



# ITS and User Interface Consistency: A Response to Grudin

CHARLES WIECHA  
IBM T. J. Watson Research Center

---

In his letter Grudin asks “Where do we get the false impression that every good design must somehow be ‘consistent’?” It has never been our intention in developing ITS to force a conformance to superficial consistency. On the contrary, our primary achievement has been to provide a tool to develop excellent interfaces by iteration through successive prototypes, each of which is subject to use, testing, and revision. By capturing the results of these evaluations in executable style rules we have provided a tool for developers to benefit by reusing the interaction techniques developed by ourselves and others.

Below, I first answer a number of the detailed comments Grudin makes about ITS. Since I believe many of his comments are based on an inappropriate attribution of a stereotypical notion of consistency to ITS, I then briefly outline the sense in which we use the term consistency. This interpretation allows us to separate concerns about adherence to a style from concerns about the quality or usability of that style.

Let’s begin with Grudin’s assertion that we “argue for establishing a set of design rules to cover all allowable interaction techniques and then prohibiting exceptions.” The section he refers to, on page 233, is titled “Feedback into improved rules.” Its focus is not on prohibiting exceptions to rules, but exactly the opposite: how to capture the rationale behind new rules so that they can be incorporated into existing styles, again for the benefit of future developers and users. The section argues that to make such exceptions in an ad hoc manner, and not to feed them back into improved rules, “would miss an opportunity to add the missing knowledge. No one else would benefit from our work.”(p. 233)

---

Author’s address: IBM T. J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1992 ACM 0734-2047/92/0100-0112 \$01.50

ACM Transactions on Information Systems, Vol. 10, No. 1, January 1992, Pages 112–114.

Grudin's letter goes on to make a number of comments about the role of tools such as ITS that represent and automatically apply rules derived by human designers. These comments seem to fall into two groups: (1) rules can't express the complexities of realistic interfaces such as Interleaf, and (2) rules have no role when human designers can't agree on an "optimal" design. Let's consider each type of comment.

As examples of the type of complexities found in real systems, Grudin outlines five general rules for menu defaulting in Interleaf:

- (1) Default to the last item selected unless overridden by another rule.
- (2) When items are logically linked (e.g. Copy and Paste) default to the related item even if it is on another menu card (as in this case).
- (3) Default to a static item when the operation is dangerous.
- (4) Default to a menu related to the current context if the user has gone on to an unrelated operation.
- (5) Add an option to toggle if there are exactly two selectable choices.

These rules are certainly within the scope of ITS, and are not supportive of Grudin's assertion that "not all design cases can be prespecified or translated into rules that contemporary systems can act upon." He is correct that ITS cannot automatically determine when items are logically linked, or dangerous, and relies on interface designers to make these determinations. I fail to see how this reduces the usefulness of such a tool.

His second point, that there are "cases for which the rules governing optimal design are unknown" is surely true, but why does it follow that "In these situations formal systems such as ITS have no role"? Again, ITS is a tool for representing design rules. Whether those rules are optimal in some sense, or how many rules are required to represent the design, is simply not relevant. It may take more rules to represent QWERTY than Dvorak, but both are possible in ITS.

Finally, let's consider what I feel is leading to these misunderstandings, the assertion that we intend ITS to be used to generate consistent interfaces according to the conventional notion of consistency. Grudin objects that "the claim that [ITS] will produce 'consistent' interfaces is a harmful error. It fosters the common confusion between consistent and good interfaces." Our definition of consistency in the paper is as follows:

Consistency means that similar functions in an interface, such as choosing a single item from a list of mutually exclusive options, are represented and interact with the user similarly wherever found in an interface. Consistency does not mean that such a function is identical throughout the interface, just that where appropriate the technique is similar and where differences exist, those differences have meaning. (p. 216)

This clearly states that the issue is one of deciding on the appropriateness of differences in terms of the need to communicate an underlying variation of

meaning. Nowhere do we fail to distinguish between consistency and quality of interfaces. This is in fact why I cited Grudin's Interleaf examples. I argued that Interleaf should be seen as an example of a consistent interface given a definition of consistency that can tolerate the notions of a spectrum of defaults and meaningful overrides. Grudin seems to believe in the idea that such overrides are necessary for good interfaces. I simply want to grow the concept of consistency to include them.

I share Grudin's dissatisfaction with the commonsense meaning of consistency. However, it is not appropriate to apply that meaning to ITS. The important issues are in fact related to the appropriateness and quality of style rules in addition to the level of conformance to them found in an interface. ITS remains an important tool for developing and evaluating a continually growing set of those rules.