Interface and the Evolution of Pidgins: Creative Design for the Analytically Inclined*

Thomas D. Erickson
Human Interface Group
Apple Computer, Inc.

There are many methods of pursuing design. The question of how to get new ideas, or how to approach an idea from a different direction, is one that designers must repeatedly answer. There are dozens of books on visual thinking, drawing, lateral thinking, creative blockbusting, and other ways of better using your neglected right hemisphere. There are even books on how to whack and kick yourself into that crazy, flexible, and fun, creative mode. They're all carefully crafted to coax you away from your comfortable, cautious, rational approach to problem-solving.

But what if you don't want to go?

What if you're unrepentantly analytic? What if your idea of having a good time is to read the dictionary, looking for words you don't know? What if your verbal and math test scores are off the scale, but your attempts at realistic drawings resemble those of two-year-olds? What if your right hemisphere was irredeemably damaged in a tragic childhood accident? What if you're inhibited, inflexible, and don't want to look silly? What if you've just got a bad attitude?

Is there no hope?

There is hope. This chapter describes a method of pursuing creative design that is pleasing to the analytically inclined. To the best of my knowledge, it has no widely known name, since no best-selling books have been written about it. However, for the purposes of this paper, I will call it design by symmetry.

This chapter plays two roles: it illustrates the conceptual approach of design by symmetry; and, in the process, it tosses out some ideas about how the Macintosh interface might evolve.

Design by Symmetry

Most approaches to creativity in design advocate throwing out rules and juxtaposing concepts that are essentially different in the hope that a new point of view, or a new concept, will serendipitously emerge [see Mountford, chapter for examples]. In contrast, this chapter describes the technique of applying extended, precise analogies to design problems. Because the term analogy can be used to

* (Published in *The Art of Human Computer Interface Design* (ed. B. Laurel), pp. 11-16, Addison-Wesley, 1990.)

refer to anything from the vaguest likeness to a mathematically precise relationship [see Polya, 1957], and because the extended and rigorous nature of the analogies pursued are central to this technique, I shall use the word "symmetry" to designate an extended, highly precise analogy.

Design by symmetry works by juxtaposing concepts that are similar at a very deep level–the concepts are symmetric in terms of some deep structure or underlying process. Once the underlying symmetry is established, the designer attempts to extend the symmetry farther, using what is known about one domain to suggest new ideas about the other. Where traditional approaches to creative design require playfulness and produce new ideas via inspiration, design by symmetry requires analysis and produces new ideas by extrapolation.

This is all rather abstract. Let's look at an example of how design by symmetry works on a real problem. The problem is one that, as you may imagine, many people have been worrying about: how ought the Macintosh interface evolve so as to accommodate the increasing demands being made of it?

Symmetry: The Macintosh Interface as a Language

In the late 1700's explorers, traders in sandalwood, and immigrants began to make contact with the members of the oceanic culture of the South Pacific islands. Sometimes the contacts were fleeting, as in the case of explorers, and sometimes more permanent, as with immigrants and traders. These contacts are of interest because they required the members of two radically different cultures to communicate, even though there was no common language. A language had to be invented. Such languages are variously known as trade languages, contact vernaculars, or pidgins.

Pidgins–which have developed in many times and places throughout the world–share a number of characteristics. First and foremost, pidgins are easy to learn. Ease-of-learning is the raison d'etre of a pidgin; its users are typically those who lack the time or inclination to learn a language–they simply want to get on with business. The other characteristics shared by pidgins are clearly related to their ease of learning. Pidgins employ only simple sentences, with very regular, if awkward, syntax. For example, a plural of a word might be formed by repeating it. Pidgins have no tenses–all statements are present tense. Pidgins have a very limited vocabulary, which may be buttressed by the use of pointing and other gestures. Finally, pidgins typically allow their speakers to deal with a very narrow segment of the culture. Though you can't discuss philosophy, make puns, or express the subtilties of the extant culture using a pidgin, you can bargain for food, trade for goods, or make travel arrangements.

What does this have to do with the Macintosh? The Macintosh interface is very much like a pidgin. First and foremost, the Macintosh interface is easy to learn. Ease of learning was the driving force behind the development of the Macintosh interface; it was targeted at users who lacked the time or inclination to learn about a computer–they simply wanted to get on with business. Like a pidgin, the Macintosh interface has a simple, noun-verb, syntax: first you select the object, then you specify the action to be carried out. Until relatively recently the Macintosh interface lacked tense. That is, everything that could be done on the Macintosh took effect immediately. Only with the

advent of the “Set Startup” command, where actions to be taken on future startups can be specified, did the Macintosh make the first tentative steps towards including the idea of tense. The Macintosh interface also has the limited vocabulary characteristic of pidgins, and is extended by using simple pointing and dragging gestures. Finally, like a pidgin, the Macintosh interface has distinct limitations in its communicative power–you can get your basic tasks done, but that’s about it.

I have made the case for the existence of an underlying symmetry between the Macintosh interface and a class of simple languages called pidgins. The next step is to examine other characteristics of pidgins, and see what sort of insights they can provide about the Macintosh interface.

Extending the Symmetry

Pidgins have another characteristic: under proper circumstances, pidgins spontaneously evolve into a more complex type of language called a creole. This is particularly interesting vis a vis the Macintosh interface, because the characteristic ways that pidgins evolve into creoles may tell us what properties a linguistic system–or an interface–must have for it to become a powerful communicative device while remaining relatively simple and easy to learn. First we’ll look at when and how pidgins evolve into creoles, and then, extending the symmetry, we’ll examine some ways in which the Macintosh interface might evolve.

The evolution of pidgins into creoles is relatively rare. It thus behooves us to look at the conditions under which such evolution does occur. Creoles develop in two types of situations: pidgin speakers may be deprived of the opportunity to use their mother tongue, or pidgin becomes so widely used in the community that it becomes natural to use it in the home. The first situation often occurred as a result of the slave trade in the Caribbean, when slaves from the same areas were deliberately separated to reduce the possibility of an uprising. In more recent times, rapid modernization, such as in New Guinea, has resulted in the intermingling of such a large number of linguistically distinct groups that no single one emerges as dominant, and a pidgin becomes an important, and high status, community language. These and other observations suggest that the prerequisites for the evolution of a pidgin into a creole are: 1) lack of a fully-developed common language 2) lack of a dominant, ‘full-featured’ language that could become a common language; 3) cultural pressures for a more complex communicative system. It is worth noting that parallels to these pressures exist in the computer world. There is clearly no interface lingua franca, nor even a clear candidate for one. Yet there certainly do seem to be pressures for more power in human-computer communication.

Now let’s look at how a pidgin evolves into a creole. One of the most fascinating aspects of the change from pidgin to creole is that–regardless of the sophistication of the adults–the transformation from pidgin to creole occurs only in the speech of pidgin-speaking children. Adults who are pidgin speakers remain pidgin speakers for life, even in the presence of children who have become creole speakers. It is this spontaneous emergence of a more complex language, from the mouths of babes, as it were, which suggests that the features of creoles are fundamental to a powerful, yet simple, communicative system. The following features spontaneously emerge as part of the pidgin to creole evolution: complex syntax–for example, devices for marking relative clauses; past and future tenses; and a much larger vocabulary.

The next step is to consider how these characteristics of creoles map onto the Macintosh interface. Clearly the Macintosh interface has a very simple syntax, essentially noun-verb. The appearance of more complex syntax in pidgin to creole evolution suggests that a more complex syntax is not necessarily an evil. For example, one element of syntax which emerges during the evolution of a creole is a relative clauses mechanism. While this could be interpreted in several ways, one possibility is that since relative clauses allow speakers to embed comments in a sentence, a general means of annotating objects would be a useful way of extending the power of the interface without over burdening the user.

The appearance of tense in creoles is easy to extend into the Macintosh interface. Time should become an integral part of the interface. For example, it should be possible to issue a command that is for some time in the future–whether it be tied to a particular event, such as start up, or to an actual time,as in: 'Call up applelink and download my mail at 3 am every morning.'

The expanded size of a creole's vocabulary results from a number of different mechanisms. One way creoles expand is that new words are coined. Obviously, new words mean new actions and new objects in the Macintosh interface. It also suggests that a useful addition to the interface would be a way of dynamically creating new vocabulary; that is, an integrated macro capacity. Such a capacity ought to provide for the creation of both objects and actions, and allow them to be used just as existing objects and actions are used. Another way a creole's vocabulary expands is that the meanings of existing words become more general–they are used in a wider variety of situations and for a wider variety of purposes. This suggests that a natural way of extending the interface is to transpose familiar interface objects to new contexts, where they may play similar roles.

This symmetry could be extended much further. Our treatment of the roles of syntax, tense, and vocabulary expansion in the Macintosh interface has been cursory. There is also more that is known about pidgin-creole evolution [e.g., see Holm, 1988] which could be extended into the Macintosh domain. However, for the purposes of illustrating design by symmetry, our example is sufficient.

Note that many of the specific features suggested by this symmetry are already appearing in the Macintosh interface; for example, the "Set Startup" command corresponds to an instance of allowing tense in the interface. But it is only one instance of tense; many more can be imagined. The real value of the design by symmetry method in this case is that it provides a framework for the evolution of the interface.

Properties of Symmetries

Design by symmetry has three characteristics which distinguish it from most other methods of fostering creativity in design. First, as noted above, design by symmetry tends to produce a conceptual framework, as well as new ideas and expressions. Since one of the banes of interface design is creeping featurism–the proliferation of individually desirable, but collectively

unmanageable features–a method which encourages a coherent approach to adding features is surely of great value.

Second, design by symmetry is ideally suited for cross-disciplinary work. The Macintosh-interface-as-language symmetry draws upon work in linguistics and anthropology for its substance. Other symmetries may draw on other disciplines. For example, Susan Brennan in her chapter examines the implications of a symmetry based on research in psycholinguistics: interface-is-conversation. She explores how what we understand of human-human conversational interaction may be extended to human-computer interaction.

Third, once a symmetry has been discovered, it has the potential of being extended in many directions. For example, the Macintosh-interface-as-language symmetry could also be extended in a different direction. It would be interesting to study the cultural conditions which promote the evolution of creoles; this might shed light on conditions which would foster the evolution of computer interfaces. Lest the notion of cultural effects on interface evolution seem entirely academic, note that one reason for the continued consistency of the Macintosh interface across products from thousands of developers is that the Macintosh 'community'–end users, developers, reviewers–has bought into the ideas of ease-of-use and consistency. It is likely that sociologists and anthropologists could discover quite a bit about what sort of things might promote the health of the 'Macintosh culture', and thus the health of the interface.

Creativity in Design

Design by symmetry is just one method of fostering creativity in design. In spite of my tongue-in-cheek opening comments, I think that all methods–be they left-brained, right-brained, or harebrained–can be of immense value to the designer. New ideas and new perspectives are always in demand.

References

Holm, John 1988 Pidgins and Creoles, Volume 1: Theory and Structure Cambridge: Cambridge University Press.

Polya, G. 1957 How to Solve It, second edition. Princeton: Princeton University Press.