Title
Instructions for Design and Designs for Conversation

Author
Paul Pangaro, Ph.D.
CyberneticLifestyles.Com
200 West 58th Street Suite 5B
New York, New York 10019 USA
+1 212 974 1066 phone +1 415 902 8020 mobile
paul@cyberneticlifestyles.com

Abstract
Effective conversation occurs when beliefs are negotiated through interaction and evolve in a framework of goals. Concurrently, goals are negotiated and evolve through interaction. Such conversations are processes of design. Similarly, design—where proposed constructions are negotiated and evolve toward goals, while goals for the design are negotiated and evolve—is a process of conversation.

Outcomes from such activities are greatly facilitated, in my view, from an understanding of Pask's two main frameworks from conversation theory: a structure for the architecture of conversations and a schema for modeling the evolution of conversations. My conviction is that these can be used to mirror how design happens. This begs the question, can it help designers design? Yes, I believe it can. This paper offers a review of Pask's frameworks and discusses their application in theory and practice.

Keywords
Conversation theory, Gordon Pask, THOUGHTSTICKER, thoughtshuffler, bio-cost, design, user interface design

Structure of the Paper
Sections of the paper have two parts. Each first part addresses the premise of the paper regarding the application of Pask's work, as outlined in the abstract. Each second part relates personal experiences in attempting to design cybernetic conversations to make such applications successfully. For example:

I. Gordon Pask is the primary source for 'conversation theory', but he is by no means the only one. Dionysius Kallikourdis, Bernard Scott, and other collaborators were present when the papers were published that I consider the vital core. [Pask et al 1972, Pask et al 1973, Pask 1975].

II. A secondary (and personally idiosyncratic) reference is that of Italo Calvino's *Mr Palomar* [Calvino 1986], a collection of short stories about the experiences of, who else, Mr Palomar. Our protagonist desires to experience cybernetic thinking in all its richness; yet he constantly encounters a daily life that contradicts cybernetic convictions. Similarly, this author has had a career of applying conversation theory that has not always been smooth. Yet, like Mr Palomar, he has not lost his convictions.

Introduction

I. Design is conversation and conversation is design.

That is, *effective conversation*—where beliefs are negotiated through interaction and evolve in a framework of goals, just as goals are negotiated and evolve—is a process of design. Similarly, *design*—where proposed constructions are negotiated and evolve toward goals, just as goals for the design are negotiated and evolve—is a process of conversation.

The premises of this volume of collected papers require, in my view, a spotlight on Gordon Pask's two main frameworks from conversation theory—a structure for the architecture of conversations [Pask 1975] and a schema for modeling the evolution of conversations [specifically, entailment meshes, see Pangaro 2001]. My conviction is that these can be used to mirror how design happens. This begs the question, can they help designers design? Yes, I believe they can, in two senses.

In the first sense, processes of design—at their most general, these comprise iterative convergence on goals via the construction of prototypes of increasing fidelity—yield to cybernetic analysis. In [Dubberly & Pangaro, in press], models from cybernetics, including conversation theory, are shown to be homologous [Beer 1966] to issues foundational to design: goal-development, system constraints, testing and iterative learning; politics, rhetoric, and agreement [Rittel 1973]. This approach to design is a delight and Dubberly has been a consistent, rigorous force for connecting models from cybernetics to the design process (his métier as a design planner). In this context, conversation theory is a critical, crowning methodology of cybernetics because it closely tracks the most complex, complicating, ineffable, and sometimes intractable aspects of systems, their design, and their taming. It is enlightening to grasp the analogy between a model for the necessary elements for agreement to be achieved and, say, the process of a team comprising the expert on the business, the expert on the market, and the expert on the expression of a novel idea, all collaborating on the creation of business value. Using cybernetic models to frame problems and progress on design projects is the core idea of a class that Hugh Dubberly and I co-teach at Stanford University each year [Dubberly & Pangaro 2006]. Thus, the first sense of how conversation theory helps designers to design is by providing models that expose a system's capacities and capabilities, and therefore their limits, where 'the system' may be a product, a service, or the team involved in the design of products and services.

About the second sense of how conversation theory helps designers to design, I can only say that I use conversation theory to think about any interaction problem I may be focusing on, whether it's training complex decision-making tasks; avoiding unnecessary expenditure of bio-cost [explained below] while using computers; allowing users to have goals when using software; or, creating a web search experience that doesn't feel like using a typewriter. I won't say it makes me a better designer; I believe that's the case but there is no way to reliably test it. However, I'm certain that conversation theory can map the cognitive transitions that are the core value of using computers—or any human

activity, for that matter—like no other theory or model I know. Later in this paper I will explain those interaction examples in more detail.

II. First, a Mr Palomar-like story: At dinner with a close family relative and his new girlfriend, I found myself in an exchange about the tenets of cybernetics and their value. I hoped to persuade of their utility by bringing up major frameworks of cybernetics, and specifically conversation theory, that aid understanding of how systems negotiate and achieve their goals. As a result of my effort, she said dismissively, 'So, cybernetics, it's *just* a methodology?' [emphasis hers]. I ate my cake in silence.

Trends

I. If I've understood anything about human action from conversation theory, I've learned that we humans create and evolve our goals in real-time and in parallel to our actions and to other goal-related 'mental processing'. We negotiate with ourselves and others in attempts to lessen or increase some attributes of desired outcomes above others. This always involves effort-minimization trade-offs that my colleagues and I have called 'bio-cost' [Dubberly, Maupin & Pangaro 2004]. Bio-cost is the human cost—the energy, time, attention, and stress—required to get what we want. Humans use conversation as an explicit, often conscious medium for minimizing bio-cost, whether by thinking about different strategies to achieve a given goal, or negotiating with others to share goals and concomitant bio-cost to achieve them. (Conversation theory makes no structural distinction between these two modes of conversation: internal to a 'person' versus 'between persons' [Pask et al 1973], see comments below about 'P-individuals'.) This frees up energy and 'head space' for other activities, including the creation of more strategies to achieve more goals with more efficiency. Bio-cost reduction is thereby seen as an evolutionary strategy beneficial to survival.

You're asking about future trends? I consider it inevitable that the disciplines of user interface design, interaction architecture, usability research—please add all phrases here that refer to the crafting of systems, products, and services built upon technology—all such disciplines will incorporate constructs (whether derived immediately from conversation theory or other, less-direct, and probably less-efficient methodologies) that explore the role of conversation, its efficiencies and effectiveness, its failures and its aesthetics. I not only believe this, I behave as if I believe it.

II. For the last few years I have been initiating conversations with major search engine companies, the biggest names that you would know, the ones you probably use every day. Naturally they claim their focus is 'to improve the search experience', which I find immediately absurd: why improve searching—who wants to spend time searching, anyway? Why do they focus on improving something that is merely a means to a more important end—that is, can we focus at least on *finding*? Or skip ahead further, how about focusing on improving *understanding?* We certainly want to know more or to understand more than when we started—if we're in the same place at the end of the journey, then what was the point?

I confess that I hesitate to say that 'understanding' or 'a new cognitive state' is the goal of any and all of these activities; I prefer to state our final goal as that of *effective action* [Stafford Beer's definition of cybernetics: 'the science of effective action'.]. However, this seems to eliminate the pursuit of knowledge for its own sake or for browsing the web as entertainment, neither of which I want to deny. In the context of search engine revenue, the definition of 'effective action' is 'making a purchase', because that's how a user often achieves the goal of searching—and how search engines companies will earn revenue, at least in the current internet marketplace. I don't want to be accused of being too mercenary, but shopping is a valid pursuit—and if 'improving search' is to mean 'better shopping outcomes for consumers, and more revenue for merchants', how can search be improved? [See Pangaro 2006 for proposals.]. And how can I make a pitch to a big search engine company that is appealing?

My approach—a failed approach, mind you, because no commercial company has yet taken me up on the offer—is to open with the line, 'Conversation *works*—so, what's the implication for search?' This is at best an engaging 'hook' to begin an exchange about the goals of searching—not finding, but learning. (Well, not just learning, but acting. Actually, in the course of learning and acting, *we are becoming someone new*—because our beliefs have changed—and for whom effective action becomes clear.).

In every conversation with search engine companies, my approach led no further than a live demonstration of 'thoughtshuffler', a recent incarnation of Paskian interaction, at least in my interpretation, based on a simple but fresh (read: at-first-bewildering) user experience [Pangaro 2005]. However, the impediment to getting them to 'buy' my technology was not the ideas or the demo, but rather the organizational structures and processes inside of these search companies. In short, my approach was not received as a valid research agenda, because it did not match the expertise of current research teams: machine learning, data mining, natural language processing, or ad-placement optimization. In a real sense, the researchers could not *see* that thoughtshuffler was worthy of their attention—it was seen as a 'user interface' rather than a revolution in interface. Therefore it was relegated to the product team, where the user interface design is simply 'tacked-on' to the all-powerful technology invented by the programmers, who rule (product managers would deny this, but it is still true).

On my approaching the product teams, usually over a rushed lunch in a noisy [but tasty] cafeteria, their response was, 'Well, thoughtshuffler is potentially interesting, can you implement it in 6 weeks?'—that being the product release cycle for changes to the public site. 'Of course, this is impossible', I offered meekly. 'How about 12 weeks?', was their conciliatory response. But there was nowhere for this conversation to go. I sat in silence and continued to eat cake.

Solutions and recommendations

Having made the claim that I have usefully used conversation theory to aid designs, it is incumbent on me to explain how. I offer a few detailed descriptions of prior work, with roots deep in conversation theory.

Thoughtshuffler Software

I. I've just mentioned the overall approach to 'conversational search' and thoughtshuffler. Next I describe a bit about the concepts behind thoughtshuffler in relation to conversation analysis and evaluation.

Today's search experiences are based on highly-limited concepts of computer science that have dominated for 50 years: the techniques of 'information retrieval' (IR). We begin with extracting repeated words and phrases from a corpus of text, and placing them into an index. Then—in the most modern, internet-based expression of this ancient technique of IR—we provide the user with a text box in which to type 'keywords' that are matched to phrases in the index. Finally (for this is as far as today's search engine's go), we see displayed all possible pieces of matching content for the user to comb through. Ha! How helpful is that?!

The good news is, the user is shown where his/her typed phrases can be found in the content that has been indexed. The bad news is, the connection between the found content and the user's understanding and goals is likely nil. Such was the case with the early web search engine success called AltaVista, successful because it was reasonably fast and could retrieve results from a large number of web pages—at a time when the web was relatively small.

It was up to an upstart to upset the prominence and respect that AltaVista earned by adding a *ranking* algorithm on top of all the above. Google added the notion of 'page rank', that is, a measure of the 'quality' of a page, based (in its early days) primarily (if not solely) on the 'popularity' of links *to* a given page. This means that, if my web page has more other web pages linking *to* it, more human beings making those links think I'm worth linking to, and so my page is 'better'. The good news is, the likelihood of getting a page with decent content on it was vastly improved (at least it was in the early days). The bad news is, the page may still have *nothing whatever* to do with the user's understanding or interests or goals or anything.

For that, we need conversation, history and memory, relationship, and more.

(Hence my motivation for bringing conversation to the web. But let me add a giant caveat here: when I say 'bring conversation to the web', I'm *not* talking about natural language parsing; that brings us back to a computation solution and an investment in programming rather than theory. We've had 50 years of programming to build 'natural language interfaces', and where's it gotten us? In contrast, I'm talking about applying a theory, or at least a framework, to interface design, to make the experience of using the web—search engines, pages of content, databases, collaboration tools, whatever—more like the

evolution of a cognitive state, more like a convergent process of designing, where the user-as-designer is converging on goals as well as outcomes.)

So, how does conversation work? At an atomic level, how do we 'get the meaning'? Von Foerster begins our answer by saying, 'The hearer, not the speaker, determines the meaning of an utterance.' [Von Foerster 2006]

Humberto Maturana [Maturana 1983] posits that the utterances (words, sentences, paragraphs, pictures, motion graphics…) *cannot* 'contain the meaning', because the nervous system is a closed system that does not 'take input'. Rather, the words and sentences and images are *triggers* of prior, acquired knowledge that conjure the meaning—and the hearer does the meaning-making.

If this is so, how do new ideas ever become conveyed? Surely the only way is by *juxtaposition* of prior concepts that are combined in new ways with *specific and novel relations* that make new concepts. Pask has much to say about the *types* of relations that are required to most-reliably convey meaning in his construct of entailment meshes [Pask 1980A, Pask & Pangaro 1980, Pask & Scott 1973]. Some involve the '*why'* of the way the concepts interrelate, that is, they involve a *goal*. Such 'why' relations give an orientation to the hearer as to intention. Another type of relation involves *how* these concepts are to be combined *in action*, in the nitty-gitty of *achieving the goal*.

This requires an example to make clear, and here is the forever-repeated example of the circle-compass-plane:

> i. A **compass** can be used to inscribe the geometric form of a **circle** on a **plane**.
>
> ii. By holding the arm of the **compass** with the sticking point stationary in the **plane**, and then by using the end of the other arm of the compass to inscribe a mark in the plane as the entire compass is rotated around the stationary point, the resulting mark is a **circle**.

You should recognize these two types of relations—a *why* and a *how*—in these two statements. Pask's claim is that the neighborhood comprising circle-compass-plane is a *concept* that is *coherent*. This particular type of relation is therefore called a *coherence.*

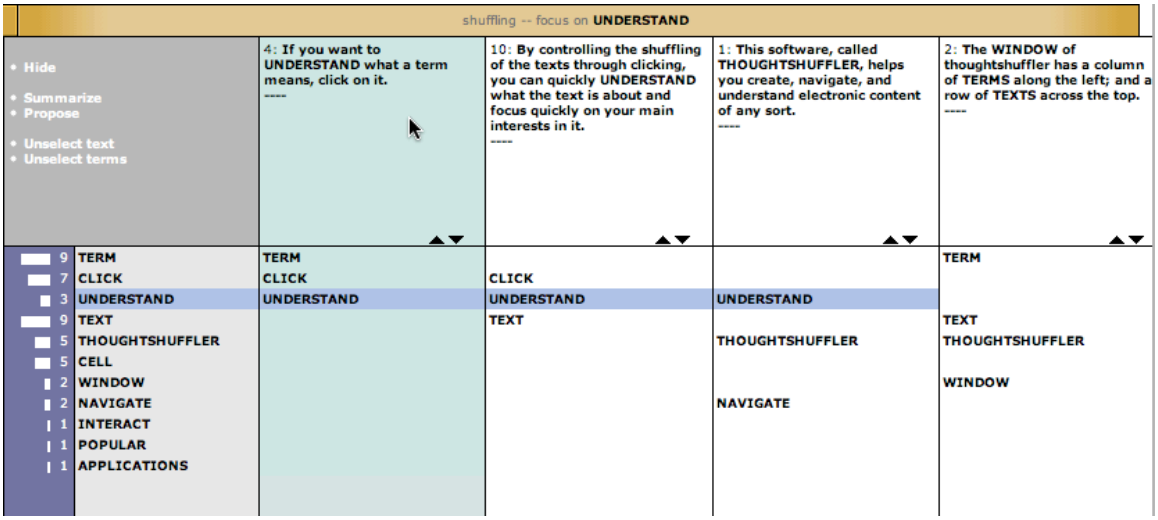
This is interpreted to mean that the two relations drawn by the statements above—one 'descriptive' and the other 'prescriptive'—are necessary and sufficient for a cognitive *understanding*. Because of the detailed interdependency of the two complementary relations—how they 'interlock' or 'make sense together'—it is more difficult (though not impossible) to mis-understand intention when both are available, either explicitly or implicitly. Thus, they can be used by the listener to construct a meaningful whole. Because of this, conversation—and cognition in general—has the degree of reliability that it does.

What does this have to do with thoughtshuffler and conversational search?

If the goal of interaction that begins with a search is to learn something new—especially when precursor to effective action—conversational search requires an interface tuned to learning. Such an interface will operate more effectively, I believe, if based on a theory of conversation.

Thoughtshuffler is that interface. It enables the easy creation, manipulation, and evolution of neighborhoods of concepts, so that the user can specify the focus and thoughtshuffler can display the content that conjures meaning in the user that is most closely associated with that focus.

Here is a snapshot of the thoughtshuffler interface as it stands today:



The left-most column contains a list of concepts, here called ‘terms’, that are pre-defined in this particular demonstration content (they can also be automatically generated in a limited sense, see next paragraph). The next column to the right, colored green, lists a neighborhood of concepts that comprise the meaning of the text above them. The remaining columns show content that the interface has ‘shuffled’ into the user’s view, all relating to the currently selected term, ‘understand’. The text snippets constitute ‘models’ (in Pask’s terminology) that explicate the relations inherent in the neighborhoods of the terms (a. k.a. ‘topics’ of entailment meshes).

Where do the entailments come from? In a ‘real’ cognitive system, such as the brain/nervous system, natural functioning creates the dynamic structures that we model as entailments. In digital, serial devices such as today’s computers, it’s problematic. Pask [see Pask 1980a] explains how it is impossible for a single, serial, digital engine to create the structures as brains do. While aware of this fundamental limitation, I’m developing an algorithm based on the contradiction-detection mechanism (Pask’s deliberately- and obscurely-named ‘Rule of Genoa’) that will do this automatically but approximately—that is, the structures proposed by the device may not accurately mirror the coherence of dynamic structures (ala how/why complementarities) inside a cognitive system such as a

brain. But I hold the conviction that the user will benefit from such triggers *because* this most closely mirrors users conjure meaning in the user.

But this feature of auto-generating entailments isn't even necessary, I believe, for the interface to provide significant benefit to the user. The ability of the user to select/de-select terms, review texts, compare and contrast them, all form a qualitatively different user experience where *the user controls the elements of making meaning*—the terms that trigger meaning, and the models that comprise neighborhoods of terms that 'hold' the meaning. New juxtapositions of terms lead to new understanding (learning).

Today's search engines are merely machines to turn users into keyword manipulators. Why not a power tool for 'keywordese'?

I believe that thoughtshuffler shows how conversation theory led to a new user interface paradigm. I don't know of any AI models that have led to a new interface paradigm, please let me know if you come across one. The range of applications for thoughtshuffler is wide, and I continue to look for ways of bringing the interface to the marketplace.

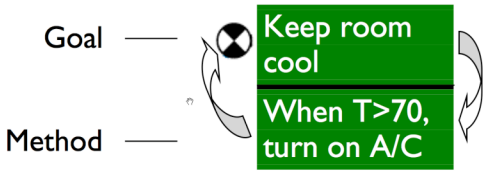
II. At a book reading last night I chatted with a self-described professor of artificial intelligence, and at one point I asked him, 'What's new in the field?' And he said, looking right at me, 'That depends on who's asking.' By way of what I thought might be encouragement to provide a serious response to a questioner worthy of his depth of experience to expend some bio-cost in answering, I offered that I studied computer science and then went to England to study cybernetics—happening to mention that it wasn't available for study at the undergraduate institution I attended. And he said, 'What, they didn't teach control theory?' I began to reply and then stopped. The hors d'oeuvres took all my attention.

Goal-Aware User Interfaces: Goaled Linking

I. Here's a useful definition of a 'goal':

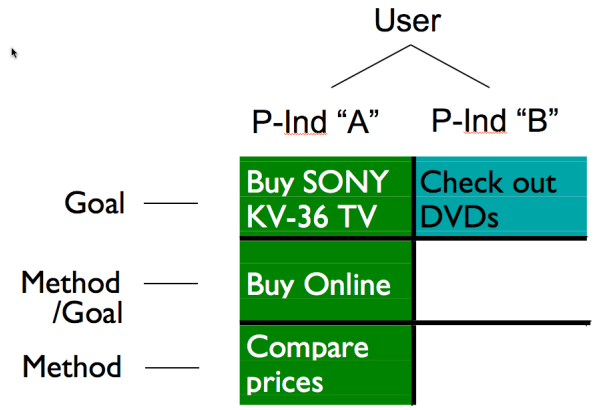
> An articulation of a desired end-state in the context of one (or more) means or methods to achieve that state—that is, sub-goals.

That's a dumbed-down paraphrase of Pask's structure for the architecture of conversations [Pask 1976, Pangaro 1989], which contains a clear, formal, complete model of goals. Here's another simplification in the form of a diagram:



Caption: The process in the upper part is the goal, while the process in the lower part is the method or action to achieve the goal. The down-arrow indicates invocation of the method, and the up-arrow indicates return of results that enters the comparator (shaded circle) in order to compare outcomes to the original goal.

This is a minimal view. In practice, humans have multiple, simultaneous goals. Again, Pask provides a useful model that he calls 'P-Individuals', or psychological-individuals, that distinguish multiple perspectives in the same human, or mechanical 'M-Individual'. A goal is one type of perspective and so Pask's model can be applied to modeling a 'User' as comprising multiple, simultaneous P-Individuals, each with a goal.



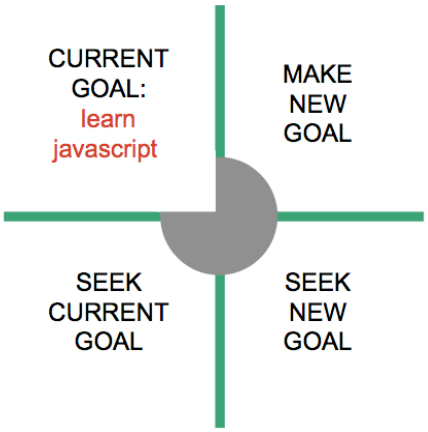
Caption: In the course of seeking to purchase a specific Sony television on the web, including comparing prices, a user is distracted into a different goal of looking at DVDs, perhaps as a precursor to purchasing one of those as well.

Who's goal is it, anyway? What I mean is, what goals are embedded in the interface? Always today, it's the goal of the software designer. How would an application of conversation theory work to improve the user experience, based on models of goals?

Humans have multiple, overlapping, and sometimes conflicting simultaneous goals, and our software should reflect that. Oh sure, you say, it does, because we can have multiple software applications open at once, each with multiple screens. Do you love having all those windows just lying around, littering your desktop, with most of them invisible behind other screens, and no structure to what's where or how to get back to it? I don't love it.

I'd like to be able to declare a simple goal and have the software help me track it.

Imagine I'm browsing and I find an 'interesting' link. Does clicking on that link mean that I'm continuing to pursue my current goal? Or, that I am distracted and want to pursue a different one, either temporarily or permanently (that is, I want to abandon my prior goal). Here's an interface idea for that, just a mock-up but something I think worth prototyping:



Caption: Imagine that you have moved your mouse over a link. Up pops the display: it indicates that your current active goal is 'learn javascript', something you declared earlier, probably by typing it or highlighting some text. You now have a choice to declare the relationship of this link to your goals:
- Pursue this link to accomplish current goal: by clicking on 'Seek Current Goal', you declare that you are still pursuing the same goal and maintaining focus, that is, to learn javascript
- Remember this link as new goal: by clicking on 'Make New Goal' you declare that you are now pursuing a new goal **but** that you don't want to change your focus yet
- Pursue link now as new goal: by clicking on 'Seek New Goal' you declare that you are pursuing a next, new goal AND you want to change your focus to that next goal.

Is this workable, practical? I can't be sure unless it's implemented and tested, but yes, I think it is. Is it exciting? Well, at least a little, because it would generate a structured goal tree that you could then navigate (I won't guess at the navigation part of the interface here, because I think it's straightforward and not, in itself, innovative). This would minimize or eliminate the need for controls such as Back, Forward, Next Window, History, Bookmarks—and, oh yes, would subsume all that excitement about Tabbed Browsing, which makes existing functions slightly simpler, instead of offering a major improvement to a fundamental problem.

II. I gave a presentation these ideas at local chapter of a national organization of computer-human interface professionals [Pangaro 2000a]. The title of my talk included the phrase 'Goal-Directed Software' and I was working away the night before, polishing my slides, when my FAX machine started chuntering away. It spat out a 'cease-and-desist' order from the attorney of a design firm that had trademarked the phrase 'goal-directed'. Outraged that someone could legally trademark a phase at the core of cybernetics, I felt bullied by this last-minute, impersonal intrusion. But then I globally-searched-and-replaced the tex of my presentation so that 'goal-directed' became 'goal-driven'. I like it better, actually, because if I'm 'directed', it sounds like I'm being shown the way by an external force, but 'driven' sounds more like I'm being driven by my goals. Actually, that's a complete rationalization. All I wanted was to have had a conversation about the issues with a person and not a FAX.

Goal-Aware User Interfaces: Runaway

I. Here's another example, I think a more powerful one, of a Goal-Aware Interface. Ever notice how often you perform the same actions at your machine? For example, every time I return to my computer after being away for more than 10 minutes, I wake it up from sleep, go to the email client, check for new mail, find my Inbox, make sure it's sorted newest-first, and then I read the latest emails. How many interface gestures is that? Too many, clearly—since it's something I do many times a day. Why isn't it one or two?

Similarly, when I read an email from a business colleague with whom I've been working for days and days, I open the email, open its attachment, edit the attachment, save it with a different filename, hit Reply to the email, add a few comments to the email, attach the updated file, and then hit Send. I'm tired out by all that, since so many of the interface actions are PURE REPITITION. We don't tolerate such repetition from our human relationships, no less our pets, so why should we allow our machines to force us to be so mechanical and repetitious? Because we don't think of humans as goal-directed—I mean goal-driven—and we don't have goal-aware interfaces.

Now again I can hear the computer scientist geeks out there, saying that all I'm talking about is already solved with keyboard macros and recording repetitive actions. Nonsense. Why do I have to become a programmer? Why can't I just have a relationship with the system that includes automatic learning? We've got Baysian learning in spam filters that is automatic, why haven't we applied it to repeated actions as well as repeated spam emails? Because we're not thinking of humans as fundamentally goal-driven, I believe. Because we don't see the huge value in bio-cost reduction for goal-aware interfaces.

I'd like to focus such automatic learning algorithms upon watching my actions and moving ahead without further effort on my part—to presume where I'm going, and to perform those actions for me. If my machine has been quiet, and then the mouse is moved, and then the email client is invoked… it's not rocket surgery to figure out what I'm going to do next! Let the machine presume I've got a goal, because I've had it so often before, and let the machine go ahead with it. Let's call the system 'Runaway' because I want it to run away with my intention.

Does that scare you? Yes, there's danger that it will do something you don't want this time. Like any interface design, the devil is in the details. Runaway needs a sophisticated 'UNDO' capability. And a clever way to let you stop, interfere, redirect its direction. Again, I consider those problems solvable with iterative prototyping; my point is as follows:

> What innovations in user experience can be led by serious application of conversation theory, its models and methods, such that computers are better collaborators?

There is ample precedent for this desire [cf. Negroponte 1975, whose focus is the role of computation in architectural design; see also Pangaro 2000b].

II. The desire for Runaway came not from a mild desire to avoid repeated functions, but from rabid anger at PC operating system crashes. Why should I have to reproduce the complex configuration of my system when I'm working on specific tasks—opening all those applications and documents, configuring the windows in the right place with the right viewing modes, all that silly repetition all caused by a fatal flaw in the technology. What improvement in the gross domestic product would occur if computers recovered from their own crashes to restore the human's environment? Could we use the human

resources thereby made available to eliminate world hunger? Maybe I was just frustrated that there was noone to yell at.

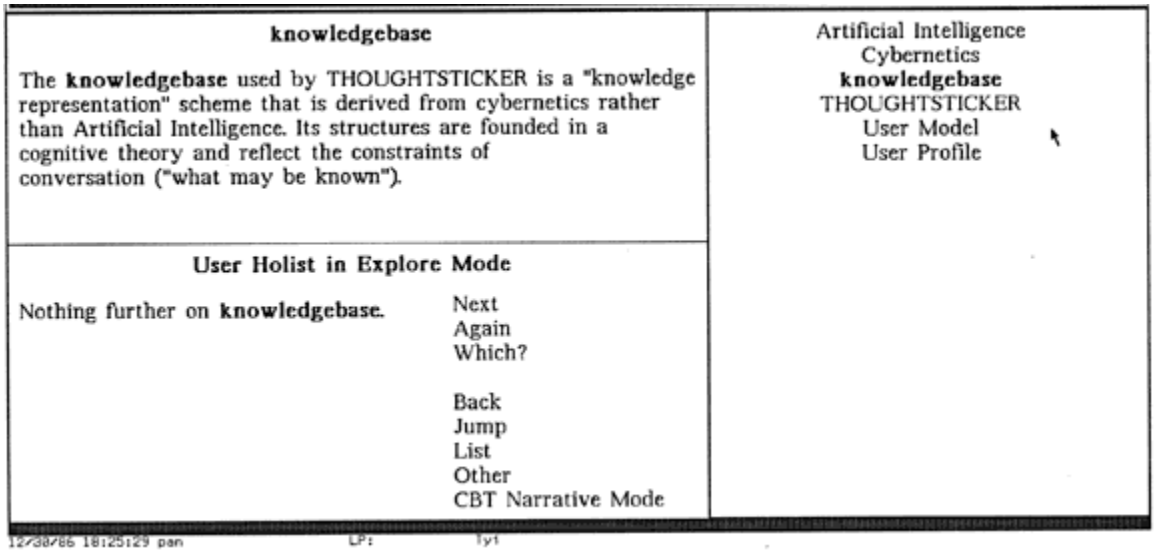
THOUGHTSTICKER and Personal Computers

I. A review of applications of conversation theory to software would be incomplete without THOUGHTSTICKER. I've written of implementation details from a 1980s version [Pangaro 2001] so here I will point up the connection to interaction design and other themes of this paper.

The goals of this implementation included the desire to update Pask's version of THOUGHTSTICKER from the 1970s, which was limited by technology and budget. For example, without enough digital storage to hold the text content for the training lessons, his version would light an LED above a cubbyhole from which the student would extract a clipboard with the context printed on paper. (I have often been astonished at the outcomes Pask achieved despite constraints; the lesson is that clarity of ideas can overcome limitations in technology, while limitations in ideas cannot be overcome by capabilities of technology.)

The best software development technology of the early 1980s was the "artificial intelligence workstation", that is, a hardware platform designed to run operating systems with huge libraries of sophisticated functions for rich graphics, user interaction, hypermedia, databases, knowledge systems, and so on. (No, these workstations were not artificially intelligent.) Funded by fans of Pask from England who were our clients, I was the first customer to put a down-payment on the Symbolics Model 3600. (What arrived was serial number 006 only because a secret government agency in Washington, DC, kept jumping into the queue ahead of us.) The development capabilities were breathtaking. We had the finest software platform in the world. We had no excuse to fail.

Our early attempts, however, were baroque in appearance and use. They were fashioned from our perspective as researchers in knowledge-based systems, rather than in training, which was our intended application. So I re-wrote the user interface from the perspective of what we called "a naïve user" and the next figure is a sample screen from that implementation, circa 1985:



Caption: THOUGHSTICKER software circa 1985, with an interface to deliver training, based on Pask's entailment meshes and theory of learning styles.

Here is a list of advantages of THOUGHTSTICKER over popular computer-based training systems of the same era [as described in Pangaro 2001]:

1. Based on a cognitive theory of human conversation developed over the period of 1955 to the present, and affirmed in empirical studies (that is, Pask's conversation theory).

2. Uses a robust knowledge representation scheme to provide a true knowledgebase; all conceptual dependencies are represented in a network structure with no fixed paths.

3. Sensitive to an individual's cognitive style, modifying responses accordingly.

4. Sensitive to individual variation in user's prior knowledge and can be tuned by a variety of user profiles (for example, naïve computer users, or experienced computer users of another particular vendors' hardware or software).

5. User is free to ask questions and explore throughout the knowledgebase at any time. The user helps direct the remedial dialogue, which is derived from a combination of user's focus, the structure of the knowledgebase, and the history of the interaction.

Imagine reading this list of benefits in 1985. The Apple Macintosh was only around about a year and not the mass-market phenomenon it today. Our demos had to begin with an explanation of a new-fangled input device: it sits on your desk and, as you move it around, a cursor moves on the screen. It's called a mouse…Needless to say, a live software demonstration took over an hour and often led to our audience's to cognitive overload.

Today, a review of the claims above is more likely to lead to a ho-hum reaction and the sense that many such systems have been successfully built, delivered via the Web—in short, that THOUGHTSTICKER is pretty much old hat. This is simply not true. Because the interactions were based on a fine-grained knowledge model of both the subject matter and the user's evolving cognitive state, the experience was more like a conversation than any other human-computer interaction to date. If these features were to be found in a web-based training system today, the world would beat a path to our door, because there could be nothing more personal than a software experience based on conversation theory.

II. Around the time the phrase 'personal computer' came into vogue—in my experience, around 1976—I was learning about conversation theory, having met Pask at the Architecture Machine Group under Nicholas Negroponte. Pask had been a consultant there for some years, and the Group's concept of "idiosyncratic systems"—a machine environment that understands the particulars of current users, and responds in kind—seems to have come directly from Pask. When I learned enough about Pask's entailment meshes that power THOUGHTSTICKER, and all the individualized interactions it created, it was immediately clear how to build such systems.

But, then, what is this artifact called 'personal computer'? Quite frankly, I was confused. Nicholas explained this as meaning, 'small enough to be owned by a person.' 'Ah!', I replied, 'OK, I guess, that meaning could fit.' But what about 'personal' in the sense of 'relating to a particular individual', that is, reacting to their beliefs, actions, values, goals? Clearly, in that sense, if you went to a computer store to ask for a personal computer and got an honest conversation, the salesperson would say, 'I'm so sorry, sir, we don't have any personal computers—but we have lots of *impersonal* ones.'

Issues, controversies, problems

I. The issues most forward in my mind are the lock-in paradigm that controls today's software development concepts. I'm using 'paradigm' in the strict sense, in Heinz von Foerster's sense that I will paraphrase as 'that unconscious framework of thinking that prevents you from seeing that you do not see.' Computers are great (when they calculate fast for us) and also horrible (when we have to tell them each detail of what we want to do, step by cruel step).

In this sense, the first 50 years of human-computer interaction has been a dark age indeed. Is that controversial, that statement? Who would argue that we have a vast distance to go before human-computer interaction is as seamless as human-to-human interaction? But wait, human-to-human interaction is full of seams, and misunderstandings, and problems. But I'd rather have those problems than the ones I have today with computers.

II. At dinner tonight with two new friends, both in their early 80s, I was attempting to describe cybernetics as distinct from artificial intelligence. 'Does this have anything to do with that Turing fellow?', they asked. 'Didn't he predict that computers would be as smart as people someday soon?' 'Well,' I explained, 'he never tried to predict the timing

of it, unlike some who were less wise, but he did create a test that might indicate that they *were* intelligent.' 'Isn't there a prize for that, they asked?' 'Well,' I said, 'yes, I believe that someone has set aside a million dollars for when a computer can pass the Turing test and can completely fool a human.' 'Well,' the wife immediately said, 'I've got one of those at home, can I collect the money now?'

References

Beer, S. (1966). Chapter 6: About Models. *Decision and Control: The Meaning of Operational Research and Management Cybernetics,* John Wiley & Sons.

Calvino, I. (1986). Mr Palomar: A collection of short stories about the experiences of Mr Palomar. USA: Harvest Books.

Dubberly, H., Maupin, C.J., and Pangaro, P. (2004). Paying Attention: Measuring the Biocost of human actions. Available at http://cyberneticlifestyles.com/biocost-treatise.html.

Dubberly, H. & Pangaro, P. (2006) Description and syllabus of Course CS377, Introduction to cybernetics and systems for design. Descrioption available at http://pangaro.com/syllabi/cs377.html#cs377cybdesign.

Dubberly, H. & Pangaro, P. Cybernetics and Service-Craft: Language for Behavior-Focused Design. Kybernetes edition on Cybernetics and Design, edited by Ranulph Glanville, in press.

Kallikourdis, D., & Scott, B. (1972). A fresh look at cognition and the individual. International. Journal of Man-Machine Studies, 4, 211-216.

Maturana, H. R. (1983). What is it to see? In Arch. Biol. Med. Exp. 16: pp. 255-269.

Negroponte, N., Editor (1975). Soft Architecture Machines. Boston: MIT Press.

Pangaro, P. (1989). Architecture of Conversations. Available at http://pangaro.com/L1L0/.

Pangaro, P. (2000a). Who's Goal Is It, Anyway? Abstract available at http://www.baychi.org/calendar/20000314/ and http://pangaro.com/abstracts/BayCHI2000-1.html.

Pangaro, P. (200b). Participative Systems. Available at http://pangaro.com/PS/index.html

Pangaro, P. (2001). THOUGHTSTICKER: An Idiosyncratic History of Conversation Theory in Software, and its Progenitor, Gordon Pask. Festschrift in celebration of Gordon Pask, Kybernetes Volume 30, Number 5/6, 2001. Available at http://pangaro.com/published/thstr-fest.html.

Pangaro, P. (2005). 'Thoughtshuffler' mock-ups, descriptions, screenshots, and limited demos. Available at http://pangaro.com/thshpix/.

Pangaro, P. (2006) Materials related to internet search. Available at http://pangaro.com/search.

Pask. G. (1975). Artificial Intelligence : A preface and a theory". In N. Negroponte (Ed.) Soft Architecture Machines. Boston: MIT Press.

Pask, G. (1980a) . An essay on the Kinetics of Language as illustrated by a Protologic Lp". Proceedings of 2nd Congress of the International Association for Semiotic Studies, Vienna, 2-6 1979. Workshop on Fuzzy Formal Semiotics and Cognitive Processes. Reprinted in Ars Semiotica, Amsterdam: John Benjamins, pp 93-127.

Pask, G. (1980b). The Limits off Togetherness. Proceedings, Invited Keynote address to IFIP, World Congress in Tokyo and Melbourne. Lavington, S, Editor. Amsterdam, New York, Oxford: North Holland Publishing Company, pp 999-1012.
Pask, G. & Pangaro, P. (1980). Entailment Meshes as Representations of Knowledge and Learning. Conference on Computers in Education, Cardiff, Wales '80.
Pask, G., & Scott, B. (1973). CASTE: a system for exhibiting learning strategies and regulating uncertainty. International Journal of Man-Machine Studies, 5, pp. 17-52.
Pask, G., Scott, B., & Kallikourdis, D. (1973). A theory of conversations and individuals (Exemplified by the Learning Process of CASTE). International Journal of Man-Machine Studies, 5 (4), 443-566.
Pask, G., Kallikourdis, D., & Scott, B. (1975): The representation of knowables. International Journal of Man-Machine Studies, 7 (1) p. 15-134.
Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a General Theory of Planning. Policy Sciences, Elsevier Scientific, Amsterdam. pp 155-169.
Von Foerster, H. (2006). *Understanding Understanding.* NY: Springer.

About the Author

Paul Pangaro grew up interested in machinery but not machines, that is, purpose not embodiment. His explorations in various media (choir singing, acting, programming, filmmaking, computer graphics, cabaret, teaching, personal avatars) explore the intersection of intention and materiality. Paul Pangaro was introduced to concepts of cybernetics while an undergraduate at MIT where he took courses from and worked on research projects with Jerome Y. Lettvin. On being hired by Nicolas Negroponte to join his Architecture Machine Group (predecessor to the MIT Media Lab), Pangaro was introduced to Gordon Pask by Negroponte. As so many have said, cybernetics simply gave form to an individual's prior intuitions; perhaps this is another way of saying that cyberneticians self-select cybernetics. Pangaro left the MIT Ph.D. program to write a dissertation with Pask at Brunel University in England (Ranulph Glanville was on his thesis committee). Pangaro met Stafford Beer and Humberto Maturana on a number of occasions (too few), had encounters with Herbert Brün (musical and otherwise), and was fortunate enough to live not far from Heinz and Mai von Foerster in their final years. It must also be said that understanding the concepts of cybernetics follows best from knowing the individuals who originated them. For all of these introductions, the author is grateful. Privileged to have extraordinary collaborators, he co-teaches cybernetics of design at Stanford University with Hugh Dubberly, who is responsible for Pangaro's interest in design, and with whom he is co-developing an approach to product design based on personal bio-cost. His work can be found at pangaro.com, cyberneticlifestyles.com, drwires.com, and cybernetic ians.com.

Additional Reading

Ashby, W. Ross. (1960). *Design for a brain.* London: Chapman and Hall.

Lettvin, J. Y. (1989). Introduction to volume 1. In W.S. McCulloch & R. McCulloch (Ed.), *Volume 1* (pp.7-20). Salinas, California: Intersystems Publications.

McCulloch, W. S., & Pitts, W. H. (1965). A logical calculus of the ideas immanent in nervous activity. *Embodiments of the mind* (pp.19-39). Cambridge, Massachusetts: The MIT Press.

Maturana, H. R., & Varela, F. (1980). Autopoiesis and cognition: The realization of the living (pp.2-62). Dordecht: Reidel. (Reprinted from *Biology of cognition,* by H.R. Maturana, 1970, Urbana, IL: University of Illinois.)

Maturana, H. R., & Varela, F. J. (1988). *The tree of knowledge.* Boston and London: New Science Library and Shambala Publications, Inc.

Reichardt, J. (1971). *Cybernetics, arts and ideas* (pp.76-99). London: Studio Vista. (Reprinted from A comment, a case history and a plan, *Cybernetic Serendipity,* by G. Pask, Ed., 1968, London.)

Pask, G. (1976). *Conversation theory.* New York: Elsevier Scientific.

Foerster, H. V. (1974). *Cybernetics of cybernetics.* Urbana, IL: University of Illinois.

Ernst, G. V. (1987). *The construction of knowledge, contributions to conceptuals semantics.* Seaside, CA: Intersytems Publications.

Norbert. W. (1948). *Cybernetics,m or control and communication in the animal and the machine.* Cambridge, MA: The Technology Press.

Winograd T., & Fernando,m F. (1986). *Understanding computers and cognition: A new foundation for design.* Norwood, NJ: Ablex Publishing Corporation.