Performance Improvement Journal

Performance Support Engineering: Building Performance-Centered Web-Based Systems, Information Systems and Knowledge Management Systems in the 21st Century

Barry Raybould

Ariel PSE Technology

Abstract

With the meteoric rise of the Internet and e-business, web-based systems for consumers and intranets for internal knowledge management systems 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 and not even an option for many e-business applications. Therefore the ability to design software systems from a performance-centered viewpoint is becoming even more urgent. Given this situation the major question facing organizations today is not whether to do performance-centered design, but how to get it done. This paper is a follow up to Performance Support Engineering: An Emerging Development Methodology for Enabling Organizational Learning (Raybould 1995). Since that paper was written the body of experience in developing performance-centered systems has grown significantly and considerable progress has been made by practitioners in elaborating the embryonic development methodology outlined in that 1995 paper. This paper summarizes the convergence of thinking among various professional disciplines that has taken place in analysis and design methodologies, and describes seven key elements of the now emerged performance support engineering development methodology. It is envisioned that this process, or processes very similar to this, will be the foundation for designing performance-centered systems at the beginning of the 21st century, whether they be consumer web applications, intranets, knowledge management systems, business information systems, or any other systems designed to support work.

Introduction

With the meteoric rise of the Internet and e-business, web-based systems for consumers and intranets for internal knowledge management systems are becoming a major focus of software engineers and human performance. 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 means that workers must achieve competency in much shorter timeframes. When software is delivered directly to 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 options 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.

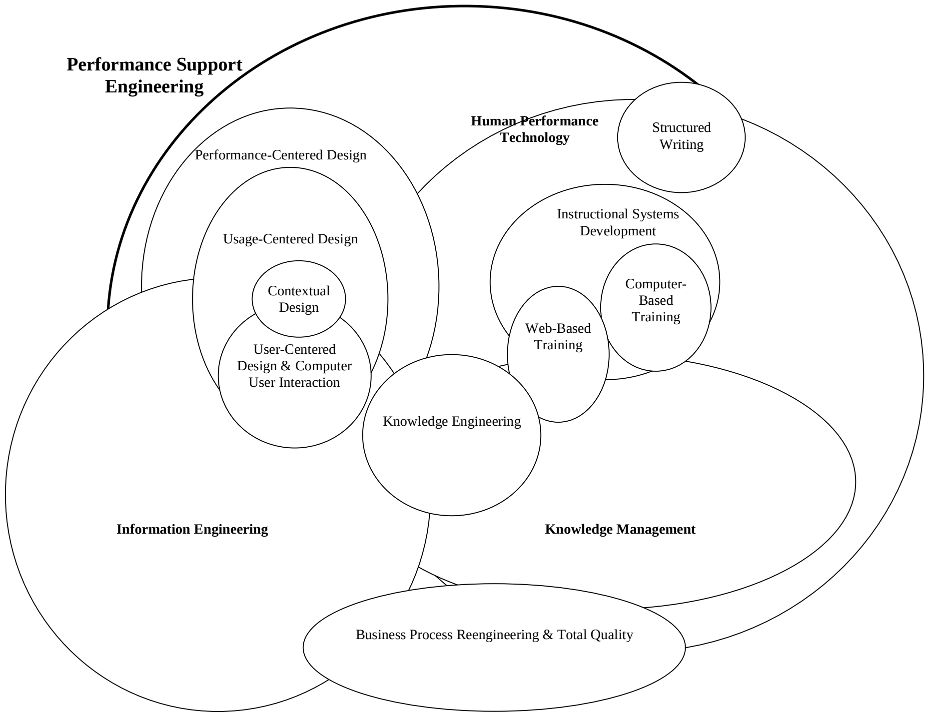
The same principles hold for knowledge management systems: just making knowledge available electronically is not sufficient. Only by building a performance-centered interface to the knowledge base is the knowledge rendered useful to achieving business goals. The major question facing organizations today is not whether to do performance-centered design, which was adequately addressed in the 90s (Gery 1991), (Winslow & Bramer 1994), but how to get it done.

This paper is a follow up to Performance Support Engineering: An Emerging Development Methodology for Enabling Organizational Learning (Raybould 1995). Since then the body of experience in developing performance-centered systems has grown significantly and practitioners have made considerable progress in elaborating the methodology outlined in the 1995 paper. 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.

The seven key elements discussed in this paper are:

1. Hybrid process
2. Performance Support Continuum
3. Group Processes
4. Systems Approach
5. Focus on Goals
6. Integration of Knowledge with Tools
7. Expanded Heuristics List

1. Hybrid Process



Over the past ten years performance support development has progressed from something of an art to a much more structured and documented methodology (Marion, 1997), (Des Jardins and Davis, 1996). In parallel, the HCI community has been moving its methods closer to performance support concepts with its trends away from user-centered design towards *usage*-centered design (Constantine, 1995) and contextual design (Beyer and Holblatz, 1998). Figure 1 shows the relationship between performance support engineering and various other professional disciplines.

Figure 1. Relationship between performance support engineering and 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). This ensures that barriers to performance are exposed and prevents designers from making erroneous assumptions about the nature of the work that might lead to inadequate design.
- The processes are more deliverable-centric rather than sequential. They revolve around a set of models, maps and representations of the work and of the design that are continually refined in an iterative process during the design and development phases as the project team learns more about the work and the emerging design.

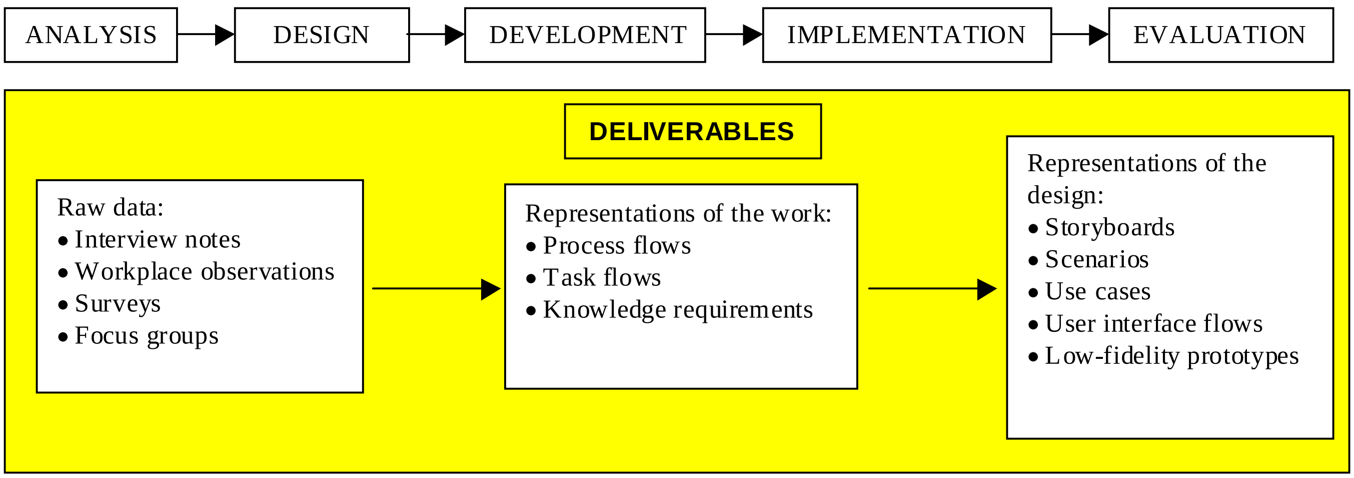


Figure 2. Generalized methodology

An example is the Performance Support Mapping® methodology (Raybould 2000) shown in Figure 3. It comprises elements of information engineering, business process reengineering, 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. The four phases of activities include:

Phase One: Look and Listen includes:

- observing current working environments and gathering data from job performers and their managers to hear their view of the work, its goals and barriers;
- talking to the management of the organization to understand their goals and what is driving the business;
- conducting surveys to gain more statistically significant data on important aspects of the work; and
- conducting 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 processes for performance technology projects (Robinson & Robinson 1995) and (Swanson 1994), knowledge acquisition techniques from the expert systems fields (McGraw 1989), (Kelly Jr 1991), (Tansley and Hayball 1993), and (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 includes:

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

Phase Three: Design the Work includes:

- redesigning work to remove barriers and roadblocks and to take advantage of technology to provide on-the-job support;
- building 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;
- creating abstract representations of the design such as user interface flows; and
- visioning alternative solutions to work problems and selecting the most viable solutions (Hupp, Polak and Westgaard, 1995), (Johann, B. 1995) and (Raybould 2000).

These models and maps include adaptations of the Phase Two models. The only difference being that in this phase we are developing ***To Be*** versions vs. ***As Is*** versions. In addition are added 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 includes:

- designing the interface of the new system using low-fidelity (paper-based) or high fidelity (computer-based) prototyping techniques;
- Evaluating the design using an expanded set of performance-centered design heuristics; and
- Testing the new interface with job performers to prove the new system meets performance improvement goals.

The basic techniques for interface design and testing are described in the HCI literature (Wiklund, M. 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.

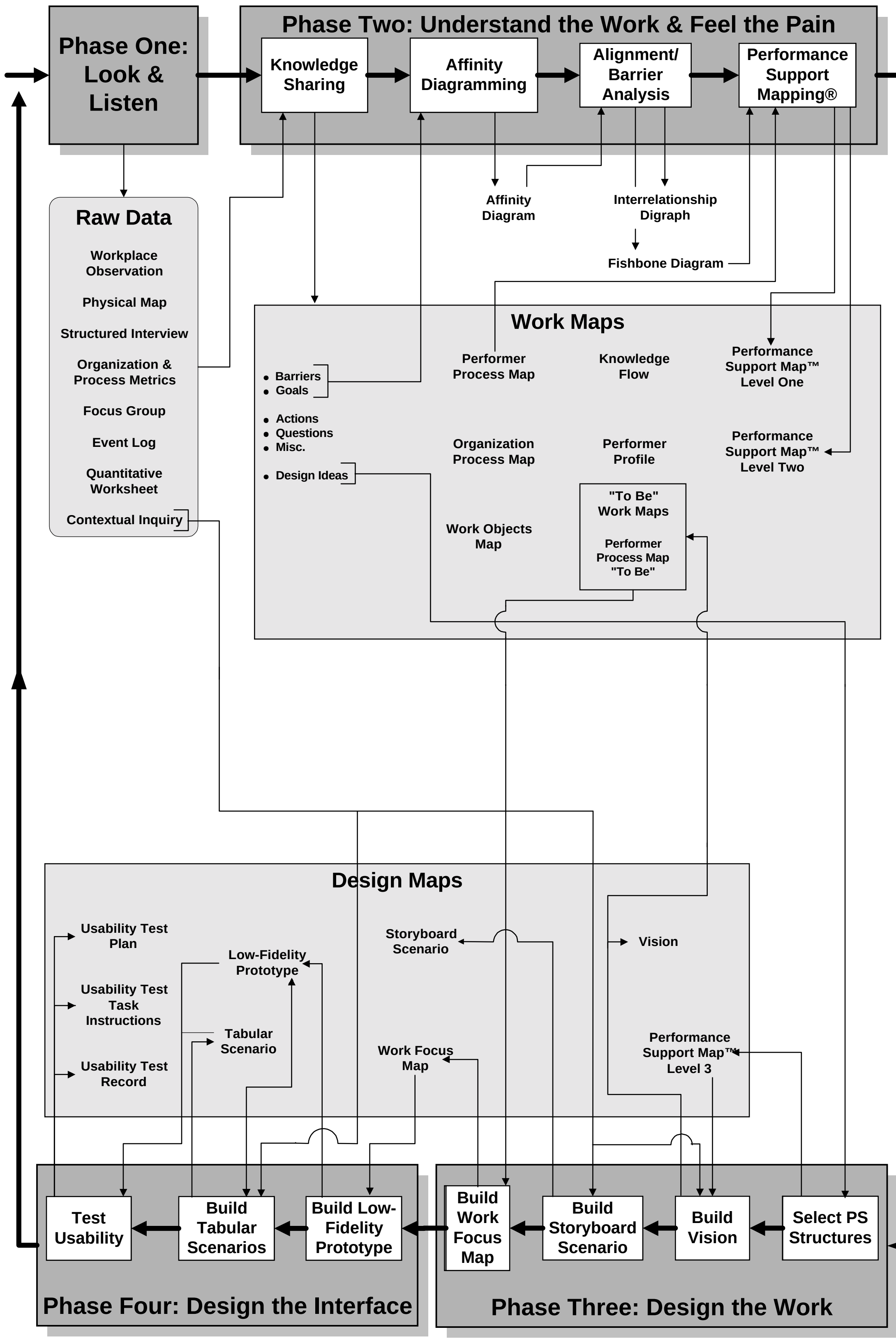
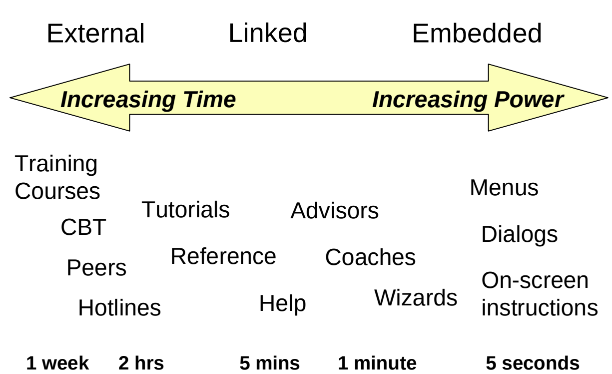


Figure 3. Performance Support Mapping® Methodology

2. 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) and (Norman, 1992). Support needs to be provided in a continuum (Raybould, 1998). 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, more granular, it becomes more powerful and less expensive in terms of lost time on-the-job to use. 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 to 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



3. 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 where group processes are important. A key problem in many projects is misalignment of organizational goals. This 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 to gain this alignment.

Alignment within as well as without 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 large shifts in design direction. There are several techniques for avoiding these obstacles, for example by 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 the larger projects in which there are multiple teams interviewing workers, job performers, subject matter experts and management. Since all team members cannot go on all the interviews problems can arise when each team starts to get a different perspective on the problems. This leads to each team developing and supporting a different design solution. The only solution to this problem has been found to be 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. Without this full understanding how does he or she move the analysis and design forward? The answer lies in the role taken by the performance support engineer. It needs to make a 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 come up with a clear understanding of the work and develop the design together. This is a similar trend that we have seen in Information Systems design techniques with JAD (Joint Application Design) sessions and with various rapid prototyping methodologies, and contextual design processes. This means a trend away from producing multiple revisions of long documents which specify requirements, and more towards to 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 the 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 you need a lot of group interaction to explore a particular problem or solution. To make this process successful it is important that the representations are simple and easily taught to the team members who have not been formally trained in performance support engineering.

4. 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, 1995) and in the work in business process reengineering (Davenport 1993) and learning organizations (Senge 1990). Taking this systems viewpoint has two key benefits. First it focuses on measurement of actual results against goals and providing a feedback loop (Figure 5). In this way the combined human/computer system becomes self regulating leading to achievement of business goals. 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.

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

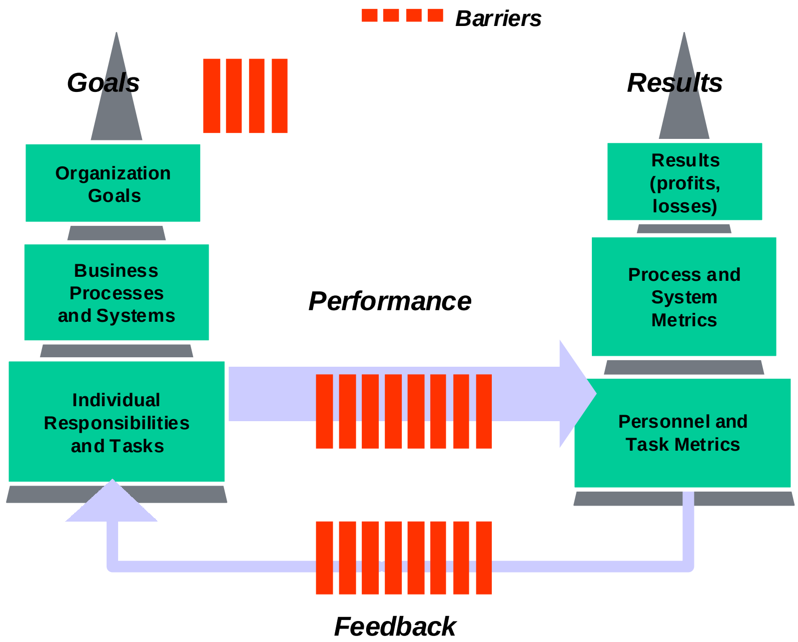


Figure 5. Systems Model with Barriers to Performance.

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 that is gathered during the analysis phase one of the generic performance support engineering development cycle shown in Figure 2 and the Performance Support Mapping® development cycle shown in Figure 3.

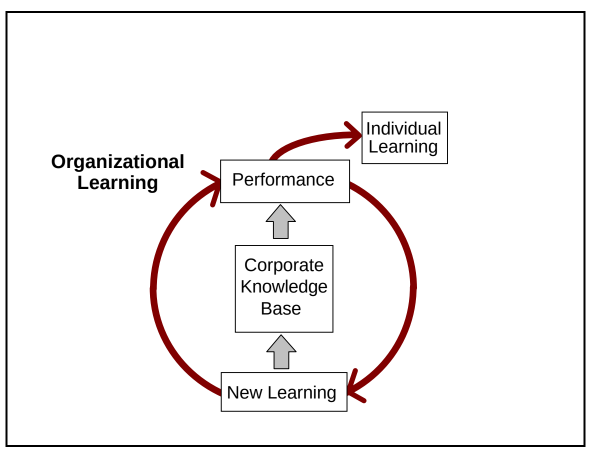


Figure 6. The Organizational/Performance Learning Cycle.

5. Focus on Goals

A key part of this systems model that accounts for the success of performance-centered designs is the focus on goals during all phases of the design and development process. Recent trends in the practices of the HCI community such as contextual design (Beyer and Holblatz, 1998), usage-centered design (Constantine, 1995) and essential modeling (McMenamin and 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 they perform.

Performance support mapping® from the performance-centered design community goes one step further by identifying a linkage to organizational goals as well as to individual goals and includes various 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 that are needed to support those tasks and decisions. This process makes sure you have identified all the factors that might negatively impact the successful completion of work so that they can be addressed. It also directs further knowledge acquisition efforts on the work (tasks and decisions) that have the most impact on the business because they either relate to the major goals of the organization or to the major barriers the organization is trying to overcome. Finally it focuses on the differences between traditional data analysis and the type of knowledge analysis that is required in performance support engineering. It is important to focus on these differences particularly for those people who have been trained from the perspective of data structures and who do not have training in knowledge structures. This is a common disconnect between members of the Information Systems and Training communities.

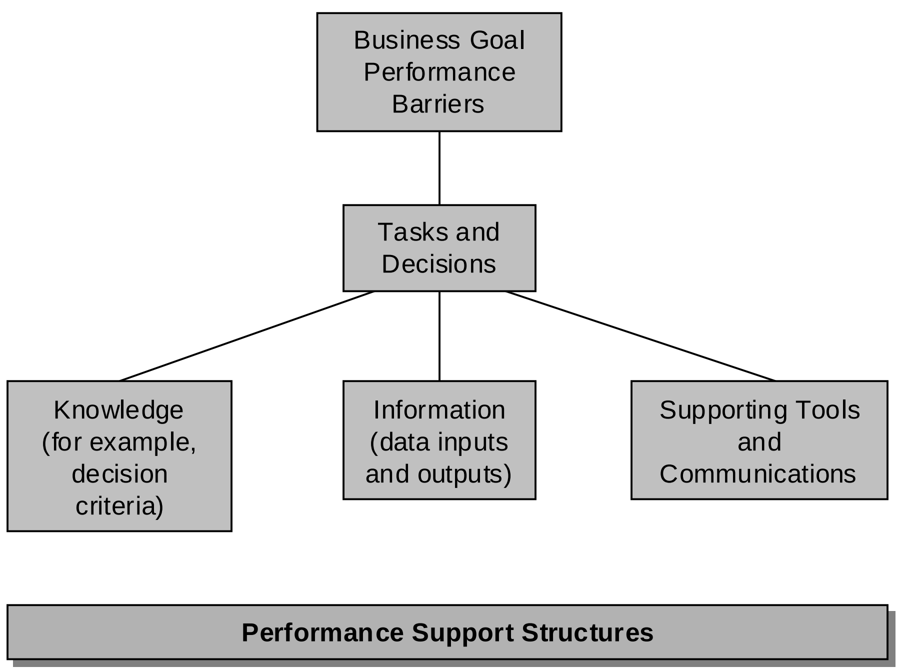


Figure 7. Performance Support Map®

6. Integration of Knowledge with Tools

There has been a distinct convergence towards the concepts of EPSS and performance centered design that emerged at the beginning of the 90's 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 (Figure 8). Technical books have become interactive electronic manuals, stand-alone expert systems have been embedded in information systems, instructor-led training courses have become web-based training modules integrated with hyperlinked background reference information.

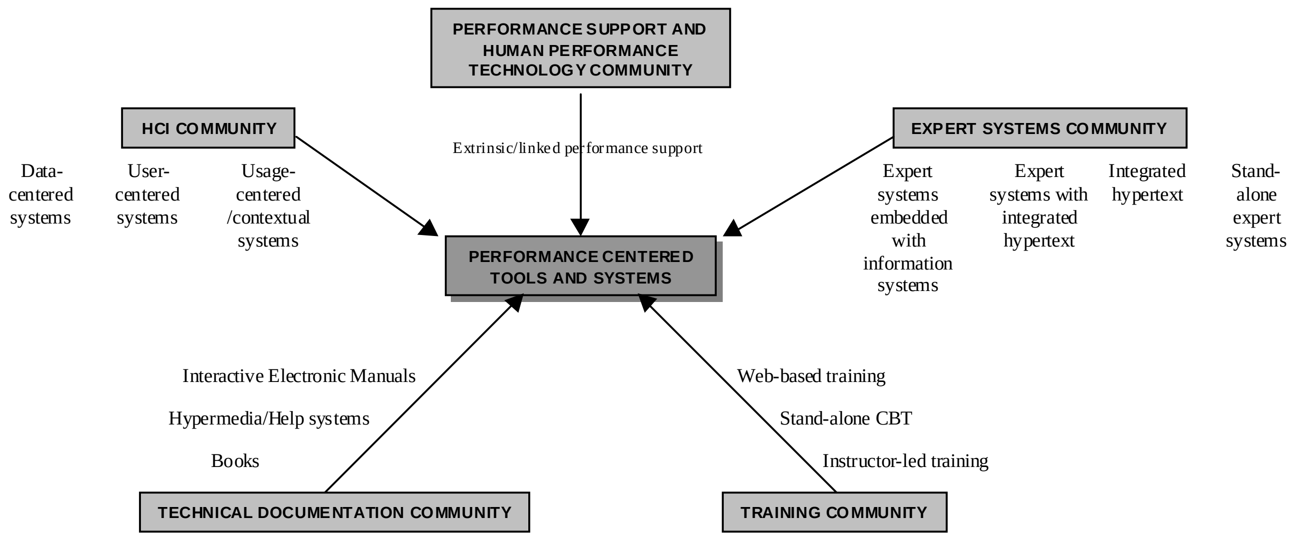


Figure 8. Convergence to performance-centered tools and systems in various professional communities

The pattern clearly shows an increasingly closer integration of knowledge and support resources into the tools that people use. In Information Systems we 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. (A simplistic way of differentiating between these concepts is as follows: "66" is data, "Pat's age is 66" is information, "Persons over age 65 are eligible for benefits" is knowledge). Although this trend has started it is still in the embryonic phase and we are only just learning the issues involved in structuring and managing knowledge bases.

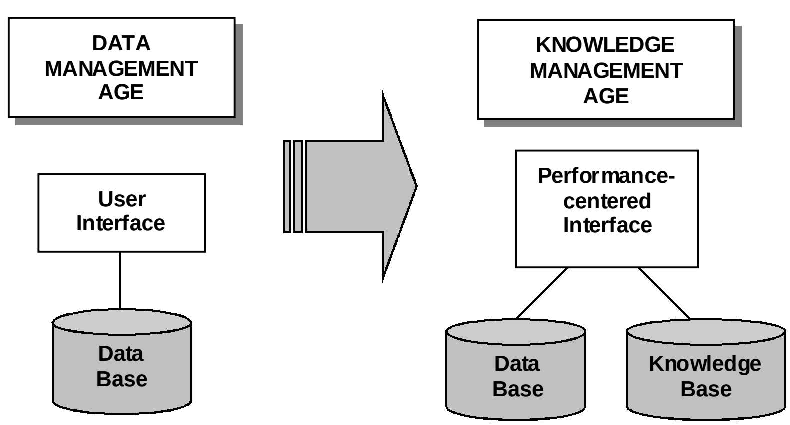


Figure 9. Evolution to integrated data and knowledge management.

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.

7. 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 of the order of eight to ten major heuristics (Molich and Nielsen, 1990). The performance-centered approach employs a larger number of design principles, for example the twenty-two design principles in (Raybould, 1998) repeated here in Table 2, and Gloria Gery's Attributes and Behaviors of Performance-Centered Systems (Gery, 1991). This larger set of principles takes into account principles involved in systems thinking described above such as goal establishment and feedback. Another key set of principles derive from question answering theories in cognitive science (Graesser & Franklin 1990) and (Lauer, Peacock & Graesser 1992), for example the principles: answers descriptive questions, answers functional questions and answers procedural questions. A key difference between the traditional user-centered design principles and the performance-centered design principles is that the latter apply *both* to the work *and* to the computer/human interaction. For example, the heuristic "provides feedback" when applied to the work would require the designer to give a sales persons feedback on their 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".

Table 2. Performance-centered design heuristics

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.
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.

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 towards the performance support approach. In this paper I have described the key elements of an analysis and design process that has been refined over many performance support projects. The methodology 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 heuristicsthat take into account both this systems viewpoint and the needs of knowledge management. It is envisioned that 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 providing the means for a knowledge management system to improve business results.

References

Beyer, H., and Holtzblatt, K. 1998. *Contextual Design*. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Constantine, L. 1995. *What Do Users Want? Engineering Usability into Software*. Windows Tech Journal

Davenport, T. H. 1993. Enablers of Process Innovation. *Process Innovation*. pp 16-18. Boston, MA: Harvard Business School Press.

Des Jardins, S., and Davis, H. 1996. *Electronic Performance Support Systems (EPSS): Making The Transition.*

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. Cambridge, MA: Ziff.

Gery, G. 1995. Attributes and Behaviors of Performance-Centered Systems. *Performance & Instruction*. Washington, D.C.: International Society for Performance Improvement.

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 & Documentation.* Waltham, MA: Information Mapping Inc.

Horton, W. 1994. *The Icon Book: Visual Symbols for Computer Systems and Documentation*. John Wiley & Sons.

Hupp, K., Polak, C., and Westgaard, O. 1995. *Designing Work Groups, Jobs, and Work Flow*. San Francisco, CA: Jossey-Bass Publishers.

Johann, B. 1995. *Designing Cross-Functional Business Processes*. San Francisco, CA: Jossey-Bass Publishers.

Lauer, W.T, Peacock, E., and Grasser, A.C. 1992. *Questions and Information Systems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Kelly, R. 1991. *Practical Knowledge Engineering*. Digital Equipment Corporation.

Marion, C. 1997. *Implementing Performance-Centered Design*. News & Views, Society for Technical Communications, May 1997.

McGraw, K., and Harbison-Briggs, K. 1989. *Knowledge Acquisition: Principles and Guidelines*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

McMenamin, S., and Palmer, J. 1984. *Essential Systems Analysis.* Prentice-Hall Yourdon Press

Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM* **33**, 3 (March), 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. *Performance Improvement Quarterly*. Washington, D.C.: International Society for Performance Improvement.

Raybould, B. 1998. *EPSS and Your Organization*. INFO-LINE Practical Guidelines for Training and Development Professionals, Issue 9806. Alexandria, VA: American Society for Training and Development. June 1998.

Raybould, B. 1998. Performance Support Structures. *EPSS and Your Organization*. pp 7-11. INFO-LINE Practical Guidelines for Training and Development Professionals, Issue 9806. Alexandria, VA: American Society for Training and Development. June 1998.

Raybould, B. 2000. *Performance Support Mapping®, Workshop Participant's Handbook*. Ariel PSE Technology (www.arielpse.com).

Robinson, D., and Robinson, J. 1995. *Performance Consulting*. San Francisco, CA: Berrett-Koehler Publishers, Inc.

Senge, P. M. 1990. *The Fifth Discipline*. *The Art & Practice of the Learning Organization*. New York, NY: Doubleday/Currency.

Swanson, R. 1994. *Analysis for Improving Performance*. San Francisco, CA: Berrett-Koehler Publishers, Inc.

Tansley, D., and 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., and Bramer, J. 1994. *FutureWork*. New York, NY: Free Press.

About the Author

Barry Raybould is one of the founding pioneers in the field of EPSS 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 just 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 can be reached at BRaybould@arielpse.com.