WEB-BASED COLLABORATIVE ANALYSIS, REUSE AND SHARING OF HUMAN PERFORMANCE KNOWLEDGE

Ian Douglas, Chris Nowicki, Judd Butler
Learning Systems Institute, Florida State University
Scott Schaffer
Educational Technology, Purdue University

ABSTRACT

The increasing complexity of operations in the modern military is creating a training bottleneck. Two important trends that seek to address this situation are the move towards focusing on performance and the move towards facilitating reuse, sharing and re-purposing of digital knowledge assets. Systems and training analysts and designers need to be given assistance in adjusting to these trends. In this paper we describe a model for configurable, web-based, software tools that support performance and reuse oriented analysis. A prototype based on this model incorporates collaboration among stakeholders, visual modeling of human performance, and it facilitates the creation of sharable human performance analysis objects. Analysis objects contain shared understanding of the gaps that exist between current and desired performance for a specific performance goal and form the basis for the specification of solutions that address those gaps. Analysis objects can be stored in database repositories that allow sharing and reuse of human performance analysis information. In addition to preventing the repetition of previous analysis work, this will also help military personnel identify the performance problems that may arise, the solutions that can support the performance, and the existing resources that can aid in the cost-effective development of new solutions.

ABOUT THE AUTHORS

Ian Douglas is an assistant professor of computer science and a faculty member of the interdisciplinary Learning Systems Institute (LSI) at Florida State University.

Scott Schaffer is an assistant professor of educational technology at Purdue University and a former research associate with LSI.

Chris Nowicki is a specialist in computer research at LSI and is the lead software developer on the research team.

Judd Butler is an associate in research at LSI and coordinates the performance improvement specialists on the research team.

WEB-BASED COLLABORATIVE ANALYSIS, REUSE AND SHARING OF HUMAN PERFROMANCE KNOWLEDGE

Ian Douglas, Chris Nowicki, Judd Butler
Learning Systems Institute, Florida State University
Scott Schaffer
Educational Technology, Purdue University

INTRODUCTION

The increasing complexity of operations in the modern military is placing pressure on training resources. One approach to this training bottleneck is to focus more intently on on-the-job performance (Harless, 1988, Mager, R. and Pipe, 1997, Rossett, 1999). In adopting this approach, some organizations have corrected situations in which only 20% of training content related to actual needs on the job. Such discrepancies can only be avoided by methods and tools that facilitate collaboration between the developers of performance technology and the end-users.

Another approach to increasing the return of investment is to make use of new technology, including distance learning and personal digital assistants, to provide just-in-time knowledge. The design of high quality, effective technology-based systems is expensive and time consuming. The Department of Defense is currently addressing this problem through the SCORM specifications (www.adlnet.org), which aim to facilitate reuse and sharing of content.

Focus on on-the-job performance analysis and focus on reuse of content seem to be developing in parallel, yet they are complimentary solutions to the same problem. Both approaches have implications for how analysts and designers of performance support and training must adjust what they do. There has been a tendency to focus too much on the technology needs for construction and delivery of solutions rather than methodological and support requirements for analysis and design preceding construction.

In previous papers Douglas and Schaffer (2002a, 2002b) have discussed a methodological framework for a needs analysis and planning process that integrates both a performance and object perspective. The basic philosophy is that instead of trying to define a single standard process model, the process model should be flexible and outputs should be standardized to facilitate reuse and sharing of analysis and design knowledge. The process model should be configurable to different situations but be focused on actual performance, and reuse should incorporate some form of:

- Visual modeling
- Collaborative development including end-users
- Consider more than one solution
- Rationale management
- Automated support for analysis and design

In this paper we will describe a model for the automated support as demonstrated in a proof-of-concept prototype called the automated object-oriented performance analysis (AOOPA) system. AOOPA is based around a set of configurable, web-based, software tools that supports the collaborative analysis of military performance problems.

The prototype incorporates collaboration and visual modeling of performance, and it facilitates the creation of sharable human performance analysis objects. Analysis objects contain a shared understanding of the gaps that exist between current and desired performance for a specific performance situation and form the basis for the specification of solutions that address those gaps.

By creating reusable analysis objects, stakeholders will begin to automate the problem-solving process. The purpose of this automation is to increase the productivity, efficiency, and creativity of any problem-solving process while minimizing time required. Analysts will be able to share previously collected analysis data and reuse some or all of the information that is available to them (see Appendix). Analysis objects can be stored in database repositories that allow sharing and reuse of human performance analysis information. In addition to preventing the repetition of previous analysis work, this will also help military personnel identify the common performance problems that may arise, the solutions that can support the performance, and the existing resources that can aid in the cost-effective development of new solutions.

A PROTOTYPE ANALYSIS SUPPORT SYSTEM

Before starting a new project in the AOOPA prototype, team members will assign responsibility for two roles: project coordinator and editor. The project coordinator enters profile information to create an analysis team. The editor will use Visual Modeler on the client machine to create a new project, including basic metadata such as:

- Project name
- Project type
- Editor
- Client
- Date
- Language

Creating Visual Models

Visual modeling provides a shared reference and navigation model throughout a project. AOOPA currently uses performance cases an adaptation from UML Use Case notation that is widely used in object-oriented systems analysis. Performance case notation provides a simple, end-user understandable means of defining a problem space.

When the editor has entered project setup information, he or she creates initial performance diagrams (see figure 1) that depict roles and performance goals.

A performance diagram is a graphic that illustrates what performers do on the job and how they interact with other performers to reach performance goals.

A role is a function or position that performs an abstracted set of on-the-job behaviors (e.g., mission commander, radio operator, vehicle inspector). The primary role is the focus of the project. The secondary role is someone who interacts with the primary role to achieve a specific performance goal. The primary role is likely to achieve several performance goals.

Performance goals represent desired performance at an individual level (i.e., performance necessary to achieve a specific organizational goal). Performance goals reflect the distinctive opportunities, capabilities, and challenges facing each performer (Clark & Estes 2002). The analysis team works collaboratively to create and edit the performance diagram. A threaded discussion feature is integrated into the prototype to facilitate collaboration. Alternatively, the analysis team can use a third party groupware application to create a performance diagram that accurately displays the problem space.

The analysis team will use the diagram to develop a

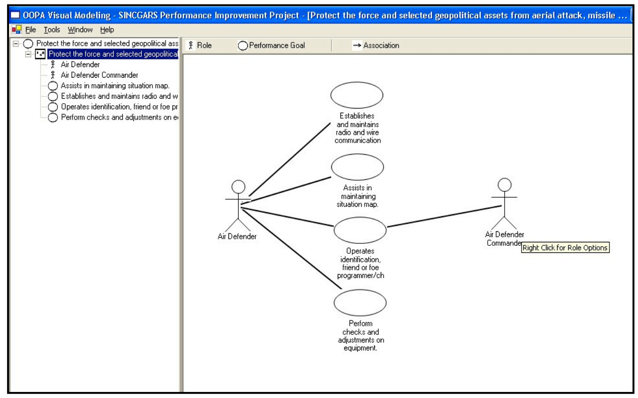

Figure 1: Performance Diagram

shared understanding of a domain and identify performance cases where there is a gap between *desired* on-the-job performance and *current* on-the-job performance. It allows the organization to pinpoint a specific performance discrepancy that could be costing time, money, and other resources.

Gaps – Identify, Rate, and Select

The focus of gap identification (see figure 2) is discovering if there is a gap between *desired* on-the-job performance and *current* on-the-job performance. It allows the organization to pinpoint a specific performance discrepancy or opportunity that could be costing time, money, and other resources.

The *desired* performance consists of the individual performance (as measured in competencies, attributes, cognitive processes and best practices) necessary for the primary role to achieve the specified performance goals.

The *current* state consists of individual performance as

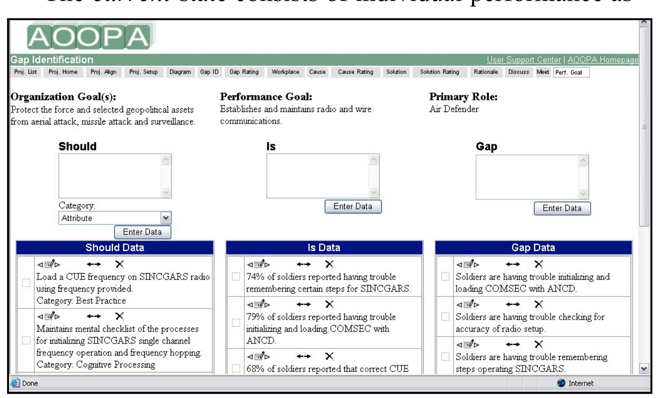

Figure 2: Example of Gap Identification Screen

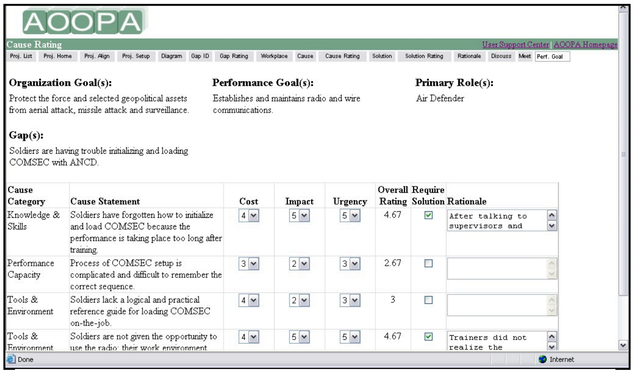

Figure 3: Prioritizing Causes

it actually exists in the present time. The performance gap is the difference between these two states. It represents a performance problem to be solved. The ultimate goal of problem-solving analysis is to close or eliminate this gap in the most cost-effective manner.

An analysis team may choose to focus only on significant gaps in performance in which case a gap rating scheme may be developed.

Cause – Identify, Rate, and Select

A cause analysis involves identifying reasons for deficient performance in an organization, which increases the likelihood that selected solutions will address the causes and close the gaps, because if you know what's causing bad performance or driving successful efforts, you know what you need to do to change or maintain it (Rossett, 1999). Establishing the causal links between current performance and organizational needs makes the analysis credible and useful (Robinson & Robinson, 1995). In a cause analysis, stakeholders review gap data, brainstorm possible causes, put them into cause categories, rate them by user-defined criteria, and select which ones to pursue.

The AOOPA prototype allows users to categorize causes so the recommended solutions are more likely to address the underlying causes. Generally, a cause category is easily linked to a solution category. Using categories is a way to speed up the problem-solving process. Causes typically fall under one of six categories: Knowledge/Skills, Motivation/Self-concept, Performance Capacity, Expectations/ Feedback, Tools /Environment, or Rewards/Incentives (Wedman and Graham, 1998). Rating cause statements allows users to decide which causes are most problematic and should be dealt with first.

The team uses ratings to prioritize causes (see figure 3) and chooses only the ones most likely to yield results; searching for solutions to all possible causes would be costly and an inefficient use of time. Typically causes are selected based on the highest average; the higher the average, the more likely that addressing the cause will improve performance.

Solution – Identify, Rate, and Select

Solution selection is the process of identifying a blend of alternative solutions that will reduce gaps in a cost effective way. By identifying a number of potential solutions it is possible and appropriate to create a blend of short-term and long-term solutions. Data from gap identification and cause analysis lead users to identify the most effective solutions. Causes and cause categories point to solution categories. Each solution category suggests a type of solution, which helps analysis team members and other project stakeholders to brainstorm possible solutions that are appropriate to a gap and its causes. Solutions generally fall into six categories: Feedback Systems, Job Aids, Training, Organization Redesign, Selection/Recruiting or Reward/Recognition. This phase of work is an opportunity for significant collaboration; stakeholders can write and revise solution statements until they reach consensus.

Solution rating and selection is fueled by stakeholder participation. Rating solution statements enables stakeholders to discuss and decide which solutions are more likely to reduce or eliminate a performance gap. Specific solution criteria can be added to each project (e.g. cost effectiveness, buy-in potential and impact). Solutions blends that have the highest potential for implementation and the likelihood of having the greatest impact are usually the best choice for the organization.

Object Matching – Analysis Objects and Solution Objects

Solution objects are reusable units of information (e.g., image, text, video) that reside in repositories and can be retrieved to construct performance improvement solutions. These discrete pieces can be reused in a number of solutions. They may be entirely electronic (e.g., online training modules), descriptions of non-electronic content, or a combination of both. The Object Manager allows users to view and manage solution objects. Users find objects by searching for them in one or more object repositories using either a manual or an automated search. Object Manager enables both types of searches (see figure 4). The manual search is similar to a typical web-based search utility. The automated search, by contrast, builds a search string from the analysis data in the AOOPA tools and uses this data to locate matches, which are then added to a potential solution list automatically.

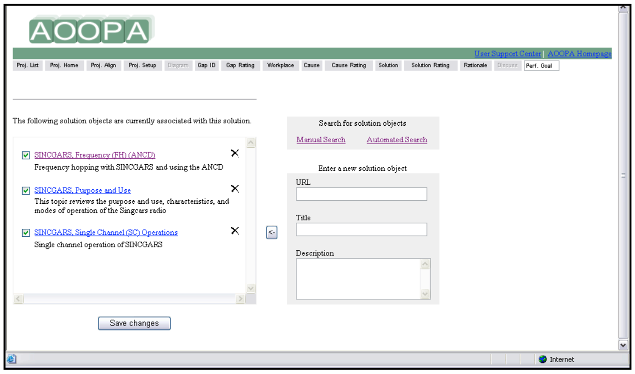

Figure 4: Searching for Solutions

Rationale Management

Rationale management involves recording the reasons behind decisions made in a project. It clarifies for the analysis team what decisions it has made; it also helps others reuse portions of problem solving since they are able to understand why the team made its original choices. Rationale would include anything that would help others understand why decisions were made. That could be quantitative (e.g., hard data, rankings) or qualitative (e.g., textual explanations). It is important to record rationale completely and accurately

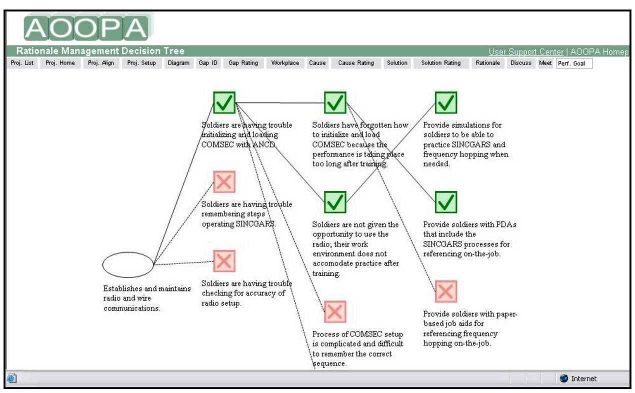

Figure 5: Rationale Diagram

regardless of who enters it or how it is entered. One method would be for stakeholders to reach consensus and then agree on who will be responsible for recording group decisions. If when team members are separated by time or geography, rationale may be a collection of individuals' reasons than a statement of consensus.

The Rationale Diagram (see figure 5) is a graphical depiction of choices made over the course of a project about gaps, causes, and solutions. The prototype application uses selections from rating screens to construct the diagram. Clicking on a box in the diagram takes users back to the screen where rationale was recorded so they can see why certain gaps, causes, or solutions were pursued while others were discarded.

THE SOFTWARE ARCHITECTURE

The software architecture for the AOOPA system is both web and component-based. It is built upon having completely separate components for modeling, collaboration, user support, and data storage (see figure 6). The design strategy for the prototype is a standard three tier architecture (Eckerson, 1995). Three-tier architecture offers performance benefits, flexibility, scalability, and reusability. These characteristics make three-layer architectures a popular choice for net-centric information systems.

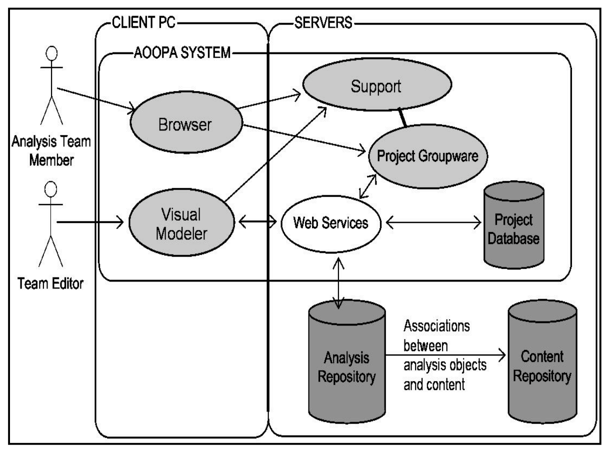

Figure 6: AOOPA Software Architecture

The top layer provides the user interfaces illustrated in the previous figures. It contains the visual elements users see as well as the presentation logic needed to present the information. Having the code that creates interfaces separate from the code that processes the inputs facilitates the system being reconfigured for a different system of modeling, collaborating, or data entry.

The middle tier provides the business logic and processes that are shared by multiple interfaces. It is the middle tier that enables the performance benefits, flexibility, scalability, and reusability by separating the business and process logic into discrete components or services. Separated process logic in a component form makes implementation and maintenance easier because new logic and changes must only be written once and placed on the middle tier to be available throughout the system. With other design strategies, a change to a function would have to be written into every application.

The prototype uses web services to implement the middle tier. Web services are an emerging standard for distributed software components (Daconta et al, 2003). They are self-describing, modular applications that can be published, located, and run across the Web. Web services perform functions which vary from simple requests to complex business logic. Other middleware implementations exist, but none has emerged as the true winner. The power of the Web as an information distributor is a key to solving the interoperability issues inherent in other existing middleware.

The third tier provides database management. The database management component ensures that the data is consistent throughout the distributed environment using techniques including locking, consistency checks, and replication.

The current system as described above presents a generic problem-solving methodology to the users. In the future, the team plans to create additional interfaces that implement specific methodologies or allow access through other hardware interfaces like palm or tablet computers. The component architecture allows users to slot these additional interfaces into or out of the system. The eventual aim is for different organizations or teams to have custom-designed interfaces using familiar collaborative tools and terminology. The AOOPA system is therefore not a model for a single tool; it is a model for a set of tools that can be configured to different processes or purposes (see figure 7).

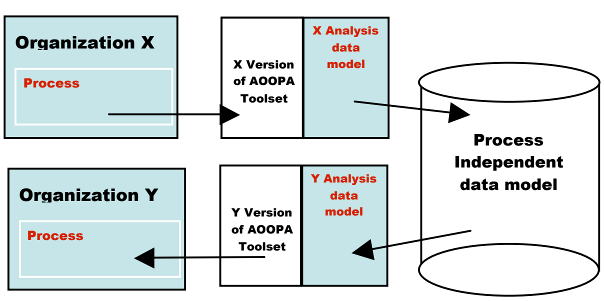

Figure 7: Configurable Process Supported by Configurable Tools

In order to facilitate reuse, the team envisions that a standard, generic data model will emerge for human performance analysis. The custom entries will be translated into standard generic forms of performance analysis knowledge, which will allow interchange of knowledge between different teams.

ACKNOWLEDGEMENTS

This work was carried out in collaboration with the Army Training Information Systems Directorate (ATISD) part of the Army Training Support Center (ATSC).

REFERENCES

Clark, R. E. and Estes, F. (2002). *Turning Research into Results.* Atlanta, GA: CEP Press.

Daconta, M. C., Obrst, L. J., Smith, K. T. (2003). The Semantic Web : A Guide to the Future of XML, Web Services, and Knowledge Management

Douglas, I. and Schaffer, S. (2002a). Object-oriented performance improvement. *Performance Improvement Quarterly*, 15 (3), pp 81-93

Douglas, I. and Schaffer, S. (2002b) "Object-oriented performance analysis". *Proceedings of the Inter-service/Industry Training, Simulation and Education Conference* (I/ITSEC), p.367-377.

Eckerson, W. (1995) "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." *Open Information Systems 10,* 1 (January 1995): 3(20).

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

Marshall, C. (2000). *Enterprise Modeling with UML: Designing Successful Software through Business Analysis*. Reading, MA: Addison-Wesley.

Mager, R. and Pipe, P. (1997). *Analyzing Performance Problems*. Atlanta, GA: The Center for Effective Performance, Inc.

Robinson, D. and Robinson J.C., (1995). *Performance Consulting: Moving Beyond Training*. San Francisco: Berrett-Koehler.

Rossett, A. (1999). *First Things Fast: A Handbook for Performance Analysis*. San Francisco: Jossey-Bass Pfeiffer.

Wedman, J. & Graham, S. (1998). The Performance Pyramid. The Journal of Continuing Higher Education, Fall, 46 (3).

Appendix: Performance Analysis Project Data

Visual Model Data	
Primary Role	The performance of this role is the focus of the analysis.
Secondary Role(s)	Other roles that interact with the primary role to achieve a performance goal.
Performance Goal	Desired achievement, results, or output at an individual level.
Role Data	Information about a role such as competencies, job title, or position number.
Gap Data	
Desired Performance	Information about what the individual should be achieving on the job.
Current Performance	The level at which an individual is currently performing.
Gap Statement	The difference between "should" and "is" data (i.e., desired and current performance).
Criteria for Closing Gap	Indicator that gap has been closed or sufficiently reduced.
Gap Criterion	Criteria used in the decision to select, reject, or prioritize a gap.
Gap Matrix	A table for rating potential gaps using standard and/or user-entered criteria.
Gap Status	Indicates whether the gap was recommended for further analysis or not.
Gap Rationale	The rationale behind a decision to select, reject, or prioritize a specific gap.
Cause Data	
Cause Statement	Statement of proposed cause.
Cause Criterion	Criteria used in the decision to select, reject, or prioritize a cause.
Cause Matrix	A table for rating potential causes using standard and/or user-entered criteria.
Cause Status	Indicates whether the cause was recommended for further analysis or not.
Cause Category	Indicates the general cause type for each cause statement.
Cause Rationale	Indications the rationale behind a decision to select, reject, or prioritize a specific cause.
Solution Data	
Solution Statement	Statement of a proposed solution.
Solution Criterion	Criteria used in the decision to select, reject, or prioritize a solution.
Solution Matrix	A table for rating potential solutions using standard and/or user-entered criteria.
Solution Status	Indicates whether the solution was recommended for implementation or not.
Solution Category	Indicates the general solution type for each solution statement.
Solution Rationale	Indicates the rationale behind a decision to select, reject, or prioritize a solution.
Solution Search	Data, specific to each solution, used to search for solution objects.