, depending on the average time-to-failure for that trial. Faults varied randomly by severity, location and type. Participants were required to correctly diagnose the root cause of each fault. Because the task was one of monitoring and diagnosis, equipment was not repaired as soon as a fault was diagnosed, but stayed inactive (or defective) for a set period of time, which varied from fault to fault. (A detailed listing of the timing and duration of faults for each trial can be found in Duez, 2003.) Examples of faults include sudden loss of contact with a node (or group of nodes), abnormally high traffic due to a large amount of corrupted traffic (bad packets), or the unavailability of a portion of the network due to prolonged high traffic between two nodes.

Participants were instructed to click on the "Problem Detected" button on the screen as soon as they detected a fault, and to speak the location of the perceived fault. They were then to diagnose it to the best of their abilities, and—having reached a diagnosis—click on the "Problem Diagnosed" button and speak their diagnosis. This diagnosis was recorded by a video recorder, and later scored by the experimenter.

2.3. Apparatus

The Network Monitoring Interface software was run on a PC clone with a 1.7 GHz Pentium processor, 256 MB of PC-133 SDRAM, with an ABIT GeForce MX 32MB video card, and viewed on a 19" ViewSonic PF790 "PerfectFlat" monitor; control took place through the use of a wheel mouse and regular (Windows-ready) keyboard. The software was run under the Windows XP Professional Edition operating system. Verbal protocols were collected using a Sony CCD-FX330 8 mm Handycam with a lapel-worn unidirectional microphone.

2.4. Participants

Participants were sampled from the population of undergraduate computer science students at the University of Toronto who responded to a notice posted to several student newsgroups. Participants' ages ranged from 19 to 35 years, with an average of 25.4 years. They had a minimum of 6 courses in computers, with an average of 1.2 relating specifically to computer networks.

Previous studies have shown a correlation between cognitive styles and performance on interfaces developed using EID (Torenvliet et al., 2000). The spy ring test (SRT) (Pask and Scott, 1972) has been used as a tool for scoring potential participants on cognitive style, making a distinction between serialist and holist cognitive styles. To eliminate cognitive style as a confounding factor, all applicants first had an SRT administered. Of a pool of 16 applicants, the five closest pairs (10 participants) were selected, and each member of a pair was randomly assigned to one of the interfaces. Applicants also filled out a demographic questionnaire, indicating their age, level of education and the number of courses they had taken which pertained to computers, as well as to computer networks specifically.

One participant dropped out of the experiments after two practice trials, citing an overloaded class schedule; two more participants (i.e., another participant pair) were contacted to raise the number of active participants; however, of these, one participant declined the offer. Therefore, a total of ten participants took part in the study, with six using the P interface, and four using the P + F interface.

2.5. Procedure

The first 2 h of the experiment, spanning two sessions, were for preliminary briefing; after this, participants completed a total 3 h of practice trials and 12 h of regular trials, for a total of 17 h of participation. Each trial was twenty minutes long and there were two trials per 1-h session, with time given to participants in each session to rest between trials. The briefing consisted of the SRT (first session), and information regarding the work domain and interface (second session). After all participants completed the SRT, they were assigned to an interface. In their second session, participants were given a description of the work domain being used (computer networks), as well as a detailed description of their interface. Participants were required to answer 100% of the questions correctly to demonstrate a thorough understanding of the material before proceeding either to the description of the interface or to the experiment. Participants were also given detailed instructions regarding what was expected of them in the experiment.

The remainder of the experiment consisted of participants performing trials while the experimenter watched (unobtrusively), recording approximate times and scoring diagnoses. Questions for clarification were allowed during the initial six practice simulations, but were not permitted after that point. Participants were asked to schedule their 15 1-h sessions on as close to a daily basis as possible. The participants spent an average of 4 weeks in the experiment, and were paid $10/h for their time.

2.6. Performance measures

Four measures were chosen to evaluate participants' performance: time-to-detection (TDe), time-to-diagnosis (TDi), average diagnosis score and percentage of faults detected. Each of these statistics (with the exception of the last) were calculated for each individual fault and then averaged over all faults in a single trial, and that average was used as the data point for that trial.

TDe represents the time that elapsed between the onset of a fault and the participant's detection of that fault, i.e., the time at which they click on the "Problem Detected" button in connection with that fault. It is important to ensure that each click on the "Problem Detected" be linked correctly with the fault that the participant has detected. Both the verbal protocol and the logs generated by the software assisted in ensuring that the correct faults were associated with each click: every time the "Problem Detected" button was clicked, information regarding the current state of the interface (which node and ports were selected) was logged; in addition, comparing with the verbal protocol to ascertain which problems were detected at which (approximate) times helped confirm which problem was being identified each time the participant clicked on "Problem Detected." The experimenter recorded an approximate time of detection for each detected fault to speed up this process and reduce uncertainty. (These approximate times could then be compared with the logs.)

TDi represents the time that elapsed between the onset of a fault and the participant's successful diagnosis of the root cause of that particular fault as determined from their verbal protocol (see below). If a participant's diagnosis was incorrect or incomplete, no TDi statistic was recorded for that fault. The experimenter rated diagnoses as the participants performed the experiment whenever possible to minimize the need to review and scrutinize the protocols.

Participants' diagnosis of each fault was scored from 0 to 3 (Pawlak and Vicente, 1996). If a fault was not diagnosed—either because the participant did not notice it, or because they were distracted by another fault and never returned—then participants received a score of 0 for that fault. If a fault was detected and a diagnosis attempted—but incorrect (wrong node, complete misinterpretation of the symptoms, etc.)—then the participant received a score of 1 (e.g., "there seems to be a problem with node MECH," when indeed the fault actually lay with the node "biostat"). If a diagnosis was almost correct (either a slight factual error—mistaking a symptom for a cause, for instance—or small error in localization), then the participant received a 2 (e.g., "node 'cdfpc' is experiencing higher than normal loads" when the load was not the cause of a problem, but were rather the consequence of a large number of CRC/Align errors). A correct root cause diagnosis received the maximum score of 3 (e.g., "the node stats2 has become unavailable due to excessive load on that node," when such was indeed the case).

Percentage of faults detected was calculated by taking the number of faults correctly detected (regardless of the quality of the diagnosis), and dividing by the total number of faults in a trial.

2.7. Data analysis

Statistical significance was assessed using 95% confidence intervals because they are more informative than traditional null hypothesis tests, such as ANOVA (see Appendix A). If the confidence interval bars for two means do not overlap, then the difference between means is statistically significant at the $p<0.05$ level. Only statistically significant results are presented. Data were analysed using statistical software packages (Matlab, Microsoft Excel and SPSS).

3. Results

3.1. Time-to-detection (TDe)

The P+F participants detected faults significantly more quickly than the P (mean of 36.8 vs. 44.6 s, respectively). The largest network studied—one with 220 nodes—produced an average TDe across all participants of 51.6 s, which is significantly slower than that observed with a network of 101 nodes (40.4 s). There were no significant differences in detection time between the three smaller network sizes.

No significant main effect was found for the average fault rate in the networks, and no significant interactions were found between any combination of interface, network size and fault rate.

3.2. Time-to-diagnosis (TDi)

The only significant main effect observed with diagnosis time was for fault frequency. The lowest fault rate (0.25 faults/min) was found to lower the TDi, with an average of 36.4 s for 0.25 fpm, compared to 47.7 s for the next-lowest fault rate, 0.6 fpm. This could be due to the extra time participants had to diagnose a fault, without fear of another fault occurring elsewhere within the network. (Although participants were not told with what average frequency faults would occur, they might have been able to sense, partway through a simulation, how "busy" it was.)

The only significant interaction was between network size and fault frequency, where—once more—there was a significant difference between TDi for 0.25 fpm and the other fault rates, but only for a network with 157 nodes.

3.3. Diagnosis score

The P+F participants scored significantly higher (average = 2.7) than P (average = 2.3). Neither fault rate nor network size produced significant main effects.

As shown in Fig. 9, the interaction between interface and fault frequency was also significant. The P+F participants scored significantly higher than the P on all fault rates, and that advantage increased at higher fault rates.

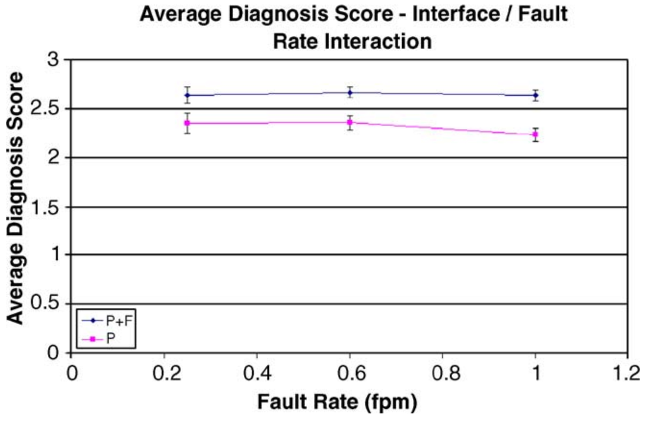

Fig. 9. Interaction between interface and fault rate for diagnosis score.

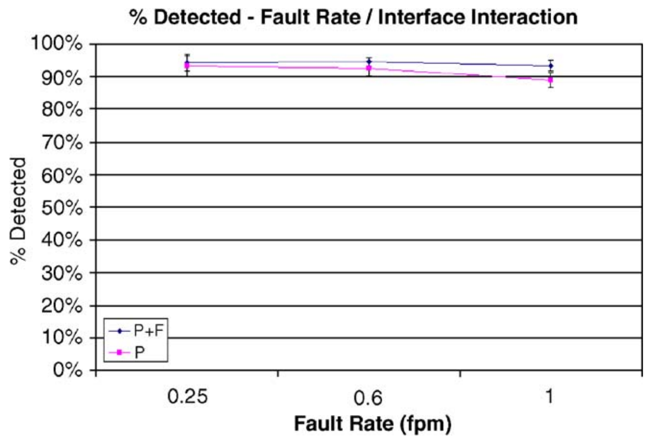

Fig. 10. Interaction between interface and fault rate for % faults detected.

3.4. Percentage of faults detected

There was no significant main effect from any of the three factors for the percentage of faults detected, but as shown in Fig. 10, there was a significant interaction between interface and fault frequency. Under lower demand loads (i.e., longer mean times-to-failure), the interfaces did not differ significantly, but as the loading increased, the average percentage of faults detected by the P participants dropped (to 89%), whereas the percentage of faults detected by those using the P + F interface remained relatively constant (93%). This difference was significant at the highest fault load (1 fpm).

Table 1
Results of power and effect size analyses

	TDe	TDi	Score	% Det
Power				
Interface	0.188	0.093	0.872	0.190
Size	0.999	0.999	0.962	0.673
Frequency	1.000	1.000	0.306	0.335
Frequency * interface	0.169	0.102	0.191	0.155
Size * interface	0.284	0.598	0.936	0.282
Frequency * size	0.228	0.579	0.780	0.931
Frequency * size * interface	0.149	0.278	0.633	0.669
Effect size (eta-squared)				
Interface	0.155	0.055	0.611	0.157
Size	0.739	0.647	0.533	0.331
Frequency	0.785	0.826	0.230	0.262
Frequency * interface	0.111	0.056	0.143	0.118
Size * interface	0.181	0.289	0.501	0.151
Frequency * size	0.134	0.220	0.343	0.458
Frequency * size * interface	0.083	0.113	0.276	0.304

3.5. Power and effect size analyses

As shown in Table 1, the power associated with the various effects varied considerably, from 0.093 and 1.0. Effects with low power may be statistically significant with a larger sample size. The effect sizes also vary widely, from 0.056 to 0.826. Values around 0.5 and 0.8 are considered medium and large size effects, respectively. The main effects tend to be larger than the 2- and 3-way interactions.

4. Discussion

This research tested the applicability of EID to a new domain, one with a different property than those, which have been primarily examined to-date: frequently changing work domain structure (as well as state). The P + F interface resulted in comparable or better performance than the P interface under all conditions tested. The advantage of the P + F interface was greatest under a heavy fault load; whereas performance with the P + F interface was relatively robust to variations in fault load, performance with the P interface declined as load increased, suggesting that the benefits of EID increase as task demands increase.

Several of these results are consistent with the theoretical motivation behind EID, which focuses (although not exclusively) on supporting discretionary decision-making tasks requiring problem solving (Vicente and Rasmussen, 1992). The significant interaction between interface and tasks demands observed here is also similar to those reported in evaluations of EID in other domains, where the presentation of higher-order functional information had only a marginal or non-significant impact on routine, procedural tasks, but led to performance improvements on more open-ended tasks requiring problem-solving behaviours (e.g., Xu et al., 1999). The advantage of the P+F interface for diagnosis tasks also replicates the findings of Burns et al. (2003) in network management, generalizing their results to larger networks, more experienced participants, and more representative tasks. But whereas we also found an advantage of the P+F interface for fault detection, Burns et al. found that a commercially available interface led to significantly faster detection times than their EID interface. The reasons for this difference are unknown. One possibility is that the 3-D forms adopted by Burns et al. are visually more complex than the hyperbolic tree used here, slowing down the initial detection of problems. Another possibility is that the commercially available interface used as a control condition by Burns et al. may have more sophisticated alarming capabilities than the P interface used here, which could quickly draw participants' attention to a problem.

We were surprised that the increase in network size did not have more of an effect on performance. The power associated with this effect ranged from 0.673 to 0.999 (see Table 1), which rules out low power as an explanation. Perhaps the innovative hyperbolic tree navigation mechanism—common to both interfaces—allowed participants to deal with large scales equally well.

5. Conclusions

EID appears to provide a promising basis for designing improved interfaces for computer network management. By identifying functional information and presenting that information using graphical integration techniques, more robust performance may be achieved compared to existing approaches to network management interface design.

5.1. Limitations and future work

The participants were not professionals in the field of network management or monitoring, but rather computer science students who had an interest in this field. Also, alerts and alarms that are normally found in network management interfaces were omitted to isolate the effects of the EID framework. Therefore, the generalizability of these results to more representative users and conditions remains to be explored.

Other issues for future research include: evaluating the impact of EID on larger networks consisting of thousands of nodes, examining the verbal protocol data in more detail to identify the participants' mental strategies, investigating the effect of the interconnectedness of nodes and isolating the performance impact of different visual coding techniques, such as the hyperbolic tree mechanism.

Acknowledgements

This research was sponsored by research grants from Microsoft Corporation (Dr. Marshall McClintock, grant monitor) and the Natural Sciences and Engineering Research Council of Canada, as well as the E. W. R. Steacie Memorial Fellowship. We would like to thank Doug Carson and colleagues at the University of Toronto NOC for their help and patience, Catherine Burns and Johnson Kuo of the AIDL at the University of Waterloo for sharing their network management research results and expertise, Ron Blair of Blair Consulting for implementing the interfaces and simulation, Ramana Rao and everyone at Inxight Software, Inc., for kindly allowing and facilitating the use of their Hyperbolic Tree technology, and the reviewers for their thorough and constructive comments.

Appendix A. (Adapted from Vicente and Torenvliet, 2000)

Because it is customary to analyse data from human–computer interaction experiments using only analysis of variance (ANOVA), it is worthwhile explaining why we chose to use confidence intervals instead (see also Loftus, 1993 for more details). Whereas the results from an *F*-test in an ANOVA merely show the probability that the data could have arisen given that the null hypothesis were true, confidence intervals directly provide information about the range of values within which population parameters are likely to be found. As such, they have several advantages over ANOVA. First, confidence intervals provide a graphical rather than an alphanumeric representation of results (see Figs. 9 and 10, for example). By applying human factors knowledge to the presentation of scientific data, this format makes it easier for researchers to extract information (rather than data) from their statistical analyses. Second, the width of a confidence interval provides an indication of the precision of measurement. Wide confidence intervals indicate imprecise knowledge, whereas narrow confidence intervals indicate precise knowledge. This information is not provided by the *p*-value given by an F test in an ANOVA. Third, the relative position of two or more confidence intervals can provide qualitative information about the relationships across a set of group means. If two confidence intervals do not overlap, then the means are significantly different from each other statistically, otherwise they are not. While this information can be gained from an ANOVA, confidence intervals add information about the order of means across groups, information that cannot be found in, for instance, an ANOVA table. Fourth, whereas an ANOVA requires researchers to perform significance tests for all permutations of effects (including many that may have no scientific interest at all), confidence intervals can be calculated selectively to answer only those questions that are of interest to the researcher. The relative simplicity of presentation that results from this more focused approach can make it much easier for readers to interpret the results of an experiment and to extract the meaning of those results. Finally, confidence intervals also allow us to assess the statistical significance of individual effects. If a confidence interval on a group mean includes zero, then the treatment does not have a significant effect.

The important point to take away is that confidence intervals provide much more information than do ANOVAs alone. Furthermore, that information is provided in a graphical format, thereby making it easier for researchers to pick up meaningful patterns perceptually (e.g., width of bands, overlap across bands, inclusion of the zero point). This added information can lead to a very different interpretation than may be obtained by reliance on ANOVA alone, which is why physical scientists and engineers virtually never use ANOVA and frequently use confidence intervals and why a growing number of psychologists and statisticians now advocate the use of confidence intervals in the statistical analysis of data from experiments with human participants (Vicente and Torenvliet, 2000).

References

Blum, R., Kaplan, J.M., 2000. Network Operations Centers. Lucent Technologies, Sunnyvale, CA.

Burns, C., Kuo, J., Ng, S., 2003. Ecological interface design: a new approach for visualizing network management. Computer Networks 43, 369–388.

Duez, P., 2003. Testing the generalizability of ecological interface design to computer network monitoring (Unpublished MASc thesis). Department of Mechanical & Industrial Engineering, University of Toronto, Toronto.

Kuo, J., 2001. Applying ecological interface design to network performance management (HSSD-01-01). Human Systems in Systems Design Engineering Technical Report Series, University of Waterloo, Waterloo.

Lamping, J., Rao, R., 1995. The hyperbolic browser: a focus+context technique for visualizing large hierarchies. In: Card, S.K., Mackinlay, J.D., Shneiderman, B. (Eds.), Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann Publishers, Inc., San Fransisco, CA, pp. 382–408.

Leon-Garcia, A., Widjaja, I., 2000. Communication Networks. McGraw-Hill, Toronto.

Loftus, G.R., 1993. A picture is worth a thousand p values: on the irrelevance of hypothesis testing in the microcomputer age. Behavior Research Methods, Instruments, & Computers 25, 250–256.

Pask, G., Scott, B.C., 1972. Learning styles and individual competence. International Journal of Man–Machine Studies 4, 217–253.

Pawlak, W.S., Vicente, K.J., 1996. Inducing effective operator control through ecological interface design. International Journal of Human–Computer Studies 44, 653–688.

Rasmussen, J., 1983. Skills, rules, knowledge: signals, signs and symbols and other distinctions in human performance models. IEEE Transactions on Systems, Man, and Cybernetics SMC-13, 257–267.

Rasmussen, J., 1985. The role of hierarchical knowledge representation in decision making and system management. IEEE Transactions on Systems, Man, and Cybernetics SMC-15, 234–243.

Torenvliet, G.L., Jamieson, G.A., Vicente, K.J., 2000. Making the most of ecological interface design: is operator selection necessary? Applied Ergonomics 31, 395–408.

Vicente, K.J., 2001. Cognitive engineering research at Risø from 1962–1979. In: Salas, E. (Ed.), Advances in Human Performance and Cognitive Engineering Research, vol. 1. Elsevier, New York, pp. 1–57.

Vicente, K.J., 2002. Ecological interface design: progress and challenges. Human Factors 44, 62–78.

Vicente, K.J., Rasmussen, J., 1992. Ecological interface design: theoretical foundations. IEEE Transactions on Systems, Man, and Cybernetics SMC-22, 589–606.

Vicente, K.J., Torenvliet, G.L., 2000. The Earth is spherical ($p<0.05$): alternative methods of statistical inference. Theoretical Issues in Ergonomics Science 1 (3), 248–271.

Xu, W., Dainoff, M.J., Mark, L.S., 1999. Facilitate complex search tasks in hypertext by externalizing functional properties of a work domain. International Journal of Human–Computer Interaction 11, 201–229.