GERSHOM'S LAW

Principles for the Design of Performance Support Systems
Intended for Use by Human Beings

by Gary J. Dickelman

This article originally appeared in CBT Solutions Magazine, September/October 1996.

About nine years ago I was reading an issue of *CBT Directions* when I stumbled on the words *performance support* - on the last page, in Gloria Gery's column. I was an authoring system developer and vendor at the time, so reading the words of CBT's erstwhile guru was part of doing business. In those few paragraphs (and hence) I learned that *performance support systems* enable people to *do it* because they provide "...access to integrated task structuring, data, knowledge and tools at the time of need." Such systems do not tax the user's (aka *performer*'s) memory, nor do they require "manipulation of too many variables." The primary focus of performance support systems, I would learn, was *task completion* with learning a mere *secondary* consequence.

Imagine my surprise. Performance support seemed a clear departure from Instructional Systems Development and ran contrary to all conventional CBT wisdom. Cool! I like change-for-the-better. I am also a defender of human attributes, which I had always felt were severely violated by the endless hours of CBT flooding the marketplace and by the machine-centered design of most computer-based business applications. Like the young daughter of a developer said after reviewing his best program, "CBT makes you *sleepy*!" - and like the payroll clerk said as her forehead was treated for palm slaps, "At least this time I don't need shards of computer monitor glass removed from my foot." At last there appeared to be a remedy in this notion of performance support - for the badly designed systems *and* for the responses to them in the form of poorly designed CBT - that would focus on the human being.

How and where do we start developing performance support systems? How do we analyze business problems and represent them for performance? How do we account for the human factors? Exactly what constitutes minimal information to solve a problem? What does it mean to have few variables to manipulate? I'll attempt to answer some of these questions, but first make a mental note: I am *not* going to discuss graphical interface design. That comes *after* creating representations of business problems. The focus is on where performance support analysis and design *begin*.

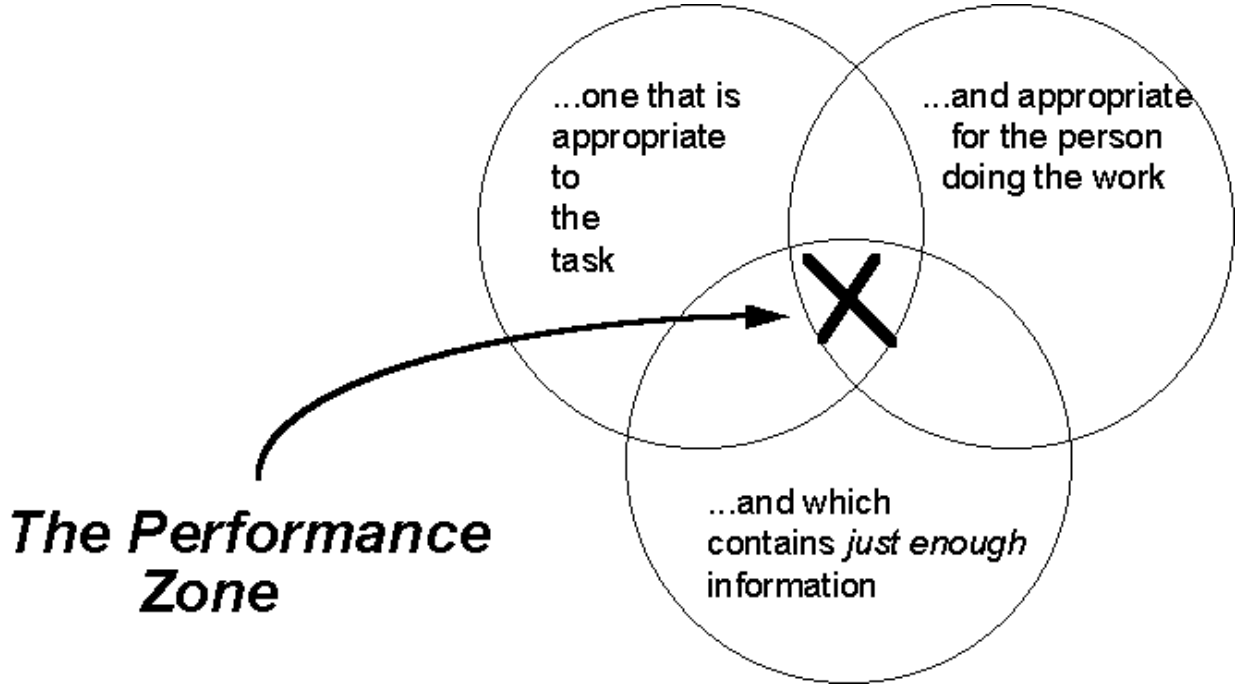
My quest for performance support design methods had me suffer through several hundred demonstrations of Microsoft Publisher, WillMaker, and Quicken. It seems that many people *know* performance support when they *see it*, some are pretty darned good at *describing* it, but very few can tell you how to go about *doing* it. So ...*big sigh*... over the next eight years I designed and developed 20 or so performance support "things" with varying degrees of success - and created a development methodology. Let me share some of the *how*s.

Performance support is nothing new. Several disciplines exist which together form what we recognize as performance support. For clarity let's distinguish between *performance centered design* - the activities and techniques of creation - and *performance support systems* - the results of good performance centered design. Gershom's Law says that performance centered design is found in:

- Total Quality Management
- Process Modeling and Simulation
- "Diversity" (that very '90s perspective)
- Games and Imagination
- Human Factors Engineering
- Usability Engineering
- Hypertext Engineering
- Decision Making For Leaders

Gershom's Law

Gershom's Law is about creating *representations* that are business-focused and human-centered. It is only after we master creating such representations that we can or should begin designing interfaces and writing code. The Performance Zone of business is the place where there is accurate representation of the business activity *and* appropriate representation for the people who have to do the work *and* just enough information. In the picture, X marks the The Performance Zone.



What about the laundry list of disciplines that comprise Gershom's Law? Clearly you won't learn them all in one short article. But in the spirit of performance support you will see their contributions to the *methodology*:

Gershom's Steps for Discovering the *Best* Representation of a Business Problem

1. Write a clear statement of the business problem.
2. Articulate a mission statement that addresses the problem.
3. Create a process model.
4. Create a diversity model.
5. Use your imagination: Draw on your experience with toys, games, gadgets, work, things - to identify something that remind you of the process and people who have to do it. *Brainstorm with the problem solvers!*
6. Consider the kind and presentation of information that is reachable and for which you "get it" quickly.
7. Take a shot at making a representation.
8. Try making another representation.
9. ...and another....
10. Test usability.
11. Decide which is the best representation based on the usability test results.

Steps 1 and 2 are the stuff of Total Quality Management. Process modeling and simulation techniques - for process improvement, redesign, and re-engineering - will help you to sketch the problem, which is step 3. Total Quality Management tools and techniques are integral to process modeling and simulation. The diversity model is a description of who has to do the work in terms of interest, personal style, learning style, skills, and values. There are formal methods of assessing diversity - like the Myers-Briggs Type Indicator (personal style) and the Holland Self-Directed Search (interest) - and informal techniques as well. But you must do step 4: If designs are to afford motivation, you must know that which motivates the performer; if you are to apply spacial metaphors then the performers had better prefer that which is spacial. Right?

Steps 1 - 4 set the context for creating representations: (1) The backdrop is a *business problem*; (2) focus is maintained with a clear mission statement around solving the problem; (3) the process model is a clear *picture* of the problem / solution workflow, capturing detail and dynamics; and (4) the diversity model is the *who*, which suggests how the problem might realistically be represented and solved. It takes common sense *and* discipline to create representations. The former because you need context, focus and a picture to know where you're at and where you're going; the latter because best practices exist, valid techniques are well-documented, and the scientific method is appropriate for our activities. Gershom says, "Don't do this by the seat of your pants!"

Steps 5 - 9 comprise all the fun. It is a fact that everything that works in systems and everything that will ever work in systems can be found within the patterns of our thoughts and lives. Deciding how best to allocate your retirement investment among three funds based on their current and projected performance is not unlike playing the game Towers of Hanoi (see Norman's Things That Make Us Smart, Addison-Wessley, 1993). Deciding how much flex benefit money to set aside for health-related payments and how much to give to charity in order to realize the lowest taxable income is nothing more that the game RummikubTM (by Pressman). Once you know what the problem "looks like" all you need is an open mind and a bit of imagination to find many representations.

A representation is an *abstraction* of a thing with respect to a *few* of its attributes. Consider the international

symbols:

For a variety of reasons we *get it* immediately. The handicap representation abstracts *person sitting* and *chair with big wheels*. Gender, age, color, and location are irrelevant. *Person wearing dress* and *person wearing pants* are the abstracted attributes and communicate *women* and *men*, respectively. Gender is important, even though the

abstraction does not necessarily discriminate (women can wear pants and men can wear dresses). To the person who needs a restroom and wishes to share one with people of like gender, the latter representations are completely usable.

A doughnut is a *surface* with *one hole* in it. So is a coffee cup (i.e., the handle is the hole). Imagine that your doughnut is made of rubber or clay, then mold the cup part from one edge, leaving the remainder of the doughnut for the handle. With respect to the attributes *surface* and *one hole* the doughnut and the coffee cup are equivalent.

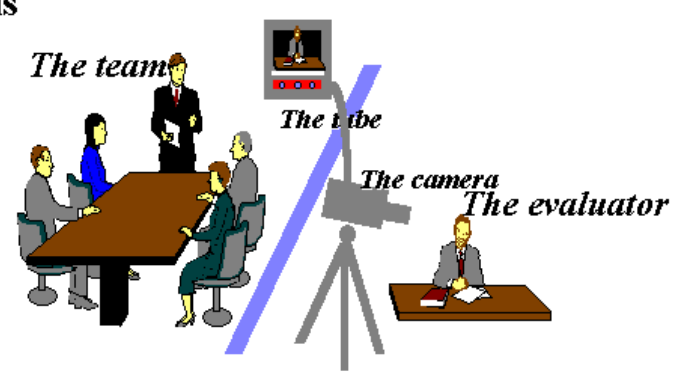
Well, that may be the case for the scientist - or topologist - but to the average coffee drinker they are quite different. A surface with a hole is necessary for grasping, but it's got to hold fluid - and it should *look* like a cup (affordance). These are the human factors.

There are many representations that abstract one set of attributes of a thing. The *best* representation addresses *a point of view*. That is why Gershom's List attacks the representation problem in steps. First find the attributes of interest by modeling the business context and create a number of representations. Next gather people who are going to solve the problem and test usability. Then select the representation that works.

The designer and the scientist do not have the proper perspective to choose the best representation. Only the performer does. So you must test usability by first determining what you *mean* by usable - what your goals are for ease of use, ease of learning, ease of navigation, affordance, requesting help, impression of simplicity and usefulness - then articulating *scenarios* that allow you to measure whether or not the goals are attained by a performer with one of the representations.

Gershom says, "Don't try this at home!" If you wish to measure something with accuracy you cannot stretch the ruler or stuff the ballot box. When designers sit alongside performers to test designs some amazing behaviors present themselves, including telepathic communication ("Move the damn mouse THERE!!"), limb manipulation ("Just put your hand on the mouse and let ME move it ...."), and "head-banging." It is much safer to apply an engineering approach to usability evaluation:

- **Determine Test Site**
- **Select Evaluators**
- **Document Usability Goals**
- **Develop Scenarios**
- **Develop Briefing**
- **Develop Debriefing**
- **Schedule Evaluators**
- **Set Up Lab**
- **Select Observation Team**
- **Conduct Dry Run**
- **Conduct Evaluation**
- **Debrief**
- **Report**



Gershom's Law

Some organizations have a usability lab with a padded control room and *every*thing. I used to think that the padding was sound-proofing, but now I know it is to protect the designers from self-inflicted injury. Gershom says, "Follow these simple steps with discipline, making objective observations and fair measurements. Then *accept the results* and follow through with changes to your representation - and test again."

What does Hypertext Engineering have to do with representations? In a word, *Reachability*. Talk show host Dennis Miller was the first to articulate why we need hypertext engineering:

Most people have the attention span of a ferret on two cappuccinos.

Systems that represent work require performers to *do* things and *get* things quickly. Customer service representatives must have answers immediately or no more than a couple of mouse clicks away. Everything must be *reachable* - which means the smallest units of information are no more than a layer or two below or above where the performer is looking. All units are equally accessible, immediately comprehensible, and make sense in and out of context. Tall order? Hypertext engineering holds the key. Any computer-human interface can be thought of as an arrangement of hyper*stuff*, where objects link to one another and have link anchors. Modal dialogues are popups, and a tool bar is a static pane, right? So the principles of hyper*text* engineering apply equally to the hyper*stuff* of interfaces - and to their prerequisite representations. Gershom says, "Everything has to be in my face in a flash or I'm outa here!"

The final discipline of Gershom's Law is Decision Making. Performance support system teams have to make many decisions as a group. Flipping a coin does not cut it. Random voting doesn't either. The Analytic Hierarchy Process, characterized by Thomas Saaty, is a discipline for making decisions based on paired comparisons of what is valued. It also measures consistency in the decision making process. Saaty's book Decision Making for Leaders (McGraw-Hill, 1982) tells us how, in plain English. Gershom says, "Performance centered systems designers are indeed leaders!"

So there you have Gershom's Law - which identifies the disciplines we must engage to responsibly create performance centered representations - and Gershom's List - which delineates a proven methodology for getting there. What about Graphical User Interface design (I hear you cry)? That's the next step - and another story. Finally, Gershom is a "stranger in a foreign land" because he crossed the line. He is (I am) now an IS Manager. My horizons have broadened beyond CBT to performance support out of necessity. Care to join me? That, too, is another story. Next time.