A methodological framework for an object-oriented performance support system

Ian Douglas and Scott Schaffer
Learning Systems Institute, Florida State University

10th of November, 2001

Introduction

For at least the past thirty years, the U.S. military has employed a systematic approach to designing instruction for training and education. The main purpose of systematically designed instruction is to yield reproducible (over many training events) learning experiences. This reproducibility is critical when standardization of task performance is required. A primary problem with executing instructional systems design (ISD) process models, such as the Army's current Systematic Approach to Training (SAT), is that force reductions and attrition over the years has eroded the ISD practitioner knowledge and experience base. This factor, along with increased time pressures to produce training products, has resulted in skipping or shortcutting many of the process steps within the SAT (Branson, Kaufman, Schaffer, 2000).

Incidentally, this is not a problem confined to Army training or the U.S. military. Studies done regarding the practices of ISD professionals in the work environment also suggest that time pressures and management impatience with the ISD process result in an ISD practice that looks a lot different than the theory (Rossett & Czech, 1996). Likewise, the preparation of instructional designers may also be fundamentally limiting the ability of the designers to effectively impact the organization, i.e., add value, by focusing exclusively on learning. Davies (1989) found that designers in the private sector were generally thought to be competent but lacking in promotability due to their lack of business orientation and a proactive approach to solving organizational performance problems (Moller, et. al., 2000).

In addition to the lack of expertise in ISD, there is also a growing discontent among many practitioners with the ISD methodology. In a recent article (Gordon & Zemke, 2000), ISD was criticized for being too slow, not necessary, leading to bad solutions and having an outdated world-view. Whatever the merits of such claims the underlying premise of ISD that there should be some systematic approach to design remains valid, i.e., if you reject traditional ISD you should have something to put in its place. Alternative paradigms and approaches to instructional systems development have been reviewed in some detail by Dills & Romiszowski (1997). They attempted to compile viewpoints that addressed the basic question of what is driving new approaches to ISD: innovative theoretical thinking or changing needs of the market place.

Within the software-engineering domain, there have been similar criticisms of traditional methodologies; this has resulted in a number of alternatives collectively described as 'agile' methodologies (Fowler, 2000). The agile approaches can be characterized as being a collection of suggested features, individual qualities and techniques, rather than a rigid definition of steps and outcomes. This is seen in the agile manifesto (http://agilealliance.org/), which is a collective statement of the common characteristics of agile methods. The manifesto includes statements such as "We welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.", which is in opposition to traditional methodology that tends to focus on defining and fixing requirements prior to building the software.

This paper represents an attempt to describe a methodological framework as an alternative to the traditional ISD model. It is motivated not just by the criticism of traditional ISD, but by three recently developing trends. The first trend is the movement towards focusing on human performance problems and opportunities rather than on solutions. Traditional thinking primarily focuses on one performance support solution, i.e. training. There are a number of emerging performance support alternatives to training, e.g. using personal digital assistants to access on-the-job performance support, automating tasks and redesigning equipment with improved human factors engineering. The second related trend is in the nature of the roles people are expected to play within organizations, which are increasingly varied, complex and changing. Traditional approaches to training that assume the learner is passive are no longer effective. For many of the new roles within an organization, information presented in classrooms will go out-of-date quickly, problem solving skills and self-directed knowledge acquisition become more valued. The third trend is the move to increase the cost-effectiveness of learning and performance support systems through sharing and reuse of resources. Traditional methodologies tend to be oriented towards developing integrated systems rather than systems constructed from pre-existing reusable components.

We deliberately use the term framework rather than methodology. As noted above, there has been dissatisfaction towards traditional instructional methodologies and some evidence suggests that they are not often followed. A framework is meant to provide a structure around which a variety of methodologies can be configured for specific groups or situations, rather than a set of rules for a single correct way of developing systems. The key characteristics of the framework are that it is:

- Performance-oriented rather than solution (training) oriented.
- Object-oriented to facilitate the reuse through the emerging knowledge/learning object technologies.
- Based around a set of recommended features

The framework represents a work in progress and a starting point for debate on the new methodological thinking required to complement initiatives such as the Department of Defense's Advanced Distributed Learning initiative.

Orienting on Performance

Training, especially classroom training, has traditionally been the dominant approach to developing human performance in organizations. Advances in the field of human performance technology, accompanied by rapid developments in information technology have now a number of emerging alternatives to classroom training. Pioneers in the field of performance technology such as Tom Gilbert and Joe Harless have long advocated the use of performance frameworks useful in identifying performance system outages. These frameworks include organizational elements such as expectations and feedback, tools and equipment, rewards and incentives, and motivation. The frameworks help analysts identify barriers to accomplished performance. The essence of this approach is to identify valuable behaviors that produce measureable results, and to remove work environment and individual barriers to performance (Gilbert, 1998). Other authors have advocated a performance-centered approach to developing electronic performance support systems (EPSS) (Raybould, 2000; Dickelman, 2000; Gery, 1991). This approach relies quite heavily on analysis of the performer and the total environment in which he/she is expected to accomplish results in order to specify the requirements of the EPSS.

> An EPSS is the electronic infrastructure that captures, stores and distributes individual and corporate knowledge assets throughout and organization to enable individuals to achieve required levels of performance in the fastest possible time and with a minimum of support from other people."
>
> Raybould, B., 1995

A systematic performance-oriented approach involves several levels of decision making before an optimal blend of solutions is selected. Three levels of this kind of decision making are illustrated in Figure 1.

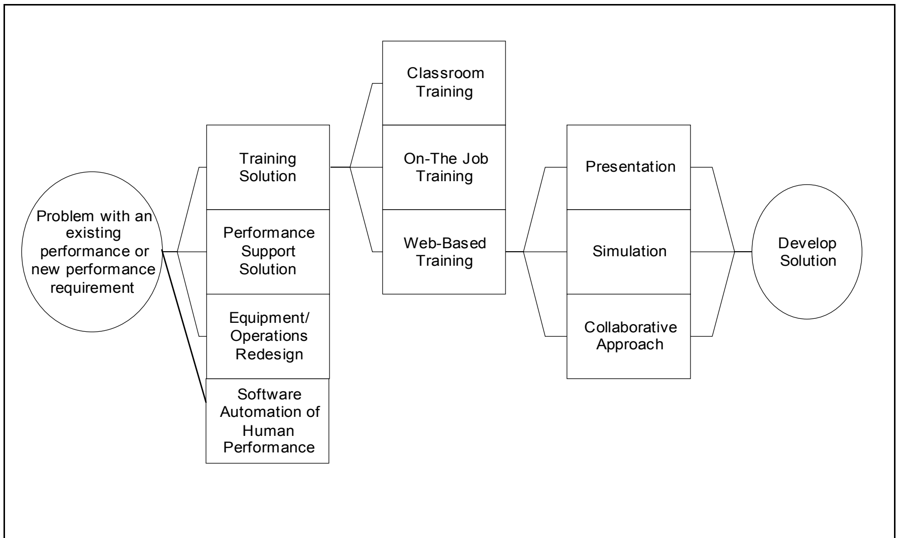

FIGURE 1: SOLUTION SELECTION IN A PERFORMANCE-ORIENTED APPROACH

The decision making process in figure 1 is an ideal one, with careful analysis completed at each stage. What actually happens in many cases is that those making the decision at the first level will jump straight to training as a solution without considering performance support or equipment/operations redesign, or other non-training solutions. In addition to skipping careful analysis at the first level, they may also skip it at subsequent levels, e.g. by jumping to web-based solutions. At the third level, judging by the functionality and appearance of most web-based training, the bias is towards static presentation and multiple choice questions knowledge testing, probably because that is similar to the conventional training most developers/instructors are familiar with. More creative solutions such as simulations and scenario-based learning are less used. Finally, at the level of implementing the chosen solution, decisions on design methods and implementation tools will often be taken based on personal preference or limited knowledge, rather than rational choice.

Orienting on Reusable Objects

There is currently much work involving technology, standards and repositories for learning objects. This work sets the technological framework necessary for a object-based approach to system development (including both training and performance supports systems). The object approach seeks to promote greater cost-effectiveness in systems development through maximizing the reuse of existing materials. There is a danger in having too much of a technology focus in the approach, which can be characterized as the "build and they will come" approach.

> "Typically, for new development approaches.....much of the initial focus is centered on the technology issues-obtaining broad agreements on the desirable properties of the technology, gaining experience in its application, and encouraging a competitive marketplace of solutions based on a set of standards across a range of vendors. However, it is often the remaining two aspects, process and people, that are seen as the dominant elements in taking a new approach 'across the chasm' from early adopter experiences into mainstream use in a wide number of organizations"
>
> Brown, 2000, with reference to Moore, 1999.

The object approach represents a paradigm shift for analysts and designers over their current approach to product development. The majority of current learning and support materials such as instructional manuals and computer-based instruction are designed as large integrated packages rather than as collections of small independent components that can be individually used and modified for multiple purposes. Design thinking needs to move from an approach that is oriented towards creating large integrated packages to one that is built around collections of specialized, reusable and granular components. Systems analysts and designers need to be given the tools and support to help them make the transition to the object approach.

The ideas around the object approach are derived from object-oriented software development and there is a danger of repeating the mismatch of technology and process that occurred here. Although object-oriented programming languages arose in the seventies, it was not until the early nineties that object-oriented approaches to analysis and design began to emerge. Although the syntax of object-oriented languages is relatively easy to grasp, the development of object thinking is often harder to achieve. Object thinking needs to run throughout the process and not just in the construction phase. Object-oriented programming without object-oriented analysis and design can result in large integrated systems being replaced by a small number of large and integrated objects.

Features of a methodological framework

System modeling

Prior to the development of a new system, it helps to create a graphical model of what you are going to build. In architectural design, for example, an architect will create a drawing of a building design. This drawing may be altered many times at relatively low cost before a final plan is used for construction. Once the building is constructed, alterations become difficult and costly. A similar approach is used in many design domains and is facilitated by a commonly used modeling notation. In software systems development the Unified Modeling Language (UML) has recently become popular for modeling software systems prior to their construction (Booch, 1999). In learning and performance systems, modeling is not yet an established part of most methodologies and there is no standard modeling notation. We propose that the potential benefits of modeling are the same within the performance systems as they are within other domains, and that system modeling should be an intrinsic part of any new framework. This will involve the development of a standard modeling notation.

Collaborative development and peer review

Traditional analysis and design can be seen as an expert-centered activity, i.e. an 'expert' in design will work with a subject matter expert in a single geographical location to design a training solution. The Internet has the potential to greatly widen the scope for collaboration of others in the analysis and design process when combined with the modeling approach noted above. If models of problem domains and intended solutions are published openly using a standard, easily understood graphical notation, these can be inspected by geographically distributed stakeholders. Thus, a group of analysis and design experts, subject matter experts and end-users/performers can discuss and pass comment on models and identify flaws prior to final construction. This process does not end with analysis and system development, however. Performers who adapt solutions to fit with a performance requirement become evaluators of the relevance and potency of the solution in terms of meeting performance needs. This evaluation data is shared with or fed back to the performance support team for model enhancements.

Integration with a repository of reusable objects
There are efforts underway to create repositories of reusable objects. The thinking of analysts and designers must be consistent with this object repository approach for reuse to be successful. Developers should begin thinking in terms of objects at the analysis phase rather than at the construction phase. We propose that analysis models should be framed in terms of analysis objects. Analysis objects are descriptions of requirements that can be matched against existing objects in repositories or form the specification for the construction of new objects.

Blended performance solutions
Currently, performance problems are mainly solved with training solutions. By identifying a number of potential solutions to performance problems it is possible, and appropriate, to create a blend of several solutions to a performance problem. Performance models created during analysis have many potential uses. They may be excerpted to create job descriptions and new employee orientation checklists. Developed solutions can also be re-purposed to develop collateral materials that will assist the performer when transferring or applying solutions in the workplace. For example, core skills might be taught in a training course, and a variety of reference materials created or excerpted from them. These job aids or knowledge bases provide performers with on-the-job access to knowledge and procedures for seldom occurring situations.

Parallel iterative development
Traditionally, analysis, design and construction of the whole solution system is completed before delivery. However, in an iterative development approach, different solutions can be selected and developed at different rates. An object-oriented systems approach allows for the decomposition of solution systems into sub-systems and objects, where parts of a system can be created largely from pre-existing objects. Thus some solutions may be identified and designed concurrently with ongoing analysis. This allows multiple, blended solutions to progress to delivery at different rates from others which reduces the overall systems development cycle time, and demonstrates the added value of the analysis process.

Managed deployment with feedback
The performance support developer's responsibility does not end with product delivery. A system of quality assurance should be in place, such that access to a system is managed and monitored, with the user's experience being recorded and fed back into subsequent development and new versions of systems. Evaluation of the products that performers access in terms of their content quality and value-added to the performer and organization, requires management oversight and participation.

Performance-based improvement
The methodological framework should be iterative rather than linear and based on desired performance rather than preconceived solutions. The test of an effective system is not in functional performance or user satisfaction, but in the closing of the performance gaps identified in the analysis across multiple levels of the performance system. There should be an iterative relationship between performance monitoring and performance support system analysis and design.

Rationale management
In Figure 1 we outlined how most performance problems have a range of potential solutions. Despite this fact developers often consider only one type of solution. It is not always the case that those making the decisions are unaware of the alternatives. It is more probable that political, historical and cultural factors dictate that there are certain solutions for certain problems. Thus, it is important for decision-makers to have a defensible approach that

promotes out-of-the box thinking with careful consideration given to potentially cheaper and better alternative solutions that are also politically palatable.

Rationale management (Moran and Carroll, 1996, Burge and Brown, 2000) is concerned with capturing the knowledge that leads to the construction of a system. In particular it requires an auditing of the decision making when choosing problem solutions. Most performance analysts will be able to produce reasons for their analysis decisions; the distinction in giving a rationale is that there should be the following:

- A description of the issues that were addressed prior to the design decision.
- A list of the alternative solutions considered.
- The criteria used in the selection.
- The argumentation used to support each alternative.
- The decision.

The benefit of rationale management is that it requires the performance analyst to use a rational process to make their decision-making explicit. It discourages decisions based on whim or prejudice, as such decision making will not have a clear rationale.

Analysis and design methods and automated support tools
There are a now a large number of software tools that assist in the implementation of automated instruction and a number that manage the delivery of instruction via the web. There are relatively few tools relating to the systematic analysis, design and documentation that should precede construction and delivery, and none that incorporate the emerging object model. There is scope for creating the equivalent of computer aided software engineering (CASE) tools for object based performance analysis and/or instructional design.

Spector and Muraida (1997), Goodyear (1997) and Kasowitz (1997) review much of the previous work in automated design support tools. Spector and Muraida note that there is strong motivation to develop such tools given there is “a lack of instructional design expertise, pressures for increased productivity of designers, and the need to standardize products and ensure the effectiveness of products.” The tools referred to in these reviews are based on the traditional models of instructional design and were mainly experimental; there are very few commercial products of this type. In contrast to this situation, there is now a range of tools designed to assist with object-oriented software analysis and design and some of the thinking embedded in such tools could be used to create new tools to support object-oriented performance support systems.

Software tools should be flexible and use the concept of ‘plug-ins’ to incorporate different analysis and design theories. Thus, emerging ideas of object design and classification can be incorporated e.g. the taxonomy of learning objects as suggested by Wiley (2000), and the internal composition of objects suggested by Merrill (1996) or Clark (1999). Examples of plug-ins from performance analysis include the use of emerging frameworks such as the Balanced Scorecard (Kaplan and Norton, 1995), Intangible Assets Monitor (Sveiby, 1998), and dynamic system modeling approaches.

Performance System Support and Solution Process Model

There are generally recognised stages in systems development: analysis, where you define and understand the problem and solution space; design, where you design the solution; construction, where you build the solution; deployment, where you make the solution available and evaluation, where you evaluate the success of the deployed solution. In different domains there are variations in the terminology and some stages are decomposed or merged. Some methodologies conceive of each stage as separate, the responsibility of different people and something that must be completed before the next stage is begun. More recent thinking tends towards iterative development with the involvement of multi-disciplinary teams, which include stakeholder representatives, in the entire development process.

One way of representing the performance support systems development processes described above is shown in Figure 2.

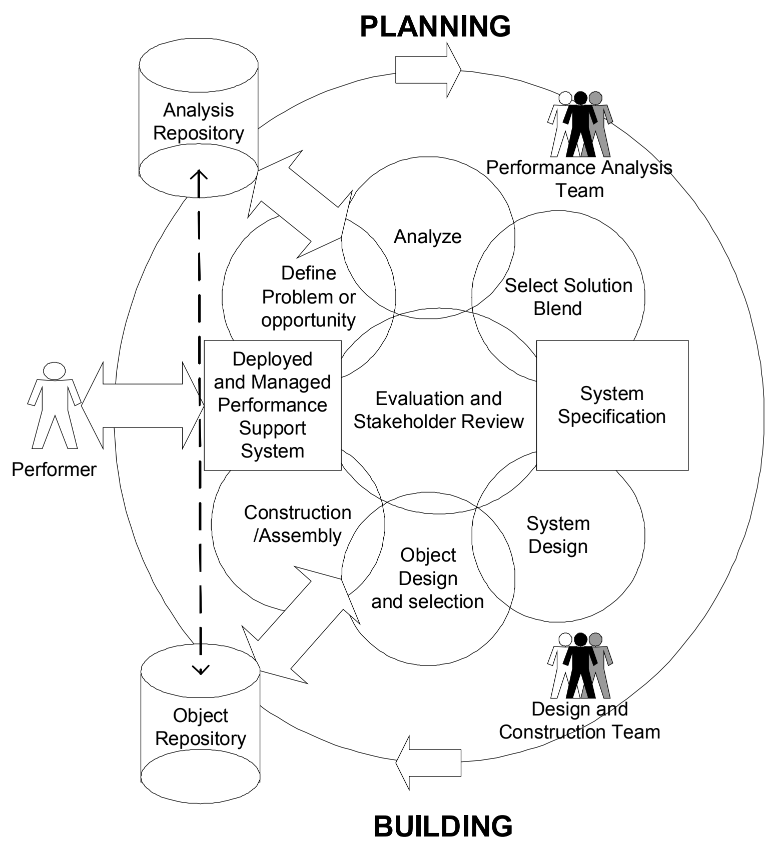

FIGURE 2: *LIFECYCLE MODEL FOR OBJECT-BASED PERFORMANCE SUPPORT DEVELOPMENT*

Two major phases, planning and building, represent the major processes in this framework. These phases are comprised of 7 major processes each supported by several activities. The processes are considered to be generic and applicable to any design and development process. Both the planning and building phases incorporate reuse through sharable database repositories. The building phase is supported by a repository of previously constructed objects and object packages. The planning phase is supported by a repository of previous problem analysis artifacts (diagrams, metrics) that link to appropriate solutions within the object repository.

Planning is defined as the development of a representative visual model of the performance environment and the system in which it operates, analysis of the fit of that model with current practices, and identification of possible solutions that will improve current practice. This representation includes definition of the mission, objectives, and targets across the system levels (strategic, operational, tactical, etc.). Performance analysis begins during this phase as one of four types of analysis: new system (systems analysis, i.e. new equipment, processes, techniques), performance problem-solving (cause analysis), opportunity analysis or problem-finding, and training re-assessment (training impact analysis).

The Analysis process is essentially a representation of the current performance system across strategic, operational, tactical levels. System outages, obstacles to performance, or unexplored opportunities relative to the defined or ideal performance system are identified in order to understand the interdependencies and dynamics inherent within the performance system. Typical approaches to performance analysis have been documented extensively in the literature and many are quite similar in terms of processes to be completed (Gilbert, 1998; Rossett, 1988; Harless, 1987; Rummler and Brache, 1995; and Gaines-Robinson and Robinson, 1995).

Solution blends are selected based on the nature and type of obstacles and strengths identified during analysis. Causes for gaps and enhancers of current performance are matched with solution strands that may represent one of many elements of a total solution package. These solution strands are prioritized based on compatibility with current practices and organizational culture, availability of objects in the repository that may be reused during solution development, and cost. This process closely linked to design with the bridge between the two being system specification. This includes development of a platform for the solution development process (see section 3.3).

The "Building" phase involves the design, development, and diffusion/adaptation of solution packages that will address outages or opportunities in the performance system.

The visual models of performance developed during planning will drive design and construction processes. New layers of visual representations will be created that represent key solution design elements including objectives, activities, and assessments. Performer and environment characteristics previously specified during the planning phase will inform the design and development decisions made during these processes.

Deployment and management of solutions is essentially the delivery of performance and learning support to performers anywhere, anytime. This phase has a large communication component, which assures that solutions are integrated into the performer's environment. This integration is supported by an interface, a performance support management system that supports searching, filtering, and contextualizaton of solution packages. It is expected that significant adaptation of solution packages will be necessary as performers seek to customize solutions to fit their requirements. Adaptation will be supported by a feedback function, which directly links performer feedback to analysts. Possible revisions to solution packages are accomplished by modifying the original visual models or representations of the performance system.

Finally, the core process that drives all other processes is evaluation and stakeholder review. Evaluation is defined as a value judgment that the gap between expected and actual results and processes was closed. This evaluation takes many forms and is alternately formative, summative, and confirmative assessment of the materials, performance, and impact of the processes and solution blends. While planning and building processes and activities do not necessarily occur in any particular sequence, it is expected that each will be completed and evaluated as part of a systemic, iterative approach to performance improvement. Stakeholder review is essentially evaluation without judgment. Iterative and concurrent analysis and design requires a high degree of ongoing adaptation and problem solving. Stakeholders are problem solvers and problem finders with the responsibility of maintaining the flow and effectiveness of the entire process.

The process model proposed by the authors in Figure 2 includes the following:

1. Analysis that is focused on supporting performance and measurable impact of solutions.
2. Specification that includes the design of the solution system architecture, specifically considers a blend of solutions, and explicitly encourages matching solution requirements against preexisting resources in shared repositories.
3. The integration of collaboration and modeling.

This process model serves as the basis of a methodological framework that is both performance and object-oriented. It aims to help developers exploit a range of performance solutions and explicitly considers reuse in systems development. Evaluation is seen as being distributed throughout the framework.

In terms of the research and development to make this framework effective, the work currently undertaken under SCORM and other initiatives focuses on object design, construction and repositories. The whole of the planning phase is currently under-represented in research and development efforts and deployment and management thinking is still dominated by the learning rather than performance perspective.

Object-Oriented Performance Analysis

Performance analysis (see figure 3) focuses on identifying critical elements of the performance environment required for success, and compares those elements to the actual performance across system levels. Success of the organization at the mission and societal levels depends on success at the unit, and individual levels.

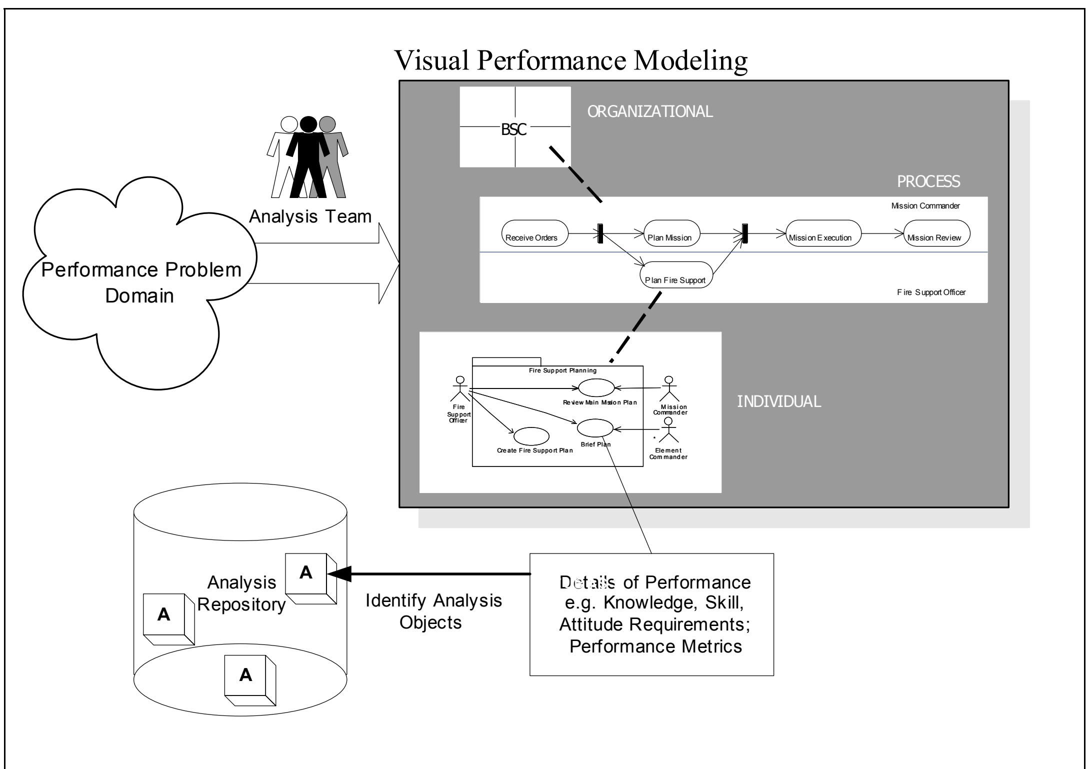

FIGURE 3: PERFORMANCE ANALYSIS COMPONENT OF THE FRAMEWORK

A major role of the performance analysis team is to link results across these levels. For example, a particular organizational unit may be having difficulty with a particular operation, e.g. an infantry battalion setting up a defensive position. The analysis team identifies and maps the roles in this system and may identify a particular role to focus upon, e.g. platoon commander. The analysis team proceeds to specify performance cases, which determine specific goals the role is expected to perform in a given context, e.g. position platoon support weapons while planning a defensive position. In detailing this performance case, the analyst together with collaborators, would identify such things as the knowledge, skills, attitudes, tools, expectations and feedback, and information requirements involved in the performance case.

This process leads to the creation of corresponding analysis objects. An analysis object would contain a description of what was required for a given case and some SCORM complaint meta-data. This will enable matching of the analysis objects against pre-existing reusable objects in SCORM compliant repositories. If no suitable match is found it serves as a guide to developers in the construction of new objects. In addition analysis objects would act as a store of performance metrics for the particular performance cases. These can later be used to identify any improved performance brought about by the solutions selected to support the performance case.

A system for creating performance support should also provide performance for those doing performance support planning and construction. In this and in subsequent stages three support facilities should be integrated into the framework:

1) Performance support to provide advice for analysis teams.
2) Visual modeling tools to support analysis thinking.
3) Collaboration and review.

As noted previously we see modeling as an important tool in the process and in the absence of a specific modeling notation we have adapted UML use case notation (Rosenberg and Scott, 1999) to illustrate performance modeling. We have also used the rectangle notation of Ostyn (2001) to illustrate learning objects, with an unfilled rectangle representing an analysis object, i.e. a specification for a required object.

System Specification

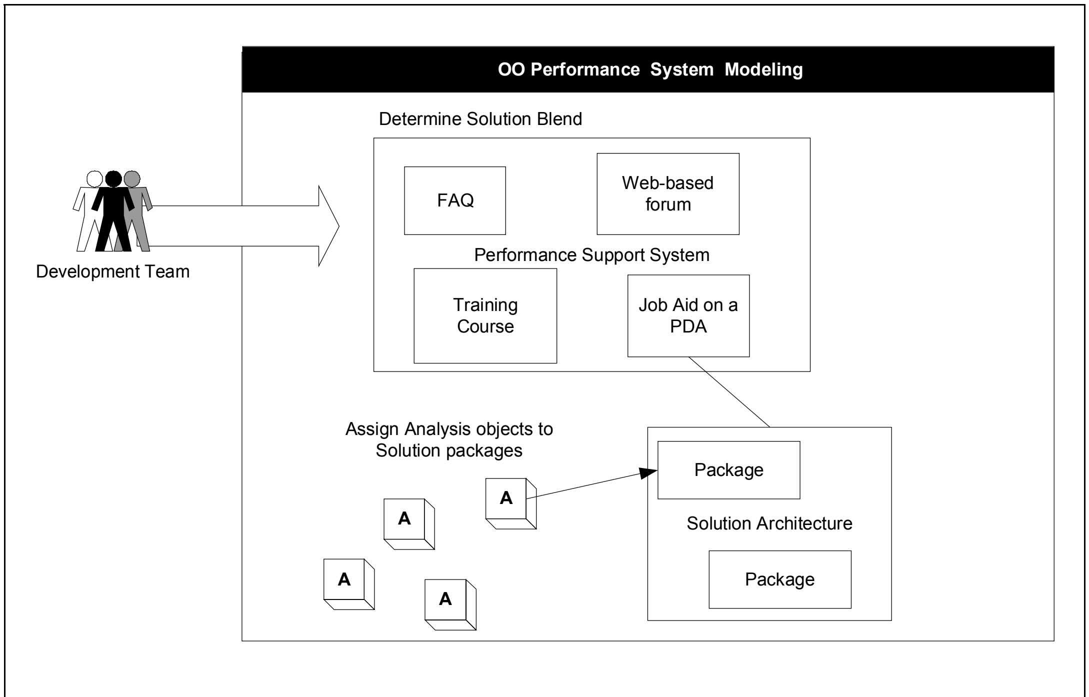

FIGURE 4: SYSTEM SPECIFICATION COMPONENT OF THE FRAMEWORK

In system specification (see figure 4) we select the blend of performance support solutions required. This is done based on the analysis of the problem area and knowledge of the strengths and weaknesses of the different potential solutions. Not all the solutions are necessarily object-based in that FAQ or web forums may also be specified. Process or equipment redesign may also be recommended. Each solution that is part of the blend must be structured into sub-systems and the objects distributed to the appropriate part of the solution blend solution. Some objects many be incorporated in more than one solution.

System specification represents the transition area between the planning and building phases of the life-cycle model in figure 2.

System Development

In system development (see figure 5), there are two parallel tasks. Firstly, at the systems level for each solution selected we must decompose the system into subsystems (e.g. a course unit in the case of a training system), distribute the objects to the subsystems and create the packaging and sequencing structure to bind the objects together. Secondly, for those analysis objects that were not matched against pre-existing objects we must design and develop the content, package it into a SCORM compliant object and submit a copy to the repository.

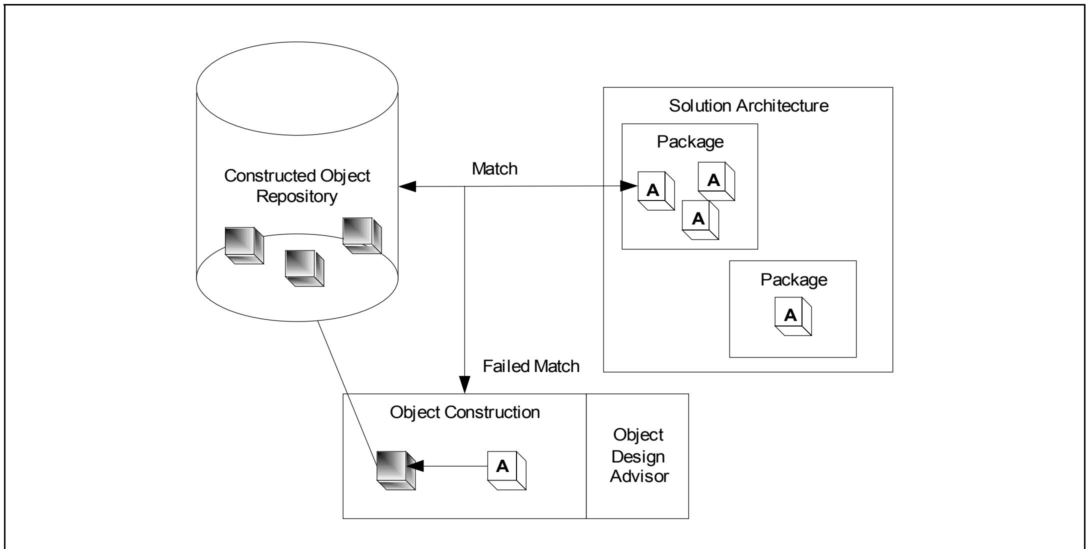

FIGURE 5: SOLUTION CONSTRUCTION

Performance Support Management System

Under current ADL conception there is still a training-oriented bias, in that objects are conceived as learning objects. The main solution considered is computer-based training delivered through a learning management system. As noted in the introduction the trend is towards thinking in terms of performance problems or new performance planning rather than being fixated on the training solution. This not only requires research into how problems are analyzed and solutions are selected, it also requires a reconsideration of how solutions are delivered and managed. Thus, if at the start of the framework we move from training analysis to performance analysis, it makes sense that at the end we move from learning management to performance support management (PSM).

If we look at the current state of performance support solutions we see a lack of integration with systems often developed and delivered independently (see figure 6). It is often left to performers to discover, integrate and synthesize the resources that are available to support them. This can lead to usability problems, as users have to contend with a variety of different systems with different interface designs. There may also be reduced utility in some of the systems due to redundant, irrelevant and inadequate information. Learning management systems solve some of these problems for on-line learning, but they do not solve the problem for performance support in general.

Performance support management is a dynamic performer-defined system that links to a database of packaged performance support systems. Performers using personal digital assistants, wearable computers, or desktop PC's can access available system and subsystem packages. Performance managers can create a customized performance support environment for a particular individual based on the roles they will perform within their organizational unit. We would envisage the possibility for a certain amount of re-sequencing and packaging of systems within this environment. In addition to providing customized access to available performance support systems, the management system should act as a collection point for evaluation information concerning the efficiency and effectiveness of the system.

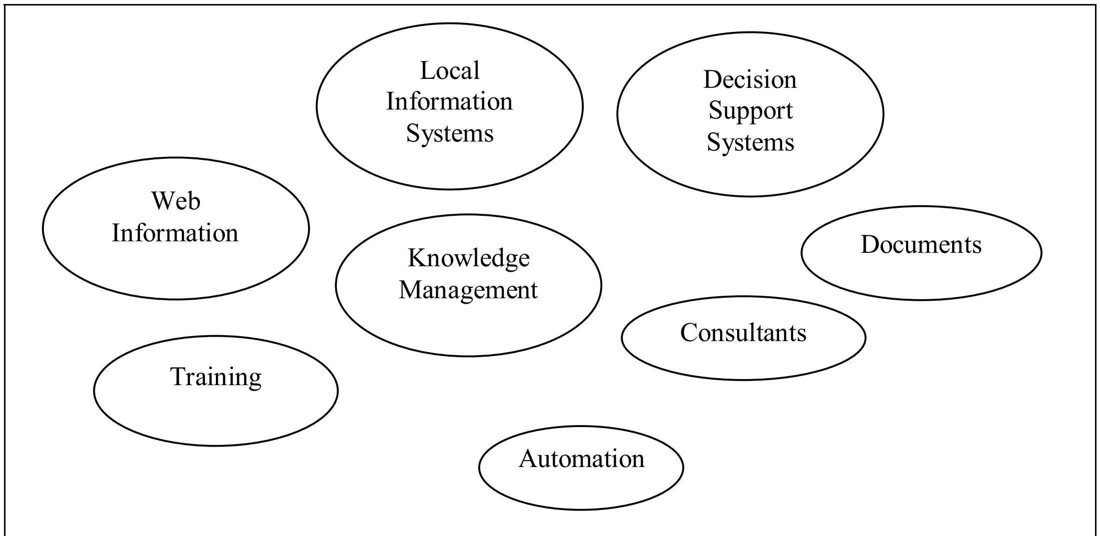

FIGURE 6: SEPARATE SILOS FOR PERFORMANCE SUPPORT

Figure 7 presents an initial model for how a performance support management system might work. Looking at this model from left to right we see a number of layers to the system. The lowest level is composed of the individual assets and objects from which systems are built. The second layer consists of a variety of object-based performance support systems constructed from the low level objects; this would include online learning, frequently asked questions or a help system that can be loaded onto a PDA. What the systems have in common is that they have been designed with a particular performance role in mind. Many of the individual performance roles are derived from the performance roles of different levels of organizational units or systems. The model for how this is structured is created during the performance analysis. The model would define layers of access to the performance support systems and subsystems.

The final layer is used to manage individual access to the appropriate support tools. In a similar way that some learning management systems give each user a unique login page which automatically links to courses they are enrolled in, the PSM will allow a single individualized access to a unique blend of integrated performance support tools. The access is based on the roles that are assigned to an individual within one or more organizational unit. Some access may be voluntary; others may be assigned by those responsible for managing individual performance. The exact details of how such a system should be managed, customized and evaluated is an important area of research. Evaluation is critical in efforts to improve the efficiency and effectiveness of the performance support systems. With a centralized point of access in a PSM, it is possible to create a centralized and automated evaluation system that can feedback to those creating and updating performance support tools. An automated system could monitor patterns of access and use, and automatically generate questionnaires to gather qualitative data when certain patterns are detected. The data from such a system can be matched against performance metrics encoded into analysis objects in the analysis repository; this would enable evaluation of the effect of performance support solutions in improving performance.

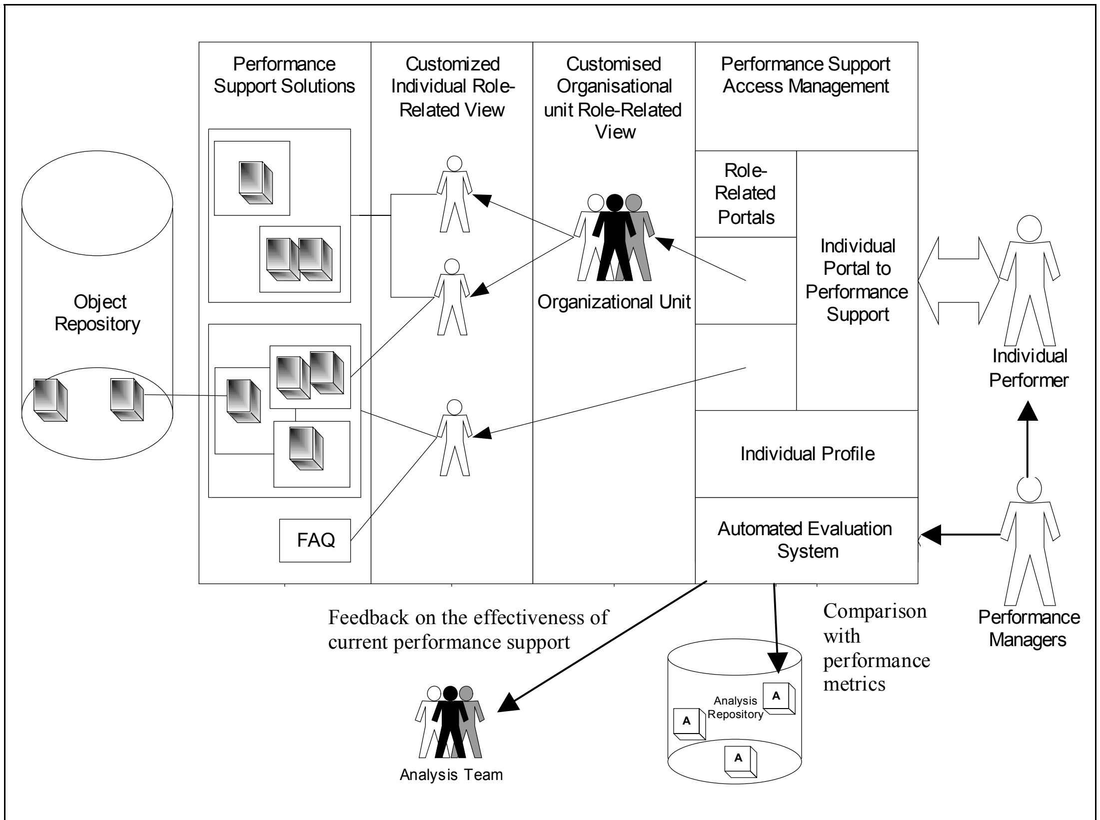

FIGURE 7: PERFORMANCE SUPPORT MANAGEMENT SYSTEM

The role of the PSM is currently taken by learning management systems (LMS), many of these systems are facilitating the technical aspect of learning objects; however, they are still rooted in the thinking that formal courses are the only solution to performance problems. If we develop an OO performance orientation in analysis it makes sense to follow that through to the delivery of a performance support system.

Conclusion

This document has presented an outline of a new methodological framework for the planning, construction and delivery of performance support systems. It envisions the development of support tools that facilitate collaboration and modeling for the planning stage of the framework. It is intended to complement current initiatives towards developing technologies and standards for reusable object-based design and construction. In terms of delivery, the document envisions an evolution of learning management systems towards performance support management systems based on the foundation of performance analysis. Newly developed learning and performance improvement solution packages are accessed, used, and evaluated in a continuous improvement feedback process. Thus, information, knowledge, learning and performance support subsystems are interrelated and accessible based on individual performer requirements.

There is a need for greater consideration, research and collaboration on the issues incorporated within the methodological framework. The current document is presented not as a definitive answer to the need for new methodologies, but as a starting point for debate and refinement of methodological thinking. The Learning Systems Institute at Florida State University in collaboration with the Army Training Information Systems Directorate (ATISD) will work to further develop these ideas.

References

Booch, G. (1999). UML in action. *Communications of the ACM*. 42(10), p. 26-28.

Branson, R., Kaufman, R. and Schaffer, S. (2000). Systems Approach to Training: Proposed process enhancements. *White paper for The Deputy Chief of Staff for Training Headquarters, TRADOC*, September 1.

Brereton, P. and Budgen, D., (2000). Component-based systems: a classification of issues. *IEEE Computer*, 33(11), p. 54-63.

Brown, A.W. (2000). *Large-scale component-based development*. Prentice Hall, NJ. Object and component technology series.

Burge, J. and Brown, D. C. Reasoning with design rationale. In John Gero (ed.) – *Artificial Intelligence in Design '*00, Kluwer Academic Publishers, Dordrecht, The Netherlands, p.611-629.

Clark, R. C. (1999). Four Architectures of Instruction. *Performance Improvement*, 39(10), p.31-38.

Davies, I.K. (1989). Total educational technology (TET): Challenging current limits. *Canadian Journal of Educational Technology*, 18(2), p. 132-137.

Dickelman, G.J. (2000). EDIT 797: Performance-based design [Online], http://pcd-innovations.com/EDIT797/id98_m.htm

Fowler, M. (2000). Put processes on a diet. *Software Development Online*, December. http://www.sdmagazine.com/documents/s=737/sdm0012a/

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

Gery, G. (1991). *Electronic performance support systems: how and why to remake the workplace through the strategic application of technology*. Tolland, Massachusetts: Gery Performance Press.

Gilbert, T. (1998). *Human competence: engineering worthy performance.* Amherst , MA: HRD Press, Inc.

Goodyear, P. (1997). Instructional design environments: methods and tools. In S. Dijkstra, N.Seel, F. Schott and Tennyson, D. (eds) *Instructional Design: International Perspectives*, Vol . 2, Mahwah, NJ: Lawrence Erlbaum.

Gordon, J. and Zemke, R. (2000) The attack on ISD The Attack on ISD. *Training*, April. Retrieved online, November 1, 2001 http://www.trainingsupersite.com/publications/archive/training/2000/004/004cv. htm.

Harless, J (1988). *Accomplishment-Based Curriculum Development.* Newnan, Georgia: Harless Performance Guild.

Kaplan, R.S. and Norton, D.P. (1996). *The Balanced Scorecard: Translating strategy into action*. Boston, MA: Harvard Business School Press.

Kasowitz, A. (1997) Tools for automating instructional design. *ERIC clearing house in Information Technology in Education. Syracuse*, NY. http://www.ed.gov/databases/ERIC_Digests/ed420304.html

Merrill, D. (1998). ID expert: a second generation instructional development system. *Instructional Science* 26(3-4) (July '98), p. 243-62

Merrill, M. David and ID2 Research Team (1996). Instructional Transaction Theory: Instructional Design based on Knowledge Objects. *Educational Technology,* 36(3), p. 30-37.

Moller, L., Benscoter, B., and Rohrer-Murphy, L. (2000). Utilizing performance technology to improve evaluative practices of instructional designers. *Performance Improvement*, 13(1), p. 84-95.

Moore, G. A. (1999). Crossing the Chasm: Marketing and Selling Mainstream Customer*s,* Harper-Collins.

Moran, T.P. and Carroll, J.M. (1996). *Design rationale: concepts, techniques, and use*. Mahwah, NJ: Lawrence Erlbaum Associates.

Ostyn, C. (2001). Interoperable adaptive content framework. *Click2Learn Draft Whitepaper*, Retrieved Sept 18, 2001, from the World Wide Web: http://home.click2learn.com/ standardswork/Adaptive_Content_Framework_for_SCORM_draftwp.pdf

Perez, R. and Emery, C. (1995). Designer thinking: how novices and experts think about instructional design. *Performance Improvement Quarterly*, 8(3), p. 80-95.

Raybould, B. (2001). Performance support engineering: building performance-centered web-based systems, information a systems, and knowledge management systems in the 21st century. *Performance Improvement*, 39(6).

Raybould, B. (1995). Performance support engineering: An emerging development methodology for enabling organizational learning. *Performance Improvement* Quarterly, 10(1), p. 71.

Rosenberg, D. and Scott, K. (1999). *Use Case driven object modeling with UML: a practical approach*, Addison Wesley Object Technology Series.

Rossett, A. (1988*). Training Needs Assessment*. Englewood Cliffs, NJ: Educational Technology Publications.

Rossett, A. and Czech, C. (1996). They really wanna, but….the aftermath of professional preparation in performance technology. *Performance Improvement Quarterly,* 8(4) p. 115-32.

Rummler, G. and Brache, A. (1995). *Improving performance: How to manage the white space on the organization chart* (2nd ed.). San Francisco: Jossey-Bass.

Spector, J. M. and Muraida, D.J., (1997). Automating design instruction. In S. Dijkstra, N.Seel, F. Schott and Tennyson, D. (eds) Instructional *Design: International Perspectives*, Vol. 2, Mahwah, NJ: Lawrence Erlbaum.

Sveiby K.E. (1997). *The new organizational wealth: Managing and measuring knowledge-based assets*. Berrett-Koehler: San Francisco.

Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley, ed., *The Instructional Use of Learning Objects: Online Version*. Retrieved May 18, 2001, from the World Wide Web: http://reusability.org/read/chapters/wiley.doc