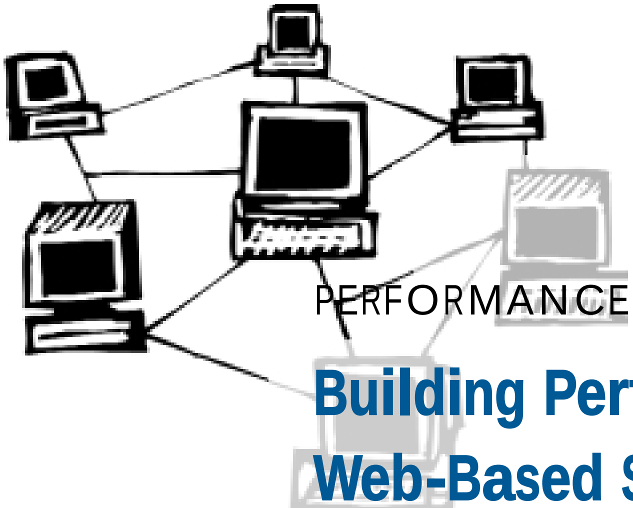

PERFORMANCE SUPPORT ENGINEERING

Building Performance-Centered Web-Based Systems, Information Systems, And Knowledge Management Systems In the 21st Century

by Barry Raybould

With the meteoric rise of the Internet and e-business, web-based systems (both the Internet and intranets) are becoming a major focus of software engineers and human performance technologists. Typical compensatory mechanisms for poor system design such as training and human support systems are becoming unacceptable from a business perspective. Competitive pressures to provide superior customer service and better products mean that workers must achieve competency in much shorter time frames. When software is used directly by consumers via the web, these compensatory mechanisms are no longer an option. While it is possible to compensate for the poor design of a customer service application by providing training and/or levels of more experienced human support, neither option is available to the end user on the web. Therefore, the ability to design software systems from a performance-centered viewpoint is a critical success factor in e-business.

The same principles hold for knowledge management systems: Just making knowledge available electronically is not sufficient. Only by having a performance-centered interface built on to the knowledge base is the knowledge rendered useful to achieving business goals. The major question facing organizations today is not whether to do performance-centered design, which was adequately addressed in the 1990s (Gery, 1991; Winslow & Bramer, 1994), but how to get it done.

The body of experience in developing performance-centered systems has grown significantly in the past five years, and practitioners have made considerable progress in elaborating the methodology. This article summarizes the convergence of thinking among various professional disciplines in analysis and design methodologies. It also describes seven key elements of the now-emerged performance support engineering development methodology. This or similar methods will be the foundation for designing performance-centered systems at the beginning of the 21st century, including consumer web applications, intranets, knowledge management systems, business information systems or any other systems designed to support work.

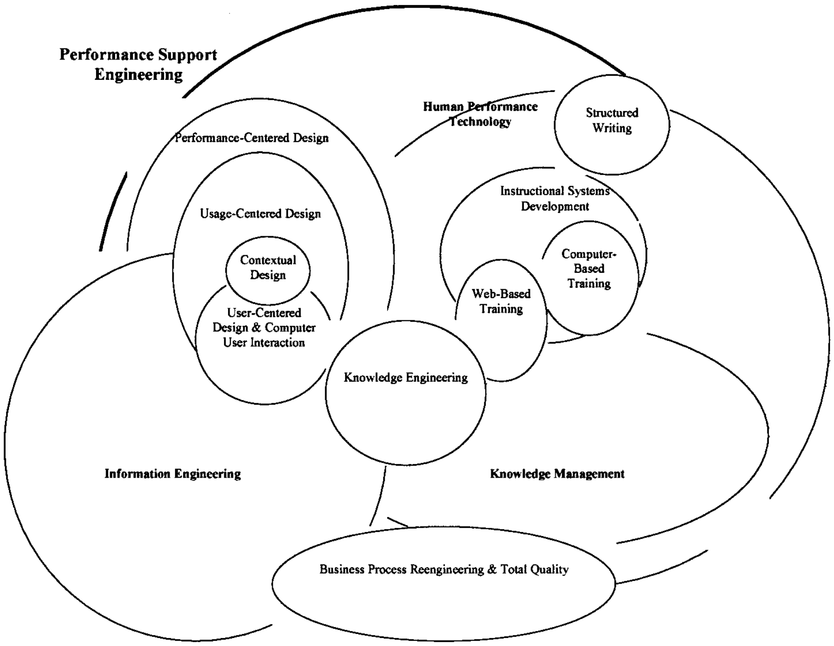

Figure 1. Relationship Between Performance Support Engineering and Other Professional Disciplines.

Hybrid Process

Over the past 10 years performance support development has progressed from something of an art to a much more structured and documented methodology (Raybould, 1995; Des Jardins & Davis, 1996; Marion, 1997). In parallel, the Human Computer Interaction community has been moving its methods closer to performance support concepts and away from user-centered design toward usage-centered design (Constantine, 1995) and contextual design (Beyer & Holtzblatt, 1998). Figure 1 shows the relationship between performance support engineering and various other professional disciplines.

Key characteristics of the emerged methodologies reveal the following patterns:

All processes start with knowledge acquisition, talking directly to job performers and subject matter experts about the work and identifying goals and barriers to performance. The design process is therefore data-driven according to the work and the performers rather than suppositions by the design team. The rule of three actuals applies: Observe actual work (not simulated work); observe actual job performers (not ex-job performers); and observe the actual work place (not an interview room). Observing this rule ensures that barriers to performance are exposed; it also prevents designers from making erroneous assumptions about the nature of the work that might lead to inadequate design.

The processes center more on deliverables rather than on a pre-established sequence of activities (see Figure 2). They revolve around a set of models, maps, and representations of the work and the design that are continually refined in an iterative process as the project team learns more about the work and the emerging design.

An example is the Performance Support Mapping® methodology (Raybould, 2000) shown in Figure 3. It comprises elements of information engineering, business process re-engineering, instructional systems design and computer-based training, human performance technology, interface design (usage-centered design and contextual design), knowledge engineering, and structured documentation.

The key phases in this process revolve around a set of deliverables that evolve raw data from job performers and subject matter experts into representations of the work and of the emerging design. There are four phases of activities.

Phase One: Look and Listen

- Observe current working environments and gather data from job performers and their managers to hear their view of the work, its goals, and barriers.
- Talk to the management of the organization to understand those goals and what is driving the business.
- Conduct surveys to gain more statistically significant data on important aspects of the work.
- Conduct focus groups to explore particularly important aspects of the work or barriers to performance.

Techniques for performing the above activities are derived from various sources, such as the front-end analysis

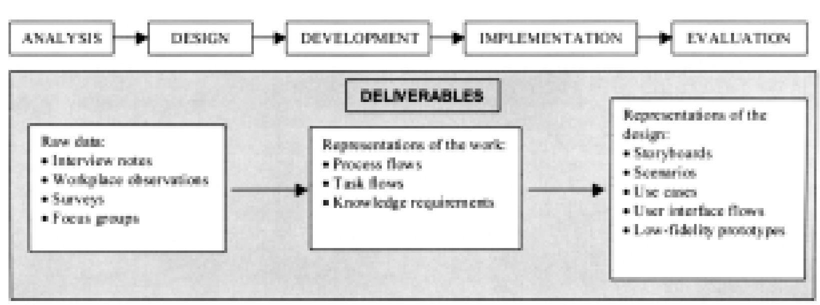

Figure 2. Generalized Methodology.

processes for performance technology projects (Robinson & Robinson, 1995; Swanson 1994), knowledge acquisition techniques from the expert systems fields (McGraw & Harbison-Briggs, 1989; Kelly, 1991; Tansley & Hayball, 1993; Dutta, 1993), techniques from the knowledge management field (Wiig, 1995), and various statistical and survey techniques for performance support projects (Raybould, 2000).

Phase Two: Understand the Work

- Develop detailed understanding of the work as it exists today.
- Create models and maps that represent the work at the individual, organizational, and process levels.
- Identify key barriers and roadblocks to peak performance.
- Align understanding of key business goals.
- Identify factors that differentiate high and low performers.
- Link goals and barriers to the various models and maps to determine where to concentrate design and analysis efforts.
- Analyze knowledge flows in the organization.
  (Hupp, Polak & Westgaard, 1995; Raybould, 2000; Horn, 1999; Beyer & Holtzblatt, 1998; Wiig, 1995).

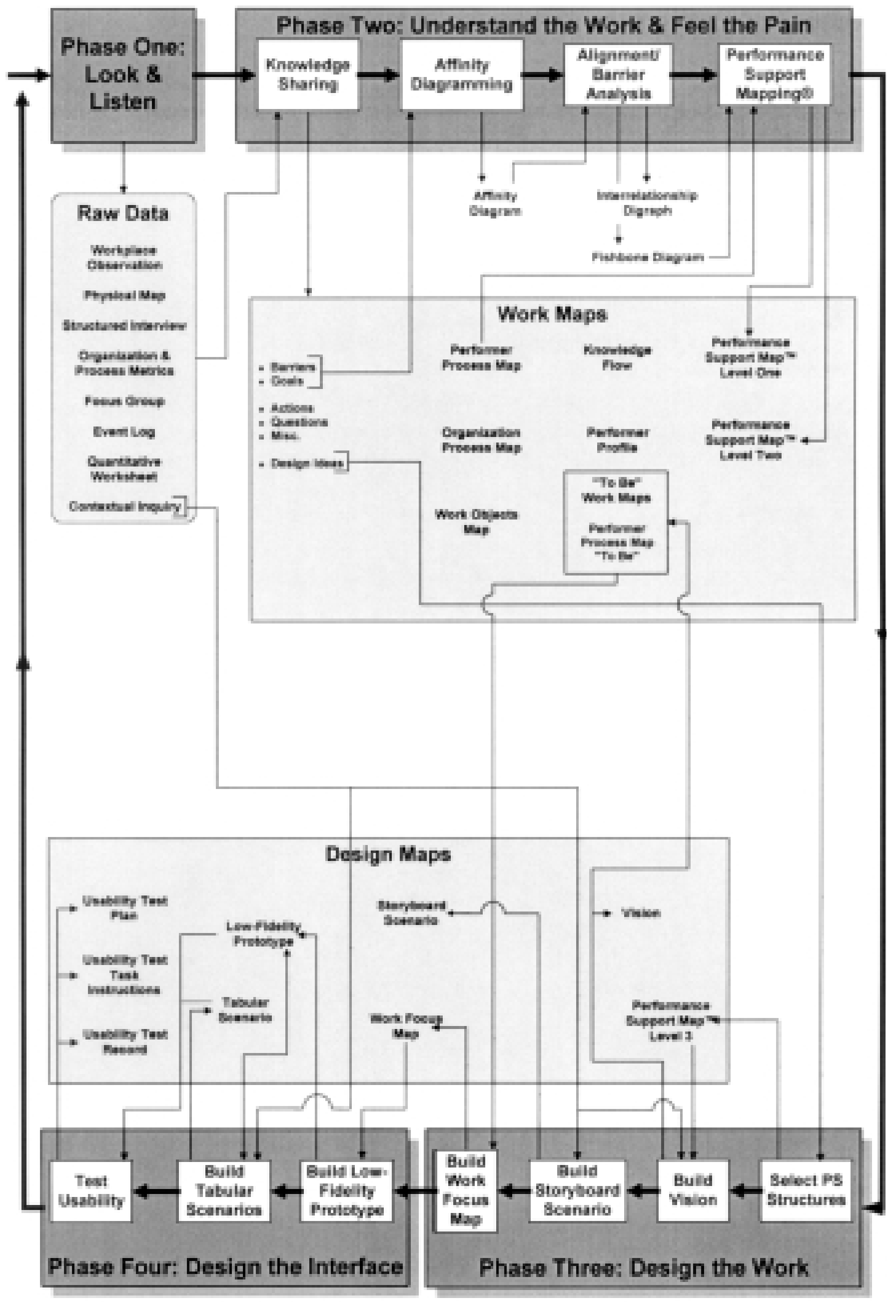

Figure 3. Performance Support Mapping® Methodology.

Phase Three: Design the Work

- Redesign work to remove barriers and roadblocks and to take advantage of technology to provide on-the-job support.
- Build various models and maps to represent the work as it should be and to serve as a blueprint for designing a system interface that will support the performer.
- Create abstract representations of the design, such as user interface flows.
- Envision alternative solutions to work problems and select the most viable solutions (Hupp et al., 1995; Johann, 1995; Raybould, 2000).

These models and maps include adaptations of the Phase Two models. The only difference is that in this phase we are developing "to be" versions versus "as is" versions. In addition, there are some high-level abstractions of the design, such as interface design flows. These facilitate discussions of navigation in the system being designed without getting into the minutiae of the interface design, such as which type of text box to use or where to place buttons on the screen.

Phase Four: Design the Interface

- Design the interface of the new system using low-fidelity (paper-based) or high-fidelity (computer-based) prototyping techniques.
- Evaluate the design using an expanded set of performance-centered design heuristics.

- Test the new interface with job performers to prove that the new system meets performance improvement goals.

The basic techniques for interface design and testing are described in the Human Computer Interaction (HCI) literature (Wiklund, 1994). The key difference is in the use of expanded heuristics, which is used during design and testing (Raybould, 2000). In the design process the heuristics are used in a cognitive walkthrough of the interface and in screen-by-screen heuristic evaluation to predict usability problems in advance. The same heuristics are applied during usability testing.

Performance Support Continuum

A consequence of integrating disciplines is integrating what were previously separate interventions. Not all support can be embedded within a tool or system, and it is not always possible to make products obvious (Horton, 1994; Norman, 1992). Support needs to be provided in a continuum (Raybould, 1998b). These include support embedded in the tool or software interface (intrinsic) to support that is linked to the tool (extrinsic, such as wizards, cue cards, coaches, advisors, and help) to support that is separate from the tool (external, such as tutorials, computer-based training, peer support, and telephone hotlines). As support moves further from the tool and requires more time off the job it becomes less powerful and more expensive to use. As support moves closer to the tool and in the process becomes more granular, it becomes more powerful to use and less expensive in terms of lost time on the job. With reference to Figure 4, the most efficient way to develop a support strategy is to start by building those structures on the right of the continuum and progressively move to the left when a particular structure proves infeasible. This is the performance-centered design approach. This is in direct conflict with traditional approaches that start with interventions at the left and move towards the right (e.g., if there is sufficient budget and time). E-commerce companies are also learning that not all transactions can be completed via the Internet alone. In some cases, human intervention is required at a certain point in the transaction. For cost reasons, these organizations are selecting structures from the continuum from right to left.

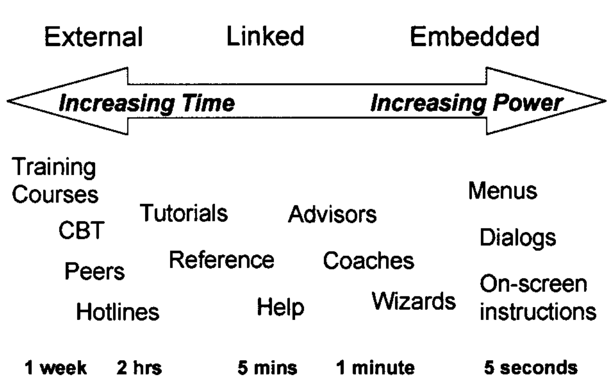

Figure 4. Performance Support Structure Continuum.

Group Processes

Another consequence of integrating disciplines is increasingly commingling people with very different skills on a single project team. Performance support engineering experience emphasizes the critical importance of group processes during design. There are multiple phases in the development methodology in which group processes are important. A key problem in many projects is misalignment of organizational goals, which results in considerable project delays as various factions emphasize their own version of the project's goals and evaluate the emerging design from their own perspective. Therefore, a key element of the performance support engineering process is to align the goals of all stakeholders. Group processes such as brainstorming, affinity diagramming, interrelationship digraphs, and fishbone analyses have proved particularly useful in gaining this alignment.

Alignment within as well as external to the project team is also critical to success. Many design projects have gone astray because the design team has splintered into factions each with its own version of the design. As the design process proceeds, these multiple versions become more and more difficult to reconcile, leading to duplication of effort and unnecessary shifts in design direction. There are several techniques for avoiding these obstacles—for example, ensuring all design options are fully explored and evaluated at the very early design stages, and group commitment obtained before proceeding with any single option.

Another aspect to alignment is alignment of understanding of the work as the early knowledge acquisition processes are underway. This is a particular problem on larger projects in which there are multiple teams interviewing workers, job performers, subject matter experts, and management. As all team members cannot go on all the interviews, problems can arise when each team starts to get a different perspective on problems. This situation results in each team developing and supporting a different solution. The only solution to this problem has been effective knowledge sharing during the design process by the separate analysis teams using group processes.

Another more difficult problem is how to involve management and job performers in the design process. In practice it is impossible for performance support engineers to gain a full understanding of the knowledge domain of a business and of management concerns during a project. But without this full understanding, how does an engineer move the analysis and design forward? The answer lies in the role taken by the engineer. The role needs to shift from that of a person who does all the analysis and design work individually to that of a facilitator. The performance support engineer needs to create a group environment in which the whole team—including the subject matter experts and job performers—comes up with a clear understanding of the work and develops the design together. This means a trend away from producing multiple revisions of long documents that specify requirements and more toward the concept of a design room, in which representations of work and of the design are posted on the walls of a room in a format that can be easily seen and discussed by all team members simultaneously and that can be easily modified. Another way to think of this is as a design space rather than a design document. Of course, documents are required during the process, but only where appropriate and not in those situations in which group interaction is necessary. To make this process successful, it is important to make representations simple and easily taught.

Systems Approach

One of the most significant and defining characteristics of performance support engineering has always been its systems focus, where the system comprises both computer and human components. Systems thinking has been applied by several related disciplines, for example, in the performance technology community (Robinson & Robinson, 1995), in business process reengineering (Davenport, 1993) and in learning organizations (Senge, 1990). Taking this viewpoint has two key benefits. First, it focuses on measurement of actual results against goals and on providing a feedback loop (Figure 5). In

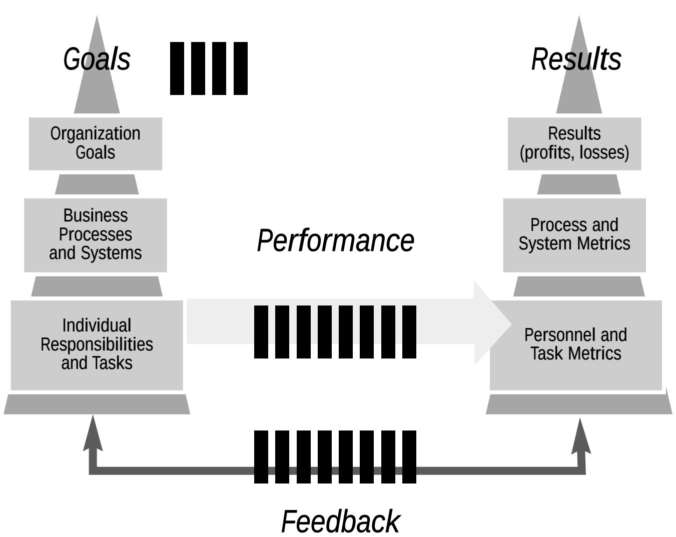

Figure 5. Systems Model with Barriers to Performance.

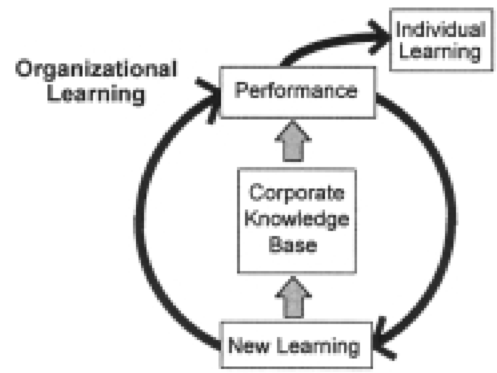

Figure 6. The Organizational/Performance Learning Cycle.

this way the combined human/computer system becomes self-regulating. In many cases it is possible to build feedback into a performance-centered tool or system used on the job. For example, a sales system might provide feedback on individual and group sales performance, or a customer service system might provide feedback on customer satisfaction.

Second, this systems viewpoint focuses on finding all potential causes for failing to meet goals, not just those related to the computer system. The other causes might include problems and barriers with the business processes, the organization and its structure, the incentive and motivation systems, or bottlenecks in feedback loops. By identifying these factors, solutions can be found and the barriers overcome.

Another systems viewpoint is the performance/learning cycle model (Figure 6) (Raybould, 1995), which relates performance-centered design concepts to those of the learning organization (Senge, 1990). Both the generic systems model and the organizational performance learning cycle model influence the raw data gathered during the analysis phase of the generic performance support engineering development cycle shown in Figure 2 and the Performance Support Mapping® development cycle shown in Figure 3.

Focus on Goals

A key part of this systems model is the focus on goals during all phases of the process. Recent trends in the HCI community—such as contextual design (Beyer & Holtzblatt, 1998), usage-centered design (Constantine, 1995), and essential modeling (McMenamin & Palmer, 1984)—have also moved in the same direction. These approaches look at the intents or goals of the job performer rather than just the tasks or actions he or she performs.

Performance Support Mapping® from the performance-centered design community goes one step further by identifying a linkage to organizational goals, and it includes various

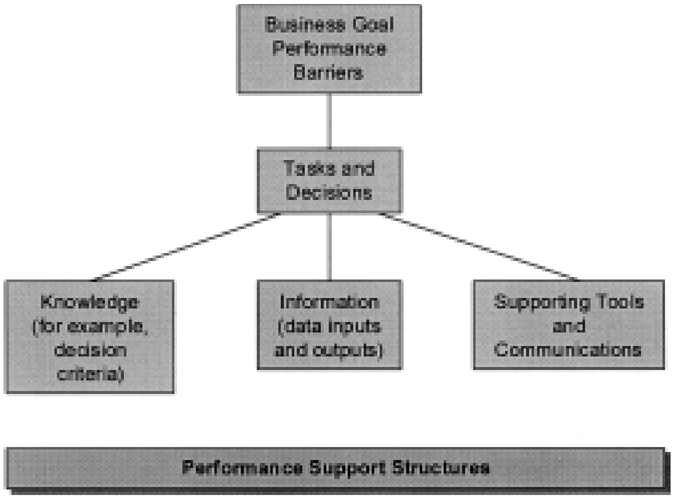

Figure 7. Performance Support Mapping®.

team alignment activities in its development process. This mapping process (Figure 7) also creates a linkage between goals and performance barriers, the tasks and decisions involved in work, and the knowledge, information, and tools needed to support those tasks and decisions. This process makes sure all the factors that might negatively impact the successful completion of work are identified, so that they can be addressed. It also directs further knowledge acquisition efforts (tasks and decisions) that have the most impact on the business because they relate either to the major goals of the organization or to the major barriers the organization is trying to overcome. Finally, this process focuses on the differences between traditional data analysis and the type of knowledge analysis required in performance support engineering. It is particularly important to focus on these differences for those people who have been trained from the perspective of data structures, who do not have training in knowledge structures.

Integration of Knowledge with Tools

A distinct convergence toward the concepts of electronic performance support systems and performance-centered design emerged at the beginning of the 1990s in the human performance technology community. The human-computer interaction field, expert systems field, and technical documentation fields have all been moving closer to those approaches advocated by the performance support community (see Figure 8). Technical books have become interactive electronic manuals, stand-alone expert systems have been embedded in information systems, and instructor-led training courses have become web-based training modules integrated with hyperlinked background reference information.

The pattern clearly shows an increasingly close integration of knowledge and support resources into the tools that people use. Information systems are moving from the data management age into the knowledge management age, in which the foundation of most information systems—the database—is being augmented by a knowledge base accessible via a performance-centered interface. The knowledge base is maintained and enhanced via a knowledge-management system (see Figure 9). A simplistic way of differentiating between these concepts is as follows: "66" is data; "Pat is 66 years old" is information; "Persons older than 65 are eligible for benefits" is knowledge. This trend is still in the embryonic phase, and we are only just learning the issues involved in structuring and managing knowledge bases.

The focus for deciding which tools and which specific areas and domains of knowledge to integrate into the performance-centered tool or system is determined through a thorough understanding of business and individual goals described earlier.

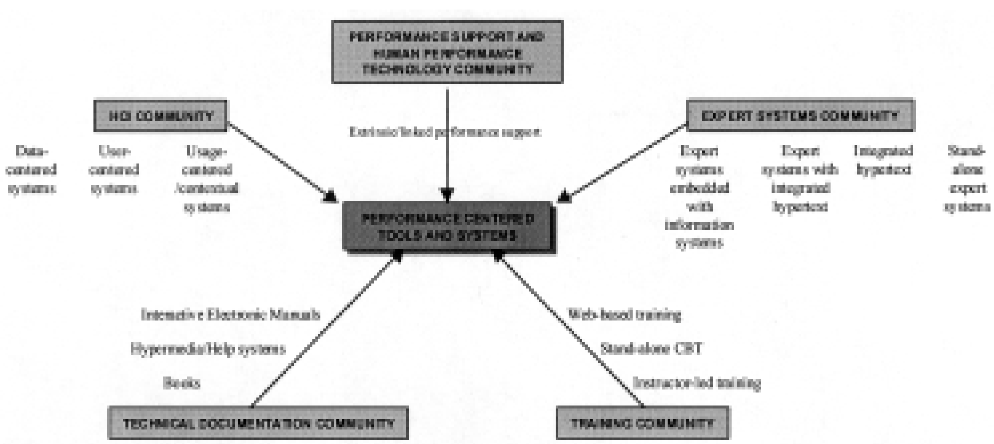

Figure 8. Convergence to Performance-Centered Tools and Systems in Various Professional Communities.

Expanded Heuristics List

Finally, many of the above characteristics of performance-centered systems are repeated in a set of design heuristics or rules of thumb. Traditional user-centered design typically uses the order of eight to ten major heuristics (Molich & Nielsen, 1990). The performance-centered approach employs a larger number of design principles, for example, the 22 design principles in Figure 10 or Gloria Gery's Attributes and Behaviors of Performance-Centered Systems

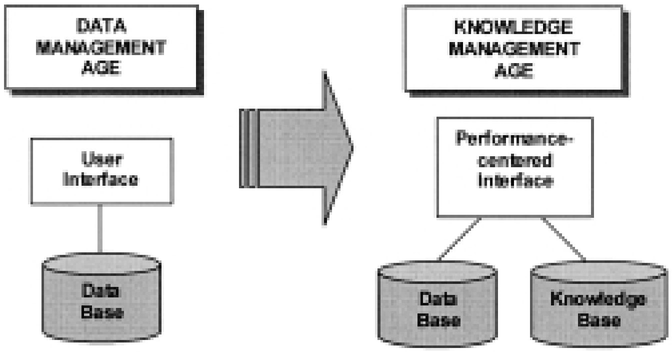

Figure 9. Evolution to Integrated Data and Knowledge Management.

(Gery, 1991). This larger set of principles takes into account principles involved in systems thinking, such as goal establishment and feedback. Another key set of principles derives from question-answering theories in cognitive science (Graesser & Franklin, 1990; Lauer, Peacock & Grasser, 1992). A key difference between the traditional user-centered design principles and the performance-centered design principles is that the latter applies both to the work and to the computer-human interaction, not just to the computer-human interaction alone. For example, the heuristic "provides feedback" when applied to the work would require the designer to give a salesperson feedback on weekly sales results. When applied to the design of the human-computer interface, this principle would result in a confirmation message when an order button was pressed stating, "Order has been sent to warehouse."

Summary

Over the past ten years there has been a gradual convergence of thinking among practitioners in the performance support community on how to develop performance-centered systems. Other professional disciplines have also been moving toward the performance-support approach. This analysis and design process, refined over many performance support projects, is a hybrid of techniques from multiple disciplines and results in a series of integrated interventions in a performance support continuum. Since the project team draws together skills from multiple disciplines and integrates job performers and subject matter experts into the analysis and design process, group processes are particularly important. Performance support engineering takes a systems viewpoint in which the system comprises both computer and human elements. Compared with traditional user-centered design processes, performance-centered design uses a wider range of design heuristics that take into account both this systems viewpoint and the needs of knowledge management. This process, or processes very similar to this, will be the foundation for designing performance-centered systems at the beginning of the 21st century and will be critical to developing successful e-commerce systems and to providing the means for a knowledge management system to improve business results.

Principle	***Description***
Advance warning	Provides advance warning of consequences.
Affordance	Visual appearance suggests use.
Answers descriptive questions	Answers: "What is this?" "What are the differences?"
Answers functional questions	Answers: "What does this do?"
Answers procedural questions	Answer: "How do I?"
Automates tasks	Automates tasks wherever possible.
Captures best practice	Captures the best practice of the experts.
Consistent	Is consistent.
Feedback	Gives feedback on what you've done or where you've been.
Forgiving	Lets you make a mistake and go back to a previous state.
Goal establishment	Helps establish what you can or want to do, or where to go.
Interprets	Answers: "Why did that happen?" "How did that happen?"
Layered	Provides increasing levels of detail to suit diverse audiences.
Matches flow of work	Matches how work presents itself to you.
Minimizes translation	Minimizes interpretation of special terms.
Proactive support	Proactively monitors and evaluates to provide support when needed.
Recognition	Relies on ability to recognize, rather than recall, knowledge.
Relevant	Omits irrelevant information.
Resources	Provides access and links to all resources and tools needed.
Search	Lets you search for answers to questions
Stimulus response path	Provides an unbroken path from stimulus to response.
Task or process focused	Directly shows the structure of the task or process.

Figure 10. Performance-Centered Design Heuristics.

References

Beyer, H., & Holtzblatt, K. (1998). *Contextual design.* San Francisco: Morgan Kaufmann Publishers, Inc.

Constantine, L. (December 1995). "What do users want? Engineering usability into software." *Windows Tech Journal, 4* (12), 30-39.

Davenport, T.H. (1993). "Enablers of process innovation." *Process Innovation: Reengineering Work Through Information Technology.* Harvard Business School Press.

Des Jardins, S., & Davis, H. (1995). "Electronic performance support systems (EPSS): Making the transition." *IVLA Conference Readings.*

Dutta, S. (1993). *Knowledge processing and applied artificial intelligence.* Oxford, England: Butterworth-Heinemann Ltd.

Gery, G. (1991). *Electronic performance support systems.* Tolland, MA: Gery Associates.

Graesser. A.C., & Franklin, S.P. (1990). "QUEST: A cognitive model of question answering." *Discourse Processes 13,* 279-303.

Horn, R. (1999). *Participant's manual for developing procedures, policies and documentation.* Waltham, MA: Information Mapping, Inc.

Horton, W. (1994). *The icon book: Visual symbols for computer systems and documentation.* New York: John Wiley & Sons.

Hupp, K., Polak, C., & Westgaard, O. (1995*). Designing work groups, jobs, and work flow.* San Francisco: Jossey-Bass Publishers.

Johann, B. (1995). *Designing cross-functional business processes.* San Francisco: Jossey-Bass Publishers.

Kelly, R. (1991). *Practical knowledge engineering.* Bedford, MA: Digital Press.

Lauer, W.T, Peacock, E., & Grasser, A.C. (1992). *Questions and information systems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Marion, C. (May 1997). "Implementing performance-centered design." *News & Views,* Society for Technical Communications.

McGraw, K., & Harbison-Briggs, K. (1989). *Knowledge acquisition: Principles and guidelines.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

McMenamin, S., & Palmer, J. (1984). *Essential systems analysis.* Englewood Cliffs, NJ: Prentice-Hall Yourdon Press.

Molich, R., & Nielsen, J. (1990). "Improving a human-computer dialogue." *Communications of the ACM, 33* (3), 338-348.

Norman, D. (1992). *Turn signals are the facial expressions of automobiles.* Reading, MA: Addison-Wesley Publishing Company.

Raybould, B. (1995). "Performance support engineering: An emerging development methodology for performance technologists and enabler of organizational learning."

Electronic Performance Support Systems: Information Kit. Washington, DC: International Society for Performance Improvement.

Raybould, B. (1998a). "EPSS and your organization." *Practical Guidelines for Training and Development Professionals,* 9806.

Raybould, B. (1998b). "Performance support structures." *Practical Guidelines for Training and Development Professionals,* 9806.

Raybould, B. (2000). Performance Support Mapping® workshop participant's handbook. Ariel PSE Technology (www.arielpse.com).

Robinson, D., & Robinson, J. (1995). *Performance consulting.* San Francisco: Berrett-Koehler Publishers, Inc.

Senge, P.M. (1990). *The fifth discipline. The art and practice of the learning organization.* New York: Doubleday/Currency.

Swanson, R. (1994). *Analysis for improving performance.* San Francisco: Berrett-Koehler Publishers, Inc.

Tansley, D., & Hayball, C. (1993). *Knowledge-based systems analysis and design.* Hertfordshire, England: Prentice Hall International (UK) Ltd.

Wiklund, M. (1994). *Usability in practice. How companies develop user-friendly products.* Cambridge, MA: AP Professional.

Wiig, K. (1995). *Knowledge management methods.* Arlington, TX: Schema Press.

Winslow, C., & Bramer, J. (1994). *FutureWork.* New York: Free Press.

Related Readings

Gery, G. (1995). "Attributes and behaviors of performance-centered systems." *Electronic Performance Support Systems: Information Kit.* Washington, DC: International Society for Performance Improvement.

Barry Raybould is one of the founding pioneers in the field of electronic performance support systems and performance-centered design. He is a recognized leader in methodology in this field and pioneered the performance-support engineering discipline for developing performance-centered systems within a knowledge management framework. He founded one of the leading consulting companies in this field and has recently completed a two-year effort to document and create a new Performance Support Mapping® training program to help organizations rapidly build up in-house expertise in performance support engineering and performance-centered design. Barry may be reached at BRaybould@arielpse.com or www.arielpse.com.