



JUS

JOURNAL OF USABILITY STUDIES

Vol. 7, Issue 2, February 2012, pp. 51-55

---

# Curating Evolution

## **Austin Henderson**

Principal  
Scalable Conversations  
644 Cragmont Ave.  
Berkeley, CA  
USA  
[henderson@rivcons.com](mailto:henderson@rivcons.com)

## **Jed Harris**

Principal  
Scalable Conversations  
978 Cragmont Ave.  
Berkeley, CA  
USA  
[jed@jive.com](mailto:jed@jive.com)

## **Abstract**

As development cycles shrink, and communication with users gets easier, our products will be increasingly the result of collaboration between designers and users. Already we are seeing this practice in open source software, wikis, blogs, and social media. As product development changes, the role of the designer must change too. In addition to their current responsibilities, designers will join users in supporting and curating changes, and most importantly, will provide the structures for users collectively to achieve adequate convergence in the face of disparate yet interacting needs.



Copyright © 2011-2012, Usability Professionals' Association and the authors. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. URL: <http://www.usabilityprofessionals.org>.

## Introduction

Consider the meanings of words. Since language was invented hundreds of millennia ago, the meanings of words have continuously evolved. Importantly, this collective evolution has been achieved, not through premeditated coordination, but rather through exploration and convergence in use. It is a buzzing, booming cauldron of meanings, yielding effective, drifting, locally-coherent yet globally disparate support for human linguistic engagement.

Then, in the sixteenth century, printing made reading and writing far more common. As a result, there was great concern among the well-educated that language would devolve in the hands of masses. The keepers of language invented the dictionary to maintain coherence in word meaning. The Johnsons and Websters positioned themselves as the arbiters, indeed the designers, of the meanings of words. However, the actual meaning of words continued to evolve in everyday use largely as they had before widespread literacy and dictionaries.

Later, in the nineteenth century, some dictionary writers recognized this inherent tension between meaning based in definition and meaning based in use. In what may be thought of as a user-centered move, word meanings in the Oxford English Dictionary (OED) were explicitly grounded in actual usage. Readers were invited to submit quotations of words from written works; that is, words as they were actually being used. The OED compilers organized these quotations to reflect the evolving meanings of words. The language rolled on, with the dictionary tracking its evolution, contributing to the self-awareness and coherence of language users without presuming to exercise control.

Now, in the twenty-first century, we have the ability to harvest an instant history of how a word has been used from Google Books. We also have the crowd sourced Wiktionary, a “collaborative project to produce a free-content multilingual dictionary. ... You can edit it.” The users of language—the speakers, readers, and writers—are also its legitimate and authoritative documenters, the dictionary creators.

What has happened to the language arbiters, the keepers of the flame of language? They continue to play two important roles. They curate the process, writing, editing, and critiquing the use of language. And they provide the infrastructure for widespread use and participation in online language resources.

Thus, over time, we’ve come full-circle in determining the meanings of words. Since prehistory, the evolution of the meanings of words was the province of every speaker. Relatively recently, it became the province of a specialized few, then the contributions of many organized by the few, and now again the province of us all. Further, because contributing to a dictionary these days requires skills and access that are widely available, there are a growing number of participants. We can all play, and have our own dictionaries if we want them.

Our point? We want to argue that all areas of design are likely to follow a similar path and reach the same end. These changes are already underway in most areas, particularly those that are based on computation. And we want to argue that there is a very important role in these ever-more-user-produced systems for those with a gift for order, good design, and usability.

## Evolvable Systems—The Only Kind

Historically, like language, all areas of human practice were directly shaped by the needs of practitioners. Accounting, message addressing and delivery, the management of employees, and so forth were managed through a locally idiosyncratic but mostly conventional set of practices that evolved as they were used and adapted to circumstances.

As aspects of these processes were increasingly supported by computers, this changed. The computer tools of the later twentieth century were very rigid (except under the hands of the few who could write code). Applications for accounting, mail, human resources, and so forth, had to be spec’d, designed, built, and deployed by specialized groups. These applications could only handle very limited complexity and nuance and were difficult to modify once created. However, the practices that these rigid tools supported were no different from language: their usage was embedded in life, and life went on—changing and demanding response. Despite whether dictionaries evolve, language use evolves. Despite whether computer-based products evolve,

the life and practices they support evolve. In the face of new needs and applications that are misaligned with those needs, users muddle through by inventing work-arounds. Even if the technical systems can't, or won't, change, the socio-technical systems (technical systems in use by people) evolve and diversify.

In response to this continuing pressure, tool-builders are beginning to make the same moves that the dictionary-builders made. They are opening tool-creation to the participation of users, either by incorporating suggestions on design or by making tools malleable in the hands of everyone. Releasing control even more, tool-creation practices are being moved from protected private centers into the public domain. Even the goal is changing: products are designed to branch into a host of variants, based on plug-ins and add-ons, generating a braided stream of interdependent pieces in endless configurations. There was only one Multics; there are many Linux's. And every service is different in each installation.

Many years ago, one of us (Henderson) evolved a design tool (Trillium) for UI designers at Xerox. Designs were created by "wiring together" interactional widgets (we called them *items*, and each item had an *itemtype*). Initially, we thought that we could find a collection of itemtypes that would work for everyone (at least those making copier interfaces), and then we would use them as the Xerox design language. A central learning was that one language does not fit all. Each product line—indeed each product—tended to have variants on the itemtypes. Our response was to let go of the itemtype definitions, open them up to everyone, particularly the designers. We evolved Trillium to support users in creating itemtypes. Trillium's user support for this was far from perfect and provided only fairly simple constructions, but it worked well enough. The resulting languages of design used by different groups could then evolve, and branch, and (through sharing) merge.

The Trillium creators, almost from the beginning, delegated creation of the design language to the community of practice of Xerox UI designers and supported them through language design tools. Furthermore, we organized and managed the essential context of user design: a community-wide conversation about the use of Trillium, the design of UIs, and the design of design languages. That context must enable participants to become aware of the conversation; find out who else is "there," determine what the discussion is about, why it is being held and to what end; determine how the discussion is being shared, both with those there and those not; enter, be present, and withdraw in socially-appropriate ways; present themselves and negotiate their self-presentation; develop standing and relations (e.g., trust, friendship) with others; and all the other nuanced interplay of multi-perspective, inter-personal interaction. We shaped and maintained this context through the practices of using mailing lists and file systems, coordinating community-wide gatherings, and introducing and nurturing video-based distance collaboration.

Trillium was an early example of this pattern of community-wide evolution of local configurations, but since the broad adoption of the internet there are many more. Version Management Systems have rapidly evolved to support diverse local configurations while still enabling consistency and sharing. Wikipedia has evolved a rich set of practices to generate an excellent encyclopedia from the "drive by" contributions of a huge community. A wide range of peer production projects have benefited from substantial user participation while improving both their design quality and their scope.

### **Drive the Bus or Get Run Over**

We have a long way to go on all these fronts, but the direction is clear. Our working – and playing – environments are evolving, multi-perspective, negotiated. To stay relevant, our system designs have to become more evolutionary, multi-perspective, negotiated. Every deployment will be different.

This is not new. Traditionally, we have made our rigid tools as broad and flexible as possible, and hoped that would do. We have recognized that we cannot keep up with all the needs of all our users all the time. We have implicitly relied on users to adapt the tools to their circumstances. And they have.

However, now it is time to recognize that we need to support those users in all that adopting and specializing and aligning and working-around. We cannot anticipate it, so we must help them do it themselves.

This affects the design of the tools, but more importantly it affects how we support the use of those tools. If work-arounds are part of users' practice, we must support work-arounds. If modifying the tools is part of their practice, we must support modification. In short, the environment of use must be seen as encompassing all the work of evolution. Instead of thinking of our design work as ending when use begins, we must support evolution of our designs in the midst of use.

Getting this right will be a huge competitive advantage. It recognizes the realities of disparate, negotiated practice. Rather than denying them, it takes advantage of them. Most powerfully, it extends the work of designing to all the users. What a design workforce! Real experts! And it can lower the bar for some designs out of the gate: just start them so that they are good enough for a big enough community, and then join with that community in building both the products and the community. Tomorrow is built on today. Perpetual bootstrap.

### **Curating Evolution**

Inevitably, users, driven by their own local needs, will tend to evolve their tools in ways that conflict with the needs of others. Indeed, they may not even know of those others or their needs. The tools are likely to evolve in diverse, even incompatible directions. As a group, users may ride off at high speed in all directions.

The role of design is to accelerate evolution. When evolution is in all directions, the role of design is to accelerate the practices of coordinating those directions, bringing coherence where necessary.

However, here again we must resist the temptation to take matters into our own hands. If coherence is needed, then it is needed by the users, so help them to create it. They will be better than we at determining where coherence is necessary. Let them negotiate it out, suiting the solutions to the circumstances at hand.

Let achieving coherence be part of the work. As the work evolves, ways of achieving coherence will also evolve. Because design accelerates evolution, we—as designers anticipating this buzzing-booming cauldron of evolution—should be supporting the designing, by everyone, of tools to achieve coherence.

### **Making It Happen**

All very well, but in practice what will this take? A few thoughts:

- Bring users into the design process: That's a no-brainer, because that was the forcing function in the first place. Engaging the user may seem straightforward; letting them share or own control may be harder.
- Give everyone a place at the table: As we look around organizations, we find that the design activity is not easy to get our hands on. It is diffused through groups and documents. It is off the radar as a thing of interest. So first, we often need a table—a recognition that the design is evolving and that the evolutionary practice itself needs care. And then everyone needs to have a place at that table. Often, we leave out more than the users—we leave out support, or documentation, or marketing, or sales, or even corporate management.
- The focal point of evolution is actual use: The design table must extend to the point of use. The user must be able to shout out when it hurts, or delights, or when they have a better idea! And all of those at the table, including other users, must be able to hear and respond.
- Agreement is highly overrated; disagreement is the place that progress is made: Diversity is great. Encourage conflict. Don't necessarily seek solutions that make everyone happy. Perhaps many interacting solutions are needed. Complete consistency is dangerous; after all, the world is changing and therefore inconsistent with what it

was yesterday. Coherence is better—just enough alignment for the purposes at hand, and no more.

- **Trouble is your friend:** Trouble is the world’s way of calling your attention to what you have to know. It may not be that the product has difficulties, it may be the alignment of that product with the world. Your support folk are the most important part of your program. They must be the best listeners on your team. Keep them very close to the table.
- **Let language drift:** The practice that people are engaged in is evolving. Therefore the language a community uses for talking about practice must be evolvable. Let users reflect this in your (their!) tools. Use language as an evolvable tool for thought.
- **Embrace ambiguity:** Ambiguity provides for saying things well enough to get on with the job. Ambiguity is the oil of thought. Ensure that oiling is part of your DNA.
- **Let context frame content:** In conversations, people exchange content. However, the context in which that exchange takes place is equally, often more, important. Curating the context is more important and harder than curating content. However, same tune again, getting started is often not so hard at all. Then it is just a small matter of enabling everyone to evolve the context along with the content.
- **Acknowledge values:** Curating is all about values. Curating evolution goes beyond competitive advantage; it will honor such conversational values as pride of creation, sense of belonging, having voice, openness to diversity, sense of control, openness to change, and responsiveness to a range of contexts.

## Conclusion

Curating evolution is not an easy job. There are strong forces arrayed against it. All those who feel more secure with consistency (engineers, programmers, designers, managers) are likely to be appalled by the prospect of a loosely-coupled evolving design process.

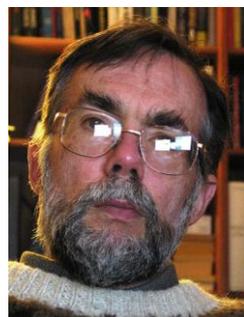
Yet change will out. We argue that curating evolution is much better than trying to deny it or stamp it out. And in the end, a lot more fun.

## About the Authors



### Austin Henderson

Austin Henderson’s 45-year career in Human-Computer Interaction includes user interface research and architecture at Xerox, Apple, and Pitney Bowes, and strategic design with Fitch, Rivendel, and Scalable Conversations. Henderson seeks to enable designers and users to better evolve tools to respond to needs in a rich and changing world.



### Jed Harris

Jed Harris has worked in computer systems design for more than forty years, at Doug Englebart’s Augmentation Research Center, Alan Kay’s Learning Research Group, as Director of the Intel AI Lab, and a Distinguished Engineer in Apple Labs.