

In a Deloitte Consulting (now Braxton Consulting) newsletter, Gloria Gery was interviewed to get perspectives on performance support ten years after her book, *Electronic Performance Support Systems*, was published.

**One on One: Gloria Gery
Performance Support – Ten Years Later**

By Jon Warshawsky

Reprinted from *Cappuccino*, an internal news magazine for Deloitte Consulting (now Braxton Consulting). 1st quarter 2001.

Gloria Gery's *Electronic Performance Support Systems*, published in 1991, started a revolution. While computer-based training (CBT) dominated the online learning market, Gery advocated point-of-need information as a superior alternative to traditional training models that assumed point-in-time classrooms were the answer.

Gery's 12 years' experience at Aetna Life & Casualty as director of Information Systems Education included responsibility for both technical and end-user training. She founded her Tolland, Massachusetts-based consulting firm, Gery Associates, in 1981 and has focused on developing training and education strategies and programs associated with implementing computer technologies. In addition, Gery works with Daryl Conner's ODR (Organizational Development Resources), Inc. and teaches and consults around their Managing Organizational Change program. A keynote speaker at the Online Learning conference, she is the 1998 inductee into the HRD Hall of Fame sponsored by Training Magazine.

Recently, **Cappuccino** caught up with Gery to reflect on performance support 10 years after the publication of her landmark work.

Cappuccino: Your initial book on EPSS appeared 10 years ago, and a lot has changed in the last decade. What's different now from a decade ago?

Gloria Gery: Some things are the same; some things are different. Some things are the same but more intense. The book was really a rallying cry to question training as a default for performance development. The first chapter was called "The Great Cover-Up," which talked about the fact that we've hidden a lot of the costs of performance development and presumed that training has worked because we didn't have an alternative ... those issues still exist, only the degree is more intense.

C: In what sense?

GG: The issues are more prevalent and stronger, in the sense that they're bigger problems ... there's a growing mismatch between people and the work. We didn't have the kind of low unemployment we have in America today, so instead of having SOME people mismatched we have MANY people mismatched because we are desperate for employees. And whether or not the

training ever COULD have met the needs of these people, many of them lack the capacity to handle a number of variables and do it in the timeframe.

One of the other issues 10 years ago was that it took too long to develop content. And the expectation curve has grown, so people now want immediate best practice performance - time has become a commodity. People will pay almost anything for speed or more time. They often compromise learning for the appearance or reality of increased speed.

C: So even if we can develop content faster we're still losing ground.

GG: If you combine less capable people with a shorter timeframe, you're pushing on a rope. Less capable people require greater time (to train), and frankly I don't think the instructional model has changed at all.

C: And the instructional model is one-time event-based training?

GG: Event-based training, where you teach subjects and leave integration of those subject or processes to the learner. So, we teach people stuff out of context. We break it down and decompose it into content, and that content may be how to use the software, or a body of knowledge about a business process, or it may be conceptual knowledge. Then we leave the weaving to them, for them to - given a set of circumstances - integrate.

C: We're still putting content into boxes.

GG: So I don't think the training model has changed. People have become less capable or are in jobs for shorter periods because of the churn and constant reorganizing of the capability of mixing and matching things in a way so that now almost everything is configurable. Nothing is stable, nothing is predictable and nothing is simple. So we have all of those factors that existed 10 years ago now multiplied, sometimes they're cubed.

This context makes performance development a lot more difficult, but to me it's a difference in degree and not a difference in kind. All of those context issues have just become bigger and more urgent and critical. Organizational flexibility, the ability to turn on a dime and get a whole group of people to change quickly, cannot be achieved with individual interventions such as instruction ... today our strategy is this, tomorrow it's that, today our products are these, tomorrow they're those, today our marketplace is this, tomorrow we're going to add that - and we need to get everybody adjusted quickly. But when you look at training, even if you have outstanding courseware that works, you can't get it through the system fast enough for people to shift.

I was just in New Zealand on one of the 22-meter America's Cup yachts - one of these tourist things where you pay to be part of the crew - and one of the things about these boats is that they're huge and nimble - that's what a lot of organizations need today. That means they have to be built for flexibility, and the old performance development model is more of a battleship model. It takes a long time to turn it, and once it gets in that direction it's going to stay in that direction.

C: In other words, we're going to ramp up a big training campaign.

GG: And it takes forever. And the degree to which people can absorb it and apply it is questionable to me. Another thing that has happened is that 10 years ago people were talking about quality, now they're living it. Ten years ago people were going to quality meetings and saying, "what is quality?" Today people have institutionalized expectations for six sigma, or for ISO certification, so we now have less tolerance among our customers ... for variability in performance. And the consequences are greater, because the supply chain is shorter, people have

just-in-time stuff, so if you get a bad lot coming in - there's a long tail on the consequences. Whereas 10 years ago, you know, so our inventory was a little lower.

C: We had more inventory, so we could cover up these problems. What about the software then versus now?

GG: One of the things that hasn't changed - and it's pathetic that it hasn't changed - is that we've invested huge amounts in new software, whether it was driven by Y2K or conversion from legacy systems, and if you look at these systems, whether it's SAP or PeopleSoft, they all look like the old (IBM) 3270 (mainframe terminal) systems. The only difference is that you have data selection control. But if you look at the design, they're still data-centric and they still require the performer to understand the task model, and understand the relationship between doing the task and using the software. Those big systems drive a third of the world's consulting in training. You see people making more money on the training than they do on the software.

C: Well, let's keep that quiet.

GG: There's so much vested interest in the status quo.

C: Sure, well compare these systems to an ATM, which is very process-based and requires basically no training. You either want to withdraw money, or deposit or transfer money.

GG: Or take a loan. So what has happened ... is that software has not institutionalized best practices around a task. We're still in the position of having to develop task models and then help people map these task models to data-centric tools. The only place I would say there is an exception to that is in some consumer-oriented Web-based applications. The reason is, as you shift work to customers, whether they're corporate customers or individuals, you can no longer assume a body of knowledge among the performer population. So you have to create a much more explicit coach to performing tasks.

C: In other words, you just have to design it well.

GG: Yes. There's no tolerance. People will leave or abandon it in all kinds of ways - people abandon shopping carts in buying systems, or shift their accounts. There's very little difficulty in changing from Schwab to Fidelity, or whatever, if the effort required is greater than the time available at the moment of need to do a task. Or they will not only abandon the task or the vendor, but they will provide very negative feedback in a short cycle.

C: So this is the crucible we should be using to design everything in.

GG: That's right. We are now in a bifurcated environment - organizational systems as compared to Web-based, consumer-focused systems - and there are some very good systems. The Web is a functionally less rich space to work in - you can't do as much logic, you can't control the display as much because users can change it, it's not state-sensitive because there's no local memory, so when you select a variable the display can't change until someone hits 'Go' or 'Enter'. So it's a much less rich design environment, but the drivers are making better design in a more restricted space than we're seeing in huge powerful machines because there is just the opposite set of conditions. We presume that people who will do the work know the software. We presume that training is available and that people will go through it. We presume a help desk.

C: Your focus is on design aspects - instead of training everyone to death and creating elaborate performance support.

GG: I have a general statement that training and documentation, whether it's documentation on paper, or online - anything that's external to the interface - are compensatory for bad design. As you move up the expense curve - documentation is cheaper than training, training is cheaper than a help desk - those are the balloon payment for failure to think about building business performance into software. When people ask me what I do now, and this has turned into my elevator answer to them, it's that I do software design, integrating things that are now all over the place. The five variables that I help people integrate are task support or task processing, knowledge, data, rules and communications.

The second thing I do ... is to improve the representation of these things, so they're more explicit, so people don't have to know as much technical jargon and so that there isn't as much of what (human interface design author) Don Norman calls 'cognitive load.' He says that many of our tasks are constructed with a high cognitive load and that there's this huge amount of transformation people have to do between looking for data and how they can use it. The shorthand description that I use is to improve representation. That includes things like use of metaphor, language simplification ... embodying knowledge in the field label.

C: The interface saves the user from a lot of extracurricular analysis, then, and it preserves the original data through integration.

GG: Integration, representation, and I'm going to add a third thing, which is filtration. We are filtering out irrelevant things and bringing context to relevant things, whether it's knowledge, data, tools, whatever. So when people talk about just-in-time, you have to add just-the-right, and only the right stuff. Don't just bring me to a context-sensitive chapter, but bring me to filtered chunks of relevant stuff, whether it's data, tools or knowledge.

I'm working with a government agency and we're working on a form of integrated wizard to answer (citizens') questions. When people call in, there are 55 major issues they can call about ... the (agency) manual is two and a half feet tall. What we are doing is working people through the best way to handle that kind of call, and then as the call progresses the paragraphs, sentences, sections actually, depending what you want to call them - the smallest relevant chunk of that body of law is made available through a link ... there's also a link to the specific display in the agency's computer software, so that if you need to get a piece of data, or enter a piece of data, the link to the actual display is there so you don't have to learn how to navigate. So we're integrating tools, knowledge and task support. Because most of these people are temporary or seasonal workers, they don't have time to be trained. This is the most important thing - we are institutionalizing best practices.

C: From a consulting perspective, there are a lot of systems that we can't go into to change the design. For performance support in these cases, what are your recommendations?

GG: Build something in front of it. You have to. I'm working with clients now who are using a piece of software that I find stunning ... the product is called "Epiplex" (www.epiance.com). What it allows you to do is essentially to build an alternate interface that embodies tasks and can go out and interact with a number of underlying systems, because there are many people that have to work with six or seven systems. This system allows you to create wizards without touching the underlying code. These can have whatever you want in terms of language and sequence. They use kinds of

dialogues that progress; it can accept data and then based on your answers to these questions it goes out and gathers more data and presents it, or actually executes in these environments, such as SAP or PeopleSoft. If you want to change your marital status, say, in PeopleSoft, you may have to make up to 42 different screen interactions. What this does is simply ask you what your new information is and automatically executes the whole thing.

C: How hard is this to implement?

GG: In order to do this authoring of this task model, and linking it to knowledge, you need a couple of things. We need to make the task model far more explicit than we ever needed to do when we were dealing with instruction. In many of these cases, the decisions at hand are based on something like a data value. So, if this is the condition, then that, so when you call up, say, 'how much money can I borrow against my 401K?' there are five or six pieces of data that will drive the answer to that question. Your age, how much money you had in that account...

C: It's performance and decision support.

GG: Yes. So in order to work through the process and answer that question you need to have a very thorough view of the variables that affect progression through the task. There's another tool that I've come across that's an amazing tool, and you know, we spend a lot of time trying to diagram in Visio, and it's a terrible, terrible thing to have to do. And once you get it diagrammed you're unwilling to change it because it's so much effort, so we constantly have incomplete or erroneous info. There is a product called Procarta (www.procarta.com), which is a tool for helping anybody involved in modeling tasks for any person ... It's right up there with Epiplex, to me, as a parallel tool that changes how we do things. It's an environment that allows you to fill up a hierarchy with processes, activities and tasks, in that order, and then through a series of forms that can be customized ... you answer a whole set of questions around things like who does this, when do they do it, how often, what's the batch size, what tools do they use ... and when you're filling out these forms, behind it there's a database and it's establishing all the relationships between people, contacts, systems, tools. It can go down to the specific screen you would need to do something, and it works you through these things. The hardest part is figuring out what's a process, what's an activity, what's a task and coming up with the right label. It then has 45 outputs, including automatic creation of a Visio diagram. So when you make a change, you go back and recompile the Visio diagram.

One of the things Procarta does is develop an HTML web site that's already active, so if you've mapped things like documents, or deliverables, or forms to an activity, the web site that is developed is a portal to the task.