

The Cognitive Ergonomics of Knowledge-Based Design Support Systems

Tamara Sumner

Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
<http://kmi.open.ac.uk/~tamara>
T.Sumner@open.ac.uk

Nathalie Bonnardel

Centre de Recherche en
Psychologie Cognitive
Université de Provence,
13621 Aix en Provence, France
nathb@aixup.univ-aix.fr

Benedikte Harstad Kallak

Computas Expert Systems A.S.
Leif Transtads Plass 6
Postboks 430
1301 Sandvika, Norway
bha@computas.no

ABSTRACT

Critiquing systems are a type of active, knowledge-based design support system. They propose to positively influence designers' cognitive processes by pointing out potential problems and contentious issues while designers work. To investigate the effects such systems have on the activities of professional designers, a design environment containing a critiquing system was designed, built, and evaluated for a specific area: phone-based interface design. Four professional designers were observed using the environment to solve realistic design tasks. Our protocol analyses indicate that such systems do influence the behaviour of designers, but often indirectly. Designers were observed anticipating the activity of the system and taking preventative steps to avoid it. Differential effects depending on the designers' level of domain experience were also observed. Overall, the system was better suited to the needs of highly experienced designers.

Keywords: Critiquing Systems, Cognitive Ergonomics, Design, Design Environments, Knowledge-based Systems, Phone-based Interfaces

INTRODUCTION

Design problem-solving requires designers to be creative and to express evaluative judgements. During iterative problem-solving designers: (1) construct partial solutions based on their current understanding of the design goals and specifications and (2) evaluate these solutions with respect to various criteria and constraints. The results of these evaluation steps feed back into the designers understanding of the problem and lead designers to focus their attention on specific features of the current solution, which guides further development or modification of these features [2]. Thus, evaluation plays a major role in design because each successive evaluation step guides the course of design activity. The final solution to a design problem is obtained progressively, through iterative cycles of solution generation and solution evaluation [2], as designers continually refine their initial, abstract specification in order to reach a concrete solution.

However, evaluation of design solutions is difficult for both experienced and inexperienced designers because:

- in complex domains, no single person can know all the relevant criteria and constraints [2],
- design solutions must be evaluated from multiple, and sometimes conflicting, perspectives [12], and
- designers do not always recognise problematic solutions.

Domain-oriented design environments [5] have been proposed as computational tools supporting designers to construct and evaluate design solutions. Critiquing systems embedded in these environments augment designers' cognitive processes by analysing design solutions for compliance with criteria and constraints encoded in the system's knowledge-base. These systems support design by allowing designers to creatively propose solutions, while the system actively helps designers to reflect on and assess various features of these solutions. *However, a major challenge is to create active knowledge-based design support systems that are cognitively ergonomic; i.e., systems that effectively support designers' cognitive work without hindering their creative processes.*

While numerous research prototypes of critiquing systems have been built [1, 7, 8], little is known about their cognitive effects on the activities of professional designers. To investigate the cognitive ergonomics of such systems, we have designed, built, and evaluated a domain-oriented design environment supporting phone-based interface design: the Voice Dialog Design Environment. Professional designers were observed using this environment to solve realistic design tasks. Analyses of these design sessions enabled us to identify reactions common to all designers and reactions depending on the designers' level of domain experience. We believe such studies are of interest both from a practical point of view, by suggesting how to build better design support systems, and from a theoretical point of view, by contributing to our understanding of designing cognitive tools.

We begin by briefly reviewing the theories and motivations underlying critiquing systems. Next, we describe the Voice Dialog Design Environment and the specific requirements guiding the development of its critiquing system. We then present the study, including the experimental situation and the results obtained. We finally discuss the implication of these results for designing cognitively ergonomic knowledge-based design support systems.

CRITIQUING SYSTEMS: THE ERGONOMIC CHALLENGE

Critiquing systems contain sets of rules for evaluating different aspects of a design solution. These rules are sometimes referred to as *critics*. Each critic is linked to associated background information and design rationale contained in a hypermedia knowledge-base. Here, we briefly review the literature on critiquing systems and discuss what it means for such a system to be cognitively ergonomic. More detailed information on critiquing systems can be found in [7-9, 11].

At a theoretical level, critiquing systems operationalise Schön's action-breakdown-reflection cycle [13]. According to Schön, this cycle underpins the activities of professional designers. In this cycle, designers engage in situated work until their expectations are not met and they experience a breakdown in the current work situation. At that moment, designers stop and reflect on how to overcome the breakdown before proceeding. These breakdowns in situated action present opportunities for learning and for the construction of new knowledge [6]. The results of the breakdown feed back into the next iteration of the design solution. Critiquing systems strive to promote this cycle and help designers to detect and overcome breakdowns by:

- *analysing* representations of the design to detect potential problems or opportunities,
- *signalling* the designer who then experiences a breakdown and is ready to reflect on its cause,
- using this opportunity to *deliver* information or design rationale associated with each critic to support the designer's reflection processes.

The designer's role is to generate and modify solutions. The system's role is to support the evaluation process by: (1) pointing out problematic situations that might remain unnoticed, (2) communicating debatable issues and different perspectives surrounding a design solution, and (3) helping designers learn about relevant criteria and issues. Figure 1 illustrates these different roles in an "idealised" model of designer-system cooperation [7].

A critiquing system could influence designer behaviour in both positive and negative ways. A cognitively ergonomic system should appropriately guide designers' problem-solving without hindering their creative processes, and thus positively influence both the design process and the design product.

The idealised model (Figure 1) indicates that positive influences on designer behaviour could include: modifying design solutions for the better in response to critics, reflecting on design issues raised by critics, recording design rationale in response to issues raised by critics, taking into account criteria that were previously unknown, and possibly learning the information presented by the critiquing system.

Conversely, negative influences on designer behaviour could include: modifying design solutions for the worse in response to debatable issues raised by critics, and being annoyed or disrupted by critiquing signals (that may be irrelevant, misunderstood or too frequent). Additionally, the system could have no influence at all if designers ignore its signals.

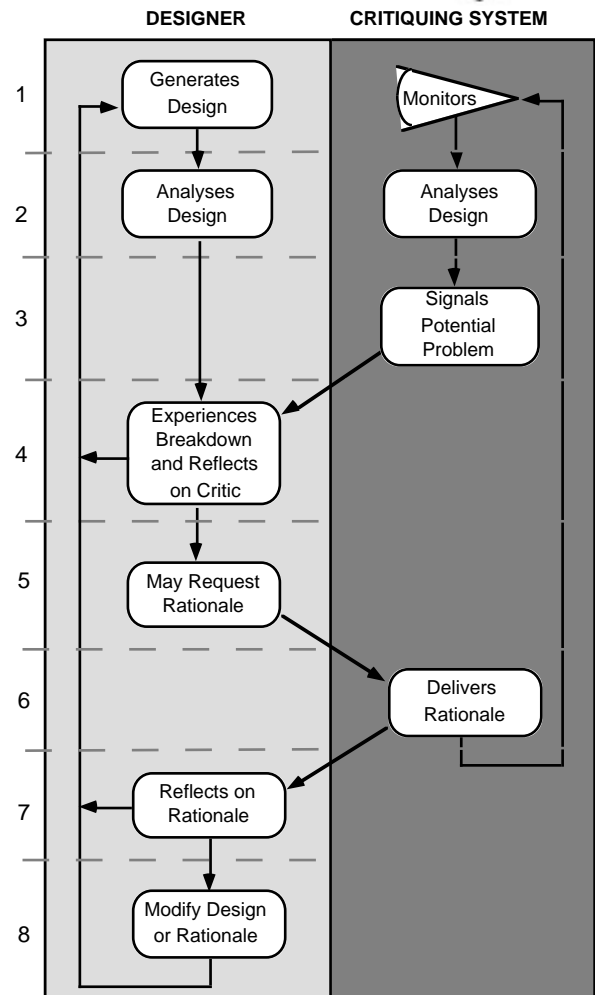


Figure 1: "Idealised" model of designer-critiquing system cooperation during design.

Clearly, the two system goals of noticing potential problems and not being disruptive are diametrically opposed. These competing goals create a very difficult design situation which has been previously articulated as the challenge to "say the right thing at the right time"[8].

To meet this design challenge, we propose an iterative system development cycle where analyses of designers' cognitive processes both precede and follow system building. An understanding of the designers' traditional processes (i.e., without the support system) can be used to guide system design in order to create a system better adapted to the designers' cognitive processes. The understanding of the designers' processes can also then be used to inform system evaluation. This pre- and post-

system approach is useful since the evaluation of complex systems is confounded by the fact that their use substantially changes the traditional task. An analysis of the designers' existing activities can serve as a baseline for assessing how the system has changed the task and whether the changes are for the better [3]. The following section illustrates how this approach was followed in the Voice Dialog Design Environment project.

VOICE DIALOG DESIGN ENVIRONMENT

The Voice Dialog Design Environment (VDDE) [9, 15] supports the design of phone-based user interfaces and was built in collaboration with voice dialog designers at U S WEST (a regional phone company in the US). Voice dialog applications include voice messaging systems for sending and receiving voice mail (e.g. "to listen to your messages, press 1") and voice systems supporting business services (e.g., "for customer service, press 2"). Designing phone-based interfaces involves defining the functionality of an application, specifying the series of audio messages that callers may hear, and creating a series of voice prompted menus requesting callers to press touch-tone buttons to perform certain actions.

VDDE (Figure 2) enables designers to construct, simulate, and evaluate their interface designs. VDDE's construction kit allows designers to sketch out the flow of an interface by arranging domain-oriented building blocks such as voice menus, prompts, and touch-tone buttons into a flow chart-style representation. Designers can hear what their interface sounds like by attaching audio recordings to building blocks in the design and simulating their proposed solution. VDDE's critiquing system monitors designers' actions and comments on potentially problematic or contentious aspects of the design.

Designing VDDE's Critiquing System

Designing a system that is useful and cognitively ergonomic requires paying careful consideration to the design of both the interface and the system's knowledge-base. When evaluating the effects of a knowledge-based system, it is the content and structure of the knowledge-base that is being assessed. Thus, several aspects of design knowledge and practices need to be understood when creating the system:

- What are the evaluative knowledge rules (i.e., criteria and constraints) and evaluative procedures relevant to the specific domain being supported?
- How should this evaluative knowledge be partitioned or organised to reflect different perspectives and design types in the domain?
- When do designers perform different types of evaluations during design? What parts of the design do various types of evaluative knowledge apply to?

To investigate these questions, we analysed the designers' traditional activities at different levels using workplace observations, interviews, analyses of design representations, and protocol analyses of design sessions. On the large-scale level, we looked at the overall design process; i.e., how a design develops over the course of several months as many stakeholders interact and debate the features of a particular design. On the small-scale level, we looked at individual design sessions as expert designers worked on real designs for one or two hours. Using verbal protocols, we analysed in detail the evaluation processes during these sessions.

Determining Evaluative Rules and Procedures

We identified that designers can take into account several types of evaluative knowledge rules: criteria concerning

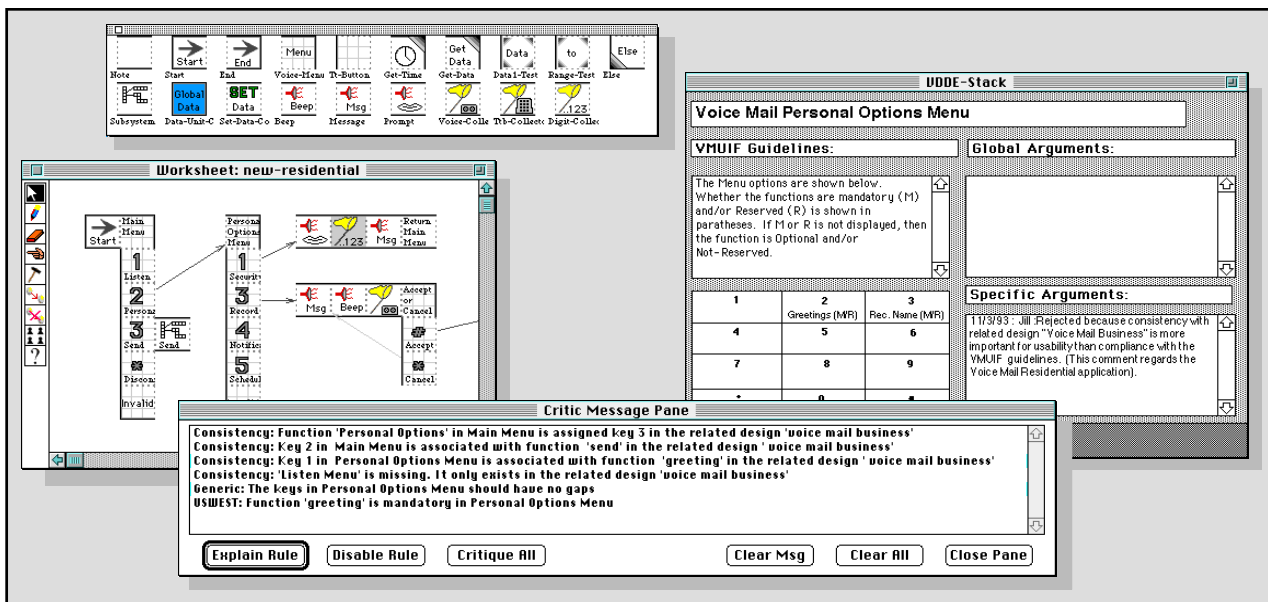


Figure 2: The Voice Dialog Design Environment. Designers select building blocks from the gallery (top window) and arrange them in a worksheet (left window) to create a graphic representation of the audio interface design. A critiquing component analyses the design for compliance with interface guidelines and product consistency. Possible problems are signalled in the critic message pane (lower window). The designer can select a critic message and elect to see the rationale behind the rule and can also add more arguments into the design rationale knowledge-base (right window). The design shown is an excerpt from a hypothetical voice messaging system.

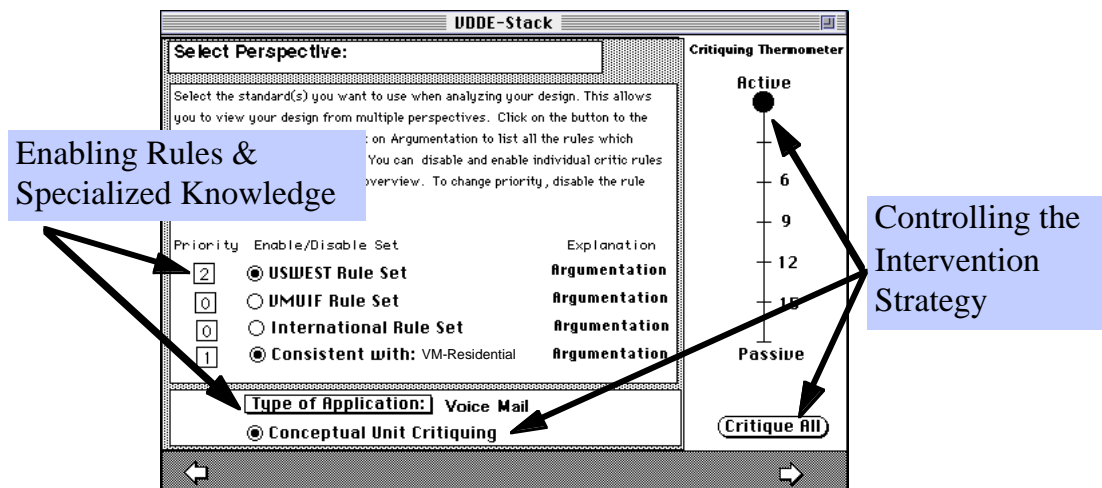


Figure 3: Designers can control the type of design knowledge used to evaluate designs and the intervention strategy used by the critiquing system.

consistency both inside and across design solutions, and criteria and constraints specified in various *user interface guidelines*. Consistency criteria include checking that menus with the same name contain the same functions, and that functions are always assigned to the same touch-tone buttons no matter where they occur (e.g., function *send* is always assigned key *three*). User interface guidelines criteria can include checking that: (i) voice menus do not contain too many prompted functions, (ii) functions in menus are in ascending order, and (iii) there are no numeric gaps between functions in a voice menu.

Related to these two broad categories are criteria concerning *completeness*. Some completeness criteria are relative to the design being compared with; i.e., checking that all the same menus are present. Other completeness criteria are related to the user interface guidelines; i.e., checking that the standard mechanisms for cancelling or backing out of operations are present. Additionally, the relevance of some criteria depends on the *type of application* being developed.

The analysis also enabled us to identify that the designers were using two different types of evaluative procedures: an *analytical* procedure for assessing the solution's features with regard to criteria and constraints, and a *comparative* procedure for assessing the solution's features by comparison with the features of another solution [2]. As a result of these findings, we embedded in VDDE's critiquing system these two evaluative procedures and critic rules reflecting the criteria designers can potentially consider.

Partitioning the Knowledge-base to Reflect Perspectives
Not surprisingly, the two broad classes of evaluative knowledge identified – consistency and user interface guidelines criteria – are related to the designers' overall design objectives. From the perspective of satisfying new or novice end-users of phone-based products, designers want to create designs that comply with different sets of user interface guidelines (regional, national, and international). However, to satisfy marketing groups and existing experienced end-users, designers also try to create designs that are consistent with related products and existing applications. The problem is that many existing

applications predate the interface guidelines and so do not conform to these guidelines. Thus, these two design objectives conflict and it is impossible to satisfy them both. Voice dialog designers must take into account these different perspectives on design and make difficult trade-off decisions between these competing objectives.

The critics in VDDE are structured to reflect these perspectives. The goal of this structuring is to support designers in making difficult trade-off decisions between the guidelines and product consistency. The knowledge-base is partitioned at the top level into four sets of critic rules that designers can enable (Figure 3). Three rule sets correspond to the regional, national, and international phone-based user interface standards. An analytical evaluation procedure is used to compare design solutions against these rule sets. A fourth consistency rule set uses a comparative evaluation procedure to compare two designs for inconsistencies. Each rule set is further partitioned to reflect generic design knowledge (relevant to most phone-based interfaces) and specialised design knowledge (relevant only to certain application types such as voice mail, call answering, and bulletin board applications).

Giving Designers Control Of the System

Designers control the activity of the critiquing system in two ways (Figure 3). First, designers choose what *design knowledge* should be used to evaluate designs by selecting and prioritising rule sets. Priorities only affect the ordering of critic messages in the message pane with higher priority messages being displayed before lower priority ones. Designers can optionally enable specialised knowledge by specifying the type of application being developed.

Second, designers can control the *intervention strategy* of the critiquing system. VDDE supports three different intervention strategies – an active, a passive, and a strategy based on conceptual units. These can be used individually or together in any combination. In VDDE, intervention strategies tie together the *rate* of critic intervention (i.e., how often critic evaluations are performed) with the *scope* of different types of design knowledge (i.e., which parts of the design are analysed with which evaluative criteria). This aspect of VDDE's design was motivated by our

analysis of individual design sessions and performance considerations that we felt would adversely impact the system's cognitive ergonomics.

Specifically, our analysis identified that designers' scope of evaluation falls into three categories: considering the function/key mapping of *individual touch-tone buttons*, examining the contents of *voice menus*, and looking back over the *entire design*. Different types of design knowledge are associated with each of these three different scopes of evaluation. VDDE mimics the observed scoping by selectively using different critic rules from the enabled rule sets depending upon the intervention strategy.

If an active intervention strategy is selected, the system automatically analyses a very localised part of the design whenever designers place or move touch-tone buttons in the worksheet. This strategy analyses the function/key mappings of *individual touch-tone buttons* and the contents of the *voice menu* of the affected touch-tone button in. This localised scope of evaluation reflects observed practices and is very efficient. Clearly it could be very disruptive if there were long system delays if large parts of the design were analysed every each design move.

If a passive strategy is chosen, the system waits for the designer to order an evaluation of the design by pressing the *Critique All* button. This causes all the design's features to be checked, not just the ones resulting from the last action Rules that exhaustively compare the features of two designs (i.e., most consistency criteria) or analyse the *entire design* (i.e., many rules concerning completeness) are only considered when this strategy is chosen.

The conceptual unit strategy analyses the contents of *voice menus*. Whenever the designer moves on to manipulating the contents of another voice menu, the system analyses the menu the designer had been previously working on. This strategy enables the same rules as the active strategy plus additional rules that check for voice menu completeness.

Using VDDE: An Envisioned Design Session

VDDE was designed to support the idealised model of design sessions depicted in Figure 1. At the beginning of a session, the designer can specify the design knowledge and intervention strategies to be used. The default is to use an active strategy and analyse designs for compliance with the U S WEST user interface guidelines.

To construct a design, designers select building blocks from the gallery and arrange them in the worksheet. As the designer works, the critiquing system monitors his or her actions in accordance with the intervention strategy and design knowledge selected. When the critiquing system detects a possible violation, a brief message signalling the violation is immediately presented in a separate critic message window (referred to as "firing"). Instead of providing designers with general admonitions (e.g., "*There should be no gaps in voice menus.*"), the system contextualises the messages to refer to the design currently being constructed (e.g., "*The keys in Personal Options Menu should have no gaps.*") To help designers identify the perspective associated with a detected problem, each message is preceded by the name of the particular rule set the critic is part of. For the design shown in Figure 2,

pressing the *Critique All* button caused critics from both the U S WEST and Consistency rules sets to fire.

Designers can request to see further explanations and design rationale associated with critics by selecting individual messages and pressing the "Explain Rule" button (see Figure 1, lower window). Each critic rule is linked to portions of the user interface guidelines contained in an on-line hypermedia knowledge-base. The "Explain Rule" feature shows designers which building blocks in their particular design caused the critic to fire (the blocks briefly flash) and brings up the relevant portions of the guidelines in the design rationale window. The designer can choose to ignore the critic message or modify the design in response to the critic message. Additionally, the designer can add new information, such as rationale behind difficult trade-off decisions made in response to critic messages, to the hypermedia knowledge-base. Individual rules can be disabled for the remainder of a design session should the designer decide that one is incorrect or simply doesn't apply to the current design.

DID VDDE INFLUENCE DESIGNERS' BEHAVIOUR?

The previous scenario indicates how we expected VDDE to be used in a typical design session. But is this really how these systems get used? Do critiquing systems actually influence designer behaviours in a positive manner? Furthermore, are there differing effects depending upon the designer's skill level? To investigate these questions, four professional designers were asked to think aloud while using VDDE to perform realistic design extensions to an existing voice messaging product.

All four had been professionally employed as phone-based interface designers by the same company. However, their level of experience in general phone-based interface design and the design of voice messaging products was quite different. Two designers were considered highly-experienced, having expertise in both phone-based interface design and voice messaging products. One had detailed knowledge of voice messaging (HIGH1); the other had general familiarity with these products (HIGH2). The medium-experience designer (MED) had been designing phone-based interfaces for three years but had no experience with voice messaging products. The low-experienced designer (LOW) had less than one year's experience in general phone-based interface design.

The critiquing system was configured as shown in Figure 2 with Consistency, U S WEST guidelines and voice messaging rules enabled. The participants were told that the priority settings (Consistency – 1, U S WEST – 2) did not reflect design goals and only determined the ordering of critics in the message pane. Each 90 minute session was videotaped and the designers' verbalisations were transcribed (about 70 pages). When examining the transcripts, we quantitatively and qualitatively analysed the critics considered by the designers, and characterised the critic's consequences on the designers' actions and reasoning. After each session, a 30 minute interview was conducted with each participant to better understand how they perceived the influence and relevance of the critics. In this section, we look at how the critiquing system exerted unexpected, positive, and negative influences on designer behaviour.

Indirect Influences : An Unexpected Model of Use

According to our model, the primary impact of critiquing systems should be on the designer's cognitive processes and actions directly *after* a critic message is presented. However, we observed the system exerting an indirect influence on the designers' reasoning. All designers (even the low-experienced designer) *anticipated critics* prior to their firing. Such anticipations were expressed through comments such as "I wonder if the system will catch me" or "I know I'm going to get dinged here." Some anticipations were confirmed by the later presentation of critics, others were not (this is discussed more fully below). Apparently just being *aware* that critics could fire, influenced the designers' reasoning to look for deficiencies in their solutions that might cause the presentation of critics. As shown in Table 1, for experienced designers such anticipations can represent a significant amount of the system's overall influence. Sometimes designers took preventative steps and modified their design solution to *avoid* having the anticipated critics fire. *Thus, a key finding is that critics exert an indirect influence on designers' cognitive processes that can lead to awareness, anticipation, and avoidance of system activity.*

Table 1. Number of presented and anticipated critics according to the designers' domain-skill level.

Designer	Presented critics	Anticipated critics	Total no. of critics
LOW	15	1	16
MED	6	3	9
HIGH2	6	3	9
HIGH1	3	2	5

Influencing Process: Promoting Reflection

Whether critics were presented or anticipated, we observed all designers *assessing the relevance of critics* before formulating an action to improve the design solution. However, we observed differences in the reflections based upon the designers' domain-skill level.

The HIGH designers appeared to *analyse more deeply* the reasons why they were planning to break rules. In a few cases, this led to modifications of the design requirements (see Table 2 for an example). When HIGH designers

decided not to follow a critic rule, they usually added their rationale for breaking the rule to the hypermedia knowledge-base once the critic fired (3 additions out of 4 broken rules for HIGH2 and 1 out of 1 for HIGH1). Thus, there is some indication that the reflections prompted by these critics primed these designers to record their thinking when the opportunity presented itself shortly thereafter.

While the lesser-experienced designers reflected on the critics, they did not appear to analyse deeply the trade-off decisions involved. They always favoured consistency issues over user interface guidelines after only a brief consideration of the critics. Also, these designers did not record their rationale for breaking critic rules: the LOW designer made no additions to the hypermedia knowledge-base and the MED designer made only one addition concerning a rule he believed was incorrect.

These differential effects were echoed in the post-session interviews. The HIGH designers felt the critics had influenced their mental processes by getting them to think more deeply about their design choices. The other designers felt the critics had little influence on their thinking. *Thus, critics do appear to promote reflection during design, though the quality of reflection, particularly in the area of considering trade-offs, appears to be deeper for more experienced designers.*

Influencing Products: Modifying Design Products

A key function of critiquing systems is detecting and pointing out potential problems which should lead to modifications in the design solution or specification. We observed both presented and anticipated critics influencing designers to modify design products. Not surprisingly, the less-experienced designers tended to be presented with more critics (Table 1) and make more solution modifications in response to critics than the more experienced designers (LOW and MED each made 3 solution modifications). Presented critics seemed to have very little direct influence on the actions of the highly-experienced designers; each of these two designers only made one solution modification in response to a presented critic. For all designers, these modifications were mostly limited to surface features such as renaming a menu option or changing the key assigned to an option.

Table 2. Example excerpt from a highly-experienced designer's verbalisation showing anticipation of a critic

Characterising Designer Activity	Corresponding part of verbalisation
Anticipation of a critic	"I'm struggling with a critic I know is going to come up. I've got four options here... and that's the maximum number I should have on the menu. But I also need to give them [the end-users] a way to listen to, so I'll need a key for that... So, I've got five options, and that's violating the design guideline."
Analysis of the reasons for breaking the critic rule	"I think the reason I backed myself into this corner is I have picked up on the way our current voice messaging vendor has integrated this [service] into their product [...] I knew I was going to get an error because I was violating design guidelines since I have too many options in the menu ..."
Attempt to modify the design... ... which leads to modifying a design requirement	"I don't like the way the menu looks. It's a good guideline. So at this point, [...] I think I would argue for removing the option to print the old documents. [...] I would re negotiate with the market unit about the need to get one of them taken out. And then I'd be able to have everything fit on the menu."

For anticipated critics, all designers modified solution features. Interestingly, the only deep design modifications, such as modifying requirements were made by HIGH designers anticipating critics. Occasionally, all designers took preventative steps and modified design products to avoid having critics fire. *Thus, critics do influence designers to modify design products, though often the influence indirectly arises from critic anticipations.*

Influencing Learning

The designers' ability to anticipate critics may be interpreted differently depending on their skill level. Due to their experience, the HIGH designers probably had previous knowledge concerning what design rules might be represented in the system. For the LOW and MED designers, there are two interpretations, either: (1) they also had previous knowledge or (2) they learned these rules while using the system. To investigate these interpretations, we further examined the anticipated critics. Our analysis showed that only HIGH designers were able to anticipate critics they had not been presented with during the current design session. The LOW and MED designers anticipated only critics that had been previously presented. *Thus, there is some indication these designers learned "in action" critic rules while using VDDE to solve the proposed design problem.*

Disruptive Influences

According to Fischer et. al., the challenge is to keep active systems quiet [8] so that they are helpful but not disruptive. VDDE was fairly quiet; not many critics fired during sessions (Table 1). However, *redundant* critics (i.e., critics that had previously fired) did disrupt the design process. These critics particularly disturbed the highly-experienced designers (each had 2 redundant critics). They felt they had 'already taken care of them' by recording why they were breaking a critic rule in the hypermedia knowledge-base. The designers expected these rules to be locally disabled by the act of recording design rationale. Some redundant critics were the result of overlapping rules being enabled by the active and conceptual unit intervention strategies. Other redundant critics appeared because it is impossible to locally disable rules given the current knowledge-base architecture. Once a rule is disabled, it is globally disabled for the rest of the design session. *Thus, having redundant critics fire is a problem; designers expect to be able to locally disable critic rules.*

Sometimes Not Enough Influence

Another problem was that *sometimes designers did not immediately notice critic activity*. Often times, the critic message window was covered up or even dragged off the monitor to make room for the growing design (the designers dragged off many windows to make room). In our effort to be realistic, we replicated a feature of "real" designs: they are large and usually take up more space than available on even the biggest monitor. *Thus, the use of a separate message window for presenting critic signals does not scale up to large designs.*

DIRECTIONS FOR THE FUTURE

Despite the empirical limitations of our study, it allowed us to identify patterns of system influence on designers' activities that have implications for our proposed development process and for the design of future knowledge-based design support systems.

Improving the Development Process

Overall, our development approach seemed to work. This approach assumes that creating a cognitively ergonomic system involves emulating key aspects of expert designers' cognitive processes in the knowledge-base design. Following this process, we created a system that showed itself to be supportive and yet not too disruptive. The amount of critiquing activity varied according to the designers' skill level as expected (Table 1) and the system did influence designer behaviour in many positive ways.

However, unsurprisingly in retrospect, by basing our design on the processes of experts, we created a system that was better suited to expert designers. While the system appeared to support less-experienced designers to learn new rules, it did not encourage them to reflect on design trade-offs as we had hoped. A redesigned system should make more explicit the importance of this trade-off process. *Towards this end, we propose that our design process be modified: cognitive analyses of designers of differing skill levels should guide system design.* Specifically, constructive interaction techniques [10] where two designers think aloud while working together may be useful. Analyses of highly-experienced designers assisting less-experienced designers could be used to better understand how designers learn to "think like experts."

Rethinking the 'Critiquing' Metaphor

The design of VDDE was motivated by a theoretical model of design support as well as by specific domain requirements. We observed designers' activities to be generally consistent with the model of use shown in Figure 1. A surprising exception was how designers anticipated and sometimes avoided critic activity. According to Figure 1, designers analyse their design products *independently* of the critiquing analysis. Contrarily, we observed an indirect influence: designers were taking into account the existence of the system and also analysing their design from what they thought was the critiquing system's perspective. The experimental instructions to think aloud could have prompted this behaviour, leading designers to rationalise their actions and explain things in terms of the experimental situation; i.e., the critiquing system. However, we do not believe this to be the case. For one thing, others have observed similar anticipations. In one study of a computer-based coaching system, second-guessing what the system might do became many students' favourite activity [4]. These students tried to anticipate what actions would provoke the coaching system to respond. Instead of writing correct algebra expressions to avoid the system, students purposefully wrote incorrect expressions to get the coach to appear!

In our case, we felt the "critiquing metaphor" significantly contributed to this behaviour. As one highly-experienced designer noted, "inviting criticism is not in line with my personality ... my own style would probably be to try to get it right ... try to anticipate the critique." The metaphor seemed particularly problematic for less-experienced designers. Instead of considering the trade-offs presented by the critics (as we hoped), these designers appeared to adopt strategies to avoid critics they anticipated being in the highest priority rule set (even when it didn't make sense). *To create systems that people will listen to and not try to avoid, we may need to consider alternate interface metaphors.*

In our studies, all designers associated consistency criteria with the needs of the marketing group; however, only the highly-experienced designers associated the interface guidelines with the needs of the user. A possible redesign is a “design consultants” metaphor where stakeholder labels are associated with each critic rule set. A designer could control the rule-base by activating a ‘user consultant’ and a ‘marketing consultant’ that each argue for these different perspectives. While this proposal would somewhat anthropomorphise the rule sets, it may emphasise that designing involves making trade-off decisions that benefit some stakeholders, often at the expense of others.

Next Generation Interfaces

Eliminating redundant critics is an important, yet difficult challenge for future knowledge-base designs. Clearly, VDDE’s intervention strategies need to take into account whether they are being used individually or together and to enable overlapping rules only once. However, an even more challenging aspect of this redundancy problem is supporting rules to be locally disabled. If designers add rationale as to why they are breaking a certain rule, they expect that rule to be locally disabled and not fire again. Such functionality requires preserving information that keeps track of previous critiquing activity and making this information available to both the system and designer. For example, both parties need to be able to tell that the “no gaps in menu” rule has been rejected for the Listen Menu.

A promising approach is to ‘push’ this information into the design representation and integrate critic signals directly into the design solution. Instead of listing messages linearly in a separate message pane, the system could signal possible problems by annotating affected parts of the design solution. Either the designer or the system could check an annotation to determine which rules had previously fired and been rejected. Such a visual critiquing approach [14] would also overcome the scaling limitations of the separate critic message window and may even give designers more incentive to record their rationale for breaking certain rules.

In summary, we created and evaluated a knowledge-based design support system to better understand how these systems influence designers’ activities. Our findings showed that these systems do influence designers’ activities, but sometimes in unexpected and indirect ways. In addition to these findings, the contribution of this research is threefold. First, we demonstrated a promising approach to system development where cognitive analyses of design activities are used to guide knowledge-base design. Second, our findings suggests how a theoretical model of design support proposed in earlier research efforts [8] should be extended. Finally, by offering reflections on how our design could be improved, we hope to provide a starting point for the next generation of active, design support systems.

ACKNOWLEDGEMENTS

This research was made possible by the generous help of designers from the Human Factors Group at U S WEST Advanced Technologies. Special thanks to Simon Buckingham Shum, Peter Scott, John Domingue and Paul Mulholland for comments and discussions on earlier drafts of this paper and thanks to Alex Repenning for use of the “thinker” and ‘moral support’ for the VDDE project.

REFERENCES

1. Bhavani, S. K. and B. E. John, “Exploring the Unrealized Potential of Computer-Aided Drafting,” *Human Factors in Computing Systems (CHI ‘96)*, Vancouver, Canada (April 13-18), 1996, pp. 332-339.
2. Bonnardel, N., “Le Rôle de l’Evaluation dans les Activités de Conception [The Role of Evaluation in Design Activities],” University of Provence, Ph.D. Dissertation, Dept. of Psychology, 1992.
3. Burkhart, B., D. Hemphill and S. Jones, “The Value of a Baseline in Determining Design Success,” *Human Factors in Computing Systems (CHI ‘94)*, Boston, MA (April 24-28), 1994, pp. 386-391.
4. Burton, R. and J. S. Brown, “An Investigation of Computer Coaching for Informal Learning Activities,” in *Intelligent Tutoring Systems*, D. Sleeman and J. S. Brown, Ed., London, Academic Press, 1982, pp. 79-98.
5. Fischer, G., “Domain-Oriented Design Environments,” in *Automated Software Engineering*, Kluwer Academic Publishers, Boston, MA., 1994, pp. 177-203.
6. Fischer, G., “Turning Breakdowns into Opportunities for Creativity,” *Knowledge-Based Systems Journal*, Vol. 7, Iss. 4, pp. 221-232, 1994.
7. Fischer, G., A. C. Lemke, T. Mastaglio and A. Morch, “The Role of Critiquing in Cooperative Problem Solving,” *ACM Transactions on Information Systems*, Vol. 9, Iss. 2, pp. 123-151, 1991.
8. Fischer, G., K. Nakakoji, J. Ostwald, G. Stahl and T. Sumner, “Embedding Computer-Based Critics in the Contexts of Design,” *Human Factors in Computing Systems (Interact ‘93 and CHI ‘93)*, Amsterdam (24-29 April), 1993, pp. 157-164.
9. Harstad, B., “New Approaches to Critiquing: Pluralistic Critiquing, Consistency Critiquing, and Multiple Intervention Strategies,” University of Colorado at Boulder, Masters Diss., Dept. of Computer Science, 1993.
10. Miyake, N., “Constructive Interaction and the Iterative Process of Understanding,” *Cognitive Science*, Vol. 10, Iss. 5, pp. 151-177, 1986.
11. Nakakoji, K., T. Sumner and B. Harstad, “Perspective-Based Critiquing: Helping Designers Cope with Conflicts among Design Intentions,” *Artificial Intelligence in Design ‘94*, Lausanne, (August), 1994, pp. 449-466.
12. Rittel, H. and M. Webber, “Dilemmas in a General Theory of Planning,” *Policy Science*, Vol. 4, pp. 155-169, 1973.
13. Schoen, D. A., *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York, 1983.
14. Stolze, M., “Visual critiquing in domain oriented design environments: showing the right thing at the right place,” *Artificial Intelligence in Design ‘94*, Lausanne, Switzerland (August), 1994, pp. 467-482.
15. Sumner, T., “Designers and their tools: Computer Support for Domain Construction,” University of Colorado at Boulder, Ph.D. Dissertation, Dept. of Computer Science, 1995.