



# Ten Years Later:

## A New Introduction to Attributes & Behaviors and the State of Performance-Centered Systems

by Gloria Gery

**W**orking with technology can be a dream or a nightmare. Lots of my early work was in figuring out how to get people who were able to do work using software through training. But as early as 1976 when I saw my first online system (now a “legacy” system), I knew that software was designed around machine capabilities and limits and data—not human performance. That began my quest to decrease the complexity (and thus the learning required) to work with technology to perform tasks. I have long believed that most training, documentation, help systems, and human support are compensatory for poor interface design, unclear terminology, confusing system dialogs and messages, and software that does not map to user mental models. So rather than continue to refine the compensatory tactics like instruction, I have worked to develop explicit design criteria to design against. In the beginning, the work had less impact than I had hoped or expected. And so almost 30 years after the first users touched actual software applications, we are still spending incredible amounts of time and money to make people somewhat successful in using software—and we are still designing software that is focused on data transformation and retrieval and building more and more features and functions that people can’t use.

In the late 1980s I began to analyze what more performance-centered systems might be like: that is, what characteristics and behavior enabled anyone using that

software regardless of their domain expertise or software skills. I wanted to go beyond the clichés *user friendly* or *self-evident* or *easy to use*. My original 1995 article, “Attributes and Behavior of Performance Centered System” articulated approximately 20 of these characteristics, gleaned from my observations and experience in compensating for poorly design software (Gery, 1995). This update briefly summarizes my current view about what has happened... and what successful systems still stand. I continue to be proud of it. And when you compare those attributes with the tasks that people can now do with software, you might agree that they have held up.

### **The Current State of Design Affairs: Performance-Centered Systems**

Improved design of software *for use* or work performance is occurring (Constantine & Lockwood, 1999). We are finally seeing some software that is truly performance centered. We are achieving both performance and learning in the same context (Gery, 2002). Some of the best examples are in consumer software applications like Intuit’s Turbo Tax or small business applications such as Performance Now and Descriptions Now by Knowledge Point; these are incredibly powerful tools that can be used by almost anyone to do complex tasks such as filing sophisticated tax returns or creating job descriptions that comply with legal requirements and automatically generate consistent and clear employee performance evaluations.

Daily we see more and more self-service applications surface on the web that are designed to enable customers to do work that was previously done by deep experts. These tools not only structure work processes and embody technical knowledge in very simple ways, but they integrate vast amounts of data from corporate or industry repositories. They also produce clear, usable deliverables such as documents and graphs. Listed below are things easily done today by normal people—untrained people, people who lack confidence, people who have never used computers. These things were impossible 10 years ago—and in some cases even last year. The ability to do these tasks easily, consistently, and well is a function of improved design and a change in assumptions about the performer population and embedding knowledge into the applications. Let's reflect a minute on what's possible today...

- **Financial Planning and Financial Resources Management:** Analyze their financial needs, develop a plan, and select appropriate investments. Conduct transactions associated with buying and selling investments. Analyze the results of their decisions. Produce related tax documents. Apply for loans. Compare alternatives. Determine how much money to withhold for taxes.
- **Travel Planning:** Plan complex travel. Compare and contrast alternatives by various criteria including price, length of time, schedule. Purchase, exchange and return tickets, or make reservations using various payment methods, awards, and coupons. Check award levels. Select or change seats or room types. Communicate special needs. Donate miles. Print precise directions in five different "views." Book all related resources, facilities, and events. Locate and rent a castle in Ireland or France or Italy. Print tickets. Check the real-time status of flights, trains, or buses. Get updates on flight or schedule changes, weather, etc. on various wireless or Internet devices. Get destination or facility reviews. Contribute comments. Communicate with fellow travelers. Obtain necessary visas and travel documents.
- **Shop:** Locate, view, compare and contrast items, and purchase them. Read product evaluations and reviews. Contribute reviews. Return products. Send wrapped gifts. Create "wish lists" so others can buy what you want. Purchase in advance and set desired delivery dates. Send greetings. Track item status.
- **Manage Your Life:** Get a driver's license or special license plates. Pay taxes. Register and vote (in some states). Obtain building permits. Change your personal information. Submit and track health care claims. Choose a medical provider. Get information on good and bad providers. Change your health care or benefit options. Order prescriptions. Ship packages. Obtain appropriate postage. Research educational alternatives and apply to schools and for financial aid. Apply for students loans. Check status of your applications. Plan a wedding. Create a gift registry. Design a deck or garden

or house or kitchen or window treatments. Order the parts and materials. Select a telecommunications plan. Consolidate your bills. Get your birth or marriage certificates. File your taxes. File patents and trademarks. Donate your organs.

- **Small Business:** Obtain loans. File wage and tax reports. Match payroll records against government records. Compares employees' wage data submitted to IRS against employees' wage data submitted to Social Security Administration. Obtain an Employer Identification Number.

Unfortunately, we don't see the same task flow or work process-oriented systems that use explicit language and make consequences clear in enterprise systems or work processing systems designed for corporate employees (except possibly in human resources self-service environments) (Gery, 1997). Those systems have improved incrementally—and when we ask why they are not even better, we are typically reminded that users will have had training, that only experts will be using the software, and that we should remember "how bad they used to be"! How said that such progress has been made in one context, but not in another! There are many barriers to change...but they are falling one by one.

## Why Did This Happen So Quickly?

This "rapid" progress has occurred for several reasons that are important to understand:

- Business strategy depended on customer ability and willingness to use these applications—and business executives knew that customers were "discretionary" users (that is, they are only there because they *want* to be would walk if software was difficult to use, confusing, or frustrating).
- No one could assume that "training" would be required...or that if it were, customers would take it.
- If large volumes of customers had to use the sites and they had questions/problems that generated calls to humans at help desks, the cost and staffing requirements would overwhelm the organization.

So... once the drivers were different and the assumptions changed, business executives demanded performance-centered systems that could be used immediately. Funny, isn't it, how expectations drive behavior? Plus the information technology people weren't in control; business managers were. The focus was on completing processes, not exclusively on the *data* processing. The result: dramatically different software: performance-centered software that—

- Can be used by people who lack domain expertise.
- Requires no training and can be used by almost anyone with limited computer navigation skills (that is, anyone who knows how to click on buttons and boxes and circles) and no domain expertise.

- Results in immediate *best practice* performance that embodies good process and follows the rules associated with completing that process. Enables people to learn *as a result of doing*, rather than requiring them to learn *in advance of doing*. (for example, the system dialogs incorporate explanations and are viewed as “instructional contexts,” not just error messages.
- Dramatically simplify work because they automate tedious and complex activities.
- Hides complexity around navigation, data access through front-ends or presentation layers that orchestrate activity and data import/export/entry among and between underlying applications.
- Filters data, activities and tasks and knowledge based on individual and organizational roles, location, security or authorization level, etc.

## How Does This Relate to the Attributes?

By conscious design...and repetitive testing with inexperienced performers, these new software interfaces embody the attributes and behaviors of performance-centered systems articulated years ago...and they do more. These systems also filter data, processes, and knowledge based on user profiles or user actions and simply represent complex concepts, data, and other resources

### Performance-Centered Systems: Examples by Industry

#### Investments and Financial Planning

Fidelity Investments: <http://www.fidelity.com>

Charles Schwab: <http://www.schwab.com>

#### Real Estate and Home-Related Tasks and Services

Home Store: <http://www.homestore.com/>

Quicken Loans and other tools/calculators  
<https://quickenloans.quicken.com>

Do it Yourself (for example, design a deck or shed, estimate construction costs, etc.)  
<http://www.diyonline.com>

#### Hospitality and Travel Industry

Expedia: [www.expedia.com](http://www.expedia.com)

United Airlines: [www.united.com](http://www.united.com)

#### Government Services

First Gov: <http://www.firstgov.gov/>

Internal Revenue Service: [www.irs.gov](http://www.irs.gov)

Massachusetts Registry of Motor Vehicles:  
<http://www.state.ma.us/rmv/>

In fact, the cumulative effect of these attributes is software that is a joy to use, software that empowers and enables disintermediates and is effective and efficient (Gery, 1996). Software that results in *performance*. What more could we ask?

## References

Constantine, L.L., & Lockwood, L.A.D. (1999). *Software for use: A practical guide to the essential models and methods of usage-centered design*. Reading, MA: Addison-Wesley.

Gery, G. (1995). Attributes and behaviors of performance-centered systems. *Performance Improvement Quarterly*, 8(2), 47-93.

Gery, G. (1996). A chart describing the attributes and behaviors of performance centered systems and the range of implementation [On-line]. Available: <http://gloriagery.com/articles/0002.asp>.

Gery, G. (1997). *A comparison of large scale systems and consumer software development* [On-line]. Available: <http://gloriagery.com/articles/0001.asp>.

Gery, G. (2002). Achieving performance and learning through performance-centered systems. *Advances in Developing Human Resources*, 4(4), 474-478.

Wisener, G. (XXXX). Overcoming barriers to performance-centered design. *BSU IPT 563: Job Performance Aids and EPSS* [On-line]. Available: [http://gloriagery.com/articles/barriers\\_to\\_pcd.pdf](http://gloriagery.com/articles/barriers_to_pcd.pdf).

**Gloria Gery** is an independent consultant based in Tolland, Massachusetts. She specializes in performance-centered software design and in developing and implementing interactive training and performance support systems. Her emphasis is in strategy development and in creating the necessary managerial approaches. In addition, she assists in developing creative interface and program structures to ensure learning and guarantee performance. Gloria developed the performance support concept and works with numerous sophisticated clients in developing performance-centered software applications. Her design work, writing, and speaking qualify Gloria to develop appropriate performance support system strategies, architecture, and approaches with clients.

Gloria also has extensive experience in developing training and education strategies and programs associated with implementing computer technologies. Her experience at Aetna Life & Casualty as Director of Information Systems Education included responsibility for both technical and end-user training. Gloria's knowledge and experience in managing complex change combines with her understanding of computer technology to assist in critical systems implementation planning. She is the 1998 inductee into the HRD Hall of Fame sponsored by *Training Magazine*. Gloria can be reached at [ggery@attglobal.net](mailto:ggery@attglobal.net) and via [www.gloriagery.com](http://www.gloriagery.com).