

Learning by Demonstration to Support Military Planning and Decision Making

**Thomas Garvey, Melinda Gervasio,
Thomas Lee, Karen Myers**

Artificial Intelligence Center, SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
{thomas.garvey, melinda.gervasio,
thomas.lee, karen.myers}@sri.com

**Carl Angiolillo, Matthew Gaston,
Janette Knittel, Jake Kolojechick**

General Dynamics C4 Systems | Viz¹
Suite 310, 2730 Sidney Street
Pittsburgh, PA 15203
{carl.angiolillo, matt.gaston,
janette.knittel, john.kolojechick}@gdc4s.com

Abstract

While the concept of learning by demonstration has been around for many years, recent advances in artificial intelligence technology have led to a resurgence of work in the field. We describe the development and application of learning by demonstration technology to support user creation of automated procedures for a rich collaborative planning environment that is in widespread use by the U.S. Army. User feedback and evaluation results show that the technology can be used effectively by the target user community and that it has tremendous potential for improving the speed and quality of performance for a range of critical tasks.

Introduction

In contrast to much of learning research to date aimed at understanding information spaces, a major thrust of the DARPA Personalized Assistant that Learns (PAL) program has been to develop methods to enable a machine to learn to assist users by accomplishing tasks on their behalf. Effective, usable *task learning* (also known as *procedure learning*) presents a range of challenges in interpreting user intent, inferring appropriate generalizations for learned procedures, and user accessibility. In this paper we describe our approach to solving these problems in the context of learning and executing procedures to accomplish user tasks in the complex, dynamic environment of military command and control (C2).

Our target application for task learning is a C2 system in wide use across the U.S. Army, namely the DARPA-developed Command Post of the Future (CPOF). CPOF is a multiuser visualization environment that allows users to collaborate at an intuitive level, sharing impressions and understanding of situations and tactics. Much of the power of CPOF comes from its generality and composability, providing tremendous flexibility to let users handle a wide

range of missions. The flip side of this flexibility, however, is that CPOF provides few built-in processes to facilitate user activities. As a result, CPOF can require significant user interaction for tasks (i.e., it is “click intensive”), leading to high user workload.

Task learning technology seemed an ideal solution. By learning to accomplish a range of tasks for the user, it could greatly reduce workload and stress, enable the user to manage more tasks with better effectiveness, and facilitate consideration of more options resulting in better decisions. As one Army officer put it, the technology could “provide time to *think*.”

Our team focused on a *learning by demonstration* paradigm for task learning for several reasons. First, the relative simplicity and reliability of the process—requiring only that the user indicates the start and end of a teaching session while in between performing the actions associated with the task as he normally would—made it more likely to be adopted by CPOF users. Second, because the users themselves teach the procedures, they gain better understanding and are more likely to trust the learned procedures. Finally, as a user-initiated approach, it provides more precise control of what is learned; in potentially life critical situations, it is important that the system learn from experts.

Building on PAL learning by demonstration technology, we developed a PAL-enhanced CPOF system (referred to as PAL-CPOF) to support user creation of automated procedures for tasks. More specifically, PAL-CPOF enables individual users and collective command staffs to create customized, automated information management schemes tailored to individual desires and the staff’s standard operating procedures, without the need to enlist software engineers for extensive recoding. In a significant evaluation conducted by the Army in December 2008, users overwhelmingly endorsed the capabilities provided by task learning. Users stressed that the PAL technology “saves time and that time equals lives.”

¹ Viz is a business area within General Dynamics C4 Systems.

This paper focuses on describing the PAL-CPOF prototype, its use for automating a range of C2 tasks, and the results of the Army evaluation of its effectiveness. We begin with an overview of the CPOF collaboration environment followed by a discussion of the process used to develop our approach for integrating procedure learning capabilities into CPOF. Next, we describe the PAL procedure learning capabilities, their integration into CPOF, the resulting PAL-CPOF system, and its use. We then present results from the Army evaluation of the technology. We close by outlining directions for future work and summarizing the main results.

Command Post of the Future

CPOF is a state-of-the-art visualization and collaboration system for command and control in the warfighting environment (see Figure 1). CPOF grew out of a need to enable command posts to effectively accommodate and process greater amounts of information through distributed, collaborative operations. Development of CPOF originated in a DARPA program focused on advanced user interface design for C2 environments. CPOF is built on the CoMotion[®] platform, which was derived from earlier user interface research on SAGE (Roth et al., 1994) and Visage (Roth et al., 1996). CoMotion[®] provides rich capabilities for collaborative visualization and information analysis. Key design concepts in CoMotion[®] include *infocentricity*, *direct manipulation*, and *deep collaboration*.

CPOF is highly *info-centric* in that its focus is on data rather than tools. In the command and control environment the ability to dynamically incorporate new information is critical to the success of an operation. While most previous visualization efforts have focused on the analysis of static data sets, CPOF visualizations operate on *live* data, continually updating themselves in response to changes in the data. Users can author new information directly in the visualizations and can drag and drop information analyses across visualization products.

The primary interaction paradigm in CPOF is one of *direct manipulation*. CPOF makes heavy use of drag-and-drop and other simple gestures, and it employs a relatively small set of visualization techniques specifically tailored to accommodate information in the command and control domain. By employing a small set of rules with great consistency, CPOF ensures simplicity and predictability in the interaction. Thus, in contrast to more complex user interfaces, it is easy for users to learn to use CPOF once they understand a few basic concepts.

CPOF supports *deep collaboration* through real-time information sharing that is always on and occurs as a natural “side effect” of user activities. By letting users tailor their visualizations to capture the way they process information, CPOF provides visibility into the evolving understanding among distributed team members and supports the collaborative creation of work products.

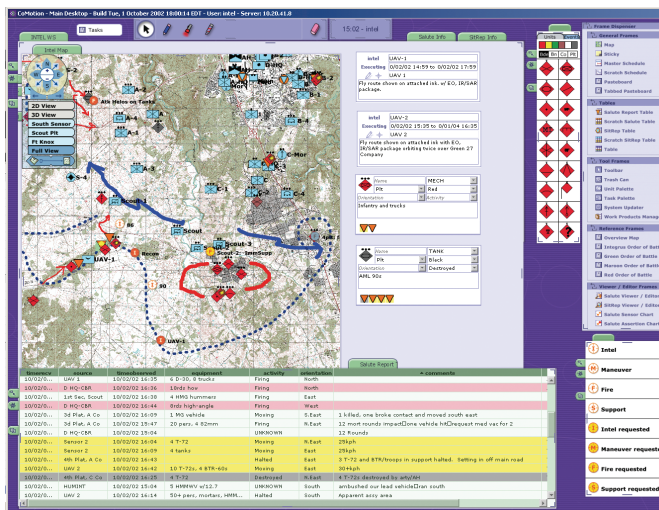


Figure 1. CPOF Display

CPOF has been used to create a range of C2 artifacts including information briefings, storyboards, common operating pictures, resource-level estimates, and fragmentary orders. Since its inception in 2004, thousands of CPOF systems have been deployed around the world. CPOF clients running on Windows desktops were targeted for transition of the PAL task learning technologies.

Path to Technology Transition

Our early observations of Army command staff showed that events such as the explosion of a roadside bomb, an attack on a convoy, or a mortar attack required the user to arrange a *battle board* that collected relevant maps, documents and other data in a presentation that provided ready access to critical information. The event also required the user to initiate and track processes such as medical evacuation, unit reinforcement, and notifications to higher headquarters; each of these activities might entail further tasks and processes. Although experienced users knew what they had to do on these occasions, it still required precious time to set up everything, make the proper notifications, and track the results. Even worse, critical steps were often overlooked by tired and stressed crews in the heat of the situation.

These activities are characteristic of the repetitive nature of much user activity in CPOF, and cry out for an automated solution. Meanwhile, the highly flexible and customizable nature of CPOF calls for an end-user programming solution to let users create their own processes. The PAL task learning technology provides just such a capability and thus was born the idea of enhancing CPOF with task learning.

To further develop and refine the procedure learning concept for CPOF, we initiated a series of user-centered exercises with Army subject matter experts (SMEs). These

exercises were conducted using the ‘Double Helix’ methodology that was applied successfully for the initial deployment of the CPOF system. The name derives from the simultaneous refinement of the technology based on user interactions and feedback while developing the Concept of Operations for its effective use.

Our Double Helix sessions allowed users to teach their systems procedures, evaluate them under stressful, simulated combat conditions, evolve the most effective concepts of operation for their application, and provide feedback to the developers regarding problems and issues with the software as well as requests for additional features. Because these exercises involved the users in the development cycle, they further enhanced user acceptance—a critical issue for technology transition.

Task Learning in PAL-CPOF

DARPA’s Personalized Assistant that Learns (PAL) program is a large-scale, multiyear effort to develop technology to realize adaptive intelligent cognitive assistance in the office environment (Myers et al., 2007). In line with the primary PAL theme of *learning in the wild*, a major research thrust in PAL is *task learning*: the ability for a system to autonomously acquire new procedures for achieving tasks. The *Integrated Task Learning* (ITL) framework in PAL brought together a range of learning technologies—including learning by demonstration—and was specifically designed to support learning over one or more application platforms (Spaulding et al., 2009). Here we describe our experience in transitioning the PAL learning by demonstration technology into CPOF.

Learning Procedures from Demonstration

The concept of end-user programming—and, in particular, programming by demonstration (Cypher, 1993; Lieberman, 2001)—has been around for many years, but recent advances in artificial intelligence (AI) technology coupled with increasingly sophisticated end users has brought a resurgence of work in the field (e.g., Allen et al., 2007; Burstein et al., 2008; Eker et al., 2009; Gervasio & Murdock, 2009; Leshed et al., 2008).

In PAL, learning by demonstration is supported by LAPDOG (Gervasio et al., 2008; Eker et al., 2009). Given a demonstration consisting of a sequence of actions performed by a user to achieve a particular task, LAPDOG generalizes the action sequence into an executable procedure for achieving similar tasks in the future. LAPDOG subscribes to a *dataflow* model of actions, where actions are characterized by their inputs and outputs and action outputs serve as inputs to succeeding actions. Thus, procedure induction involves ensuring that every action input is supported by a previous output—possibly through the insertion of information-producing actions (Gervasio & Murdock, 2009).

LAPDOG performs *parameter generalization*, generating a replacement of constants with variables or

expressions over variables (e.g., list or record accessors, list constructors) that reflect the dataflow observed in the demonstration. This generalization enables procedures to be reused in different situations from which they were taught and to different effect. Within the CPOF environment, for example, users found it helpful to create procedures that could display different classes of events on a map by teaching a procedure for a single class.

Finally, LAPDOG performs *structure generalization*, inducing loops over collections of objects by identifying and generalizing repeated sequences of actions in the demonstration. LAPDOG’s loop induction algorithm supports advanced features such as the ability to learn loops over both ordered and unordered collections of objects (lists and sets) and loops that process multiple lists simultaneously (Eker et al., 2009).

Parameter and structure generalization are performed synchronously, so LAPDOG can learn loops over arbitrarily nested lists or lists of records, and loops over collections that are the output of previous loops. While LAPDOG was not specifically designed to learn from a single example, it is capable of doing so through the use of heuristics for filtering the set of alternative generalization hypotheses to a single best generalization.

LAPDOG learns procedures that can be executed within the SPARK agent framework (Morley & Myers, 2004). SPARK is a procedural reasoning framework designed to support the sophisticated control mechanisms required by practical systems while maintaining the rich reasoning provided by a formal representation language. By virtue of SPARK’s hierarchical representation of tasks, LAPDOG readily supports task composition by demonstration. Procedures invoked as part of a demonstration for achieving a new task effectively become subprocedures in the new learned procedure.

Learning from demonstration was a natural match to CPOF for a number of reasons. First, it fulfilled the need for a means by which tedious processes in CPOF could be automated. Second, the paradigm of learning from demonstration was ecologically compatible with the CPOF philosophy of providing users maximum flexibility in the use of tools. Third, LAPDOG’s ability to learn from a single example made it a particularly attractive approach for CPOF, where tasks were often lengthy and time-consuming for a user to demonstrate.

Developing an Action Model

LAPDOG is a generic, application-independent task learning framework. The use of LAPDOG within a particular application requires the definition of an *action model* that provides a semantic specification of individual actions in the application along with corresponding mechanisms for instrumenting and automating those actions. In essence, the action model provides the basic building blocks from which learned procedures are created.

An important lesson learned in the CPOF deployment was the necessity of modeling actions at the right level of

abstraction. The final action model used in the CPOF deployment was the result of multiple iterations between AI researchers and CPOF developers to identify a level of abstraction that both provides a user-friendly representation of procedures and is practical from a systems engineering perspective.

Our first attempt at an action model captured events at the level of primitive data changes—for example, creating an entity, changing the value of an entity attribute, and moving a contained entity to the front/back of its parent container. This had two distinct advantages. First, instrumentation was straightforward as this was the level at which CPOF naturally operated. Second, it allowed for a compact action model, needing only twenty-eight actions to capture all user activities on CPOF.

While this level of modeling was desirable from the application perspective, it was a poor vehicle for capturing user intent. As an extreme analogy, consider instrumenting an email client at the level of mouse and keyboard clicks. While the resulting execution trace will capture all the user's actions, it will not allow meaningful distinctions to be made between actions such as the user clicking on a message vs. clicking on the *Write* button, or the user typing an email address in the *To* field vs. typing a message in the *Body* field. Furthermore, many of the action parameters will not lend themselves well to generalization—for example, how does one generalize the *x* and *y* coordinates for a mouse click or the characters being typed in a window?

Another significant consequence of the low-level action model was that the resulting procedures were sometimes very lengthy and were often incomprehensible to the end users. An action that is conceptually a single action from the user's perspective typically generates several lower-level instrumentation events that typically do not represent what the user believes to have just done. Thus, for the most part, this earlier version of PAL-CPOF learned ungeneralized macros, which were somewhat useful but left little opportunity to capitalize on LAPDOG's generalization capabilities and were unsuitable for any subsequent procedure visualization and editing.

Our current action model represents actions at a higher level, in terms of their effects on the user-facing objects in CPOF. This action model includes elements such as dispensing a frame, adding an object to a collection, centering a map on an entity, and setting the title of an object. This was an intuitively better approach as it depicted actions at a level that was much more understandable to humans. However, it presented the challenge of translating the readily available low-level instrumentation events into the higher-level actions more suitable for representing a demonstration. For example, conceptually distinct actions sometimes had similar manifestations, while a sequence of events sometimes corresponded to a single action and sometimes to a sequence of interdependent actions. From a user's perspective, performing what amounted to the same gesture sometimes mapped to different actions depending on their execution context. Thus, significant effort was

spent on designing and implementing this translation layer in order to provide LAPDOG with an action sequence that closely corresponded to the user's conception of the demonstration.

The resulting action model is also much larger and, currently at seventy-six actions, it covers the great majority of but not all conceivable user activities on CPOF. However, this action model results in significantly more meaningful procedures that include distinct dataflows that can be captured and generalized through LAPDOG. It also opens up the opportunity for end-user editing of learned procedures. Through the careful development of an action model at the right level of abstraction, we are thus better able to leverage the power of task learning.

Triggers

Learning by demonstration greatly enhances the CPOF experience: with the ability to automate tedious, repetitive procedures, users are freed to focus on more cognitively demanding tasks. Based on user feedback from Double Helix exercises, we took the additional step of providing *triggers* to enable automatic invocation of learned procedures as a way to enhance the utility of procedure learning.

We focused on a small set of trigger types that were likely to provide the most value to users. *Time triggers* invoke procedures at absolute or relative times and/or with a specific frequency. *Data triggers* invoke procedures based on particular changes to properties of various objects in the system. *Area triggers* invoke procedures based on particular changes to predefined areas on a map.

Triggers and procedures together proved to be a powerful combination that provided users with the ability not only to monitor state changes but also to respond to them automatically. This paved the way for the technology to provide even more assistance in automating user workflows. For example, triggers enable users to teach their systems to respond to events automatically by arranging their battle boards, notifying selected staff, and preparing reports to inform higher headquarters for the user's approval and dissemination. Additional examples are provided later in the Use Cases section.

User Interface

To attain user acceptance of the technology, it was important that the interface to task learning be consistent with the interaction paradigm CPOF users have come to expect. PAL-CPOF provides a *PAL appliance* for coordinating procedure learning and execution (see Figure 2). Aside from a button to start/pause/end a demonstration and a button to invoke learning, the interface provides a rudimentary capability for users to organize the action sequence into a series of higher-level steps. With the original, low-level action model, this facility was crucial for users to be able to structure demonstrations that were often hundreds of actions long. While less critical for the shorter, more meaningful procedures now learned with the

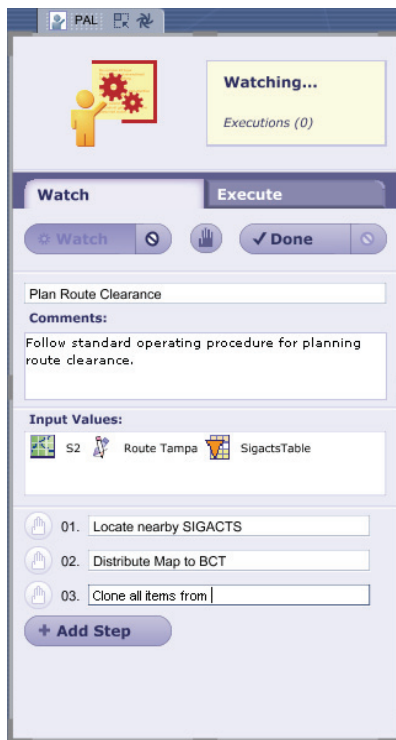


Figure 2. PAL Appliance within the CPOF UI

higher-level action model, other features that were developed as part of this facility remain useful—for example, the ability to step through a procedure or to set breakpoints.

In line with the data-oriented perspective of CPOF, the execution interface displays all the learned procedures, which can be pulled out individually for limited parameter editing. Although action parameters are variablized during learning, their demonstrated values are retained and used as defaults when the procedure is executed. These values can be changed manually by the user before execution or automatically by a trigger. By virtue of CPOF operating on live data, even a procedure that always uses the default values or lacks parameters can execute differently each time it is invoked.

Triggers are defined through the use of a *trigger palette*, which provides specialized templates for the different types of triggers. Each template provides an area into which a user can drag the procedure to be executed. Since different triggers may be useful in different situations, users can easily control which triggers are active at any given time. An *Activity Manager* tracks the execution of triggered procedures and maintains a history of trigger firings.

PAL-CPOF Architecture

Figure 3 depicts the architecture of the PAL-CPOF system. In addition to the LAPDOG learning by demonstration system and the SPARK Task Executor, a procedure editing module (described in the Future Directions section below)

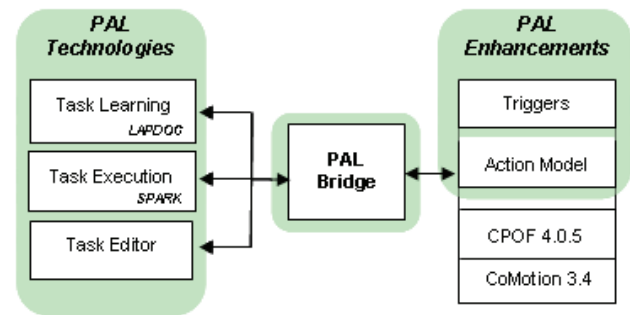


Figure 3. PAL-CPOF Architecture

provides users with the means to visualize and modify learned procedures. A clean separation is maintained between the PAL technologies and CPOF by means of an explicit bridge module, through which all communication flows (i.e., instrumentation events and automation requests). By having a single, unified interface between PAL and CPOF, we standardize the integration of PAL technologies into CPOF and simplify future integration of other PAL systems.

Use Cases

Army personnel and SMEs have produced a large number of use cases for PAL procedure learning. In several cases, they applied the procedure learning enthusiastically to address personal frustrations with CPOF usage that they had been harboring for years. Here, we describe briefly two such use cases.

“While You Were Away.” A major issue with a live collaborative system is that it requires constant attention; looking away for a minute may mean missing critical information. Learned procedures can alleviate this stressful situation by tracking and aggregating changes that occur while the user is not actively watching, allowing him or her to focus on other activities. For example, the commander can use a collection of learned procedures to capture activity (e.g., shared annotations and markup) and information from the workspaces of multiple staff members while away. The learned procedures can organize the collective information in a way tailored to the commander’s style and preference, thus enabling rapid realization of situational awareness.

Tracking Route Clearance Progress. Route clearance is a process that involves sequentially clearing adjacent areas along a designated route between start and destination locations. Route clearance requires a high degree of consistency to track which areas have been deemed clear and when. Procedures have been taught (a) to track and display the progress as the clearance is under way and (b) to monitor and track the progress of units, with instantaneous updates regarding potential threats in their areas of operation. This latter capability provides

overwatch of vehicles on the route to ensure that once cleared of threats, they remain secure.

Other cataloged use cases include

- Preparing the Commander's daily briefing
- Archiving old data on significant activities
- Tracking hypothetical scenario data
- Preparing initial storyboards
- Distributing completed storyboards
- Facilitating information transfer across shift changes
- Notification of important events
- Modifying system privileges

We have identified two main advantages to using a procedure learning system rather than custom prebuilt components to meet these use cases. First, individual users have been able to adapt the software to suit their specific needs. Second, the solutions can be built quickly by experts in the field, rather than by programmers thousands of miles away who must rely on second-hand accounts of requirements and problems.

Evaluation

Initial User Response

In the two years spent developing the PAL-CPOF prototype, numerous user evaluation sessions were held to obtain feedback on the utility and usability of the technology. These sessions generated tremendous enthusiasm from a number of active duty personnel, with enthusiasm levels somewhat correlated with their CPOF experience. One staff officer remarked that the procedure learning technology would enable him to “eliminate 90% of my job.” An Army Captain from Ft. Stewart went on record stating that the technology “has the potential to save countless man-hours by conducting routine, repetitive tasks with little or no input from the user. Those man-hours could then be reallocated to other tasks or even free up soldiers to conduct combat operations.”

BCBL Evaluation

Because of this enthusiastic initial response, the decision was made to proceed with a more formal evaluation to assess the potential value of the procedure learning technology to the Army. This evaluation was conducted by the Army Battle Command Battle Laboratory (BCBL) at Ft. Leavenworth during December 1-5, 2008, as part of a larger effort to assess the usability and effectiveness of a range of PAL technologies.

The purpose of the evaluation was to test the hypotheses that a Brigade Combat Team (BCT) commander and staff using PAL-enhanced versions of CPOF and a companion tool for intelligence analysis—compared to nonenhanced versions of these systems—will achieve better situational

awareness and situational understanding, complete more tasks within a given amount of time, and produce higher-quality products within a given amount of time.

The evaluation compared the performance of two teams composed of representatives from each of the core functions of a BCT (Ops, Intel, Sustainment, Effects, Battle Captain, and Commander), operating in a stressful environment of escalating, ad hoc, time-sensitive tasks related to stability operations.² The teams were composed of active duty and retired Army personnel, all with significant relevant military experience. Team members received several weeks of training on CPOF and PAL-CPOF capabilities.

For the first two days of the evaluation, one team used the PAL-enhanced systems while the other used the systems without PAL to conduct an air assault mission. The teams then switched for the final two days, starting afresh on a scenario involving provision of security for a religious pilgrimage. Two groups of scenario controllers played the adversary, fed the teams events, responded to their actions, simulated subordinate commands (battalions and companies), and simulated orders and requests from higher command. Extensive scenarios and contingencies were scripted to support different responses resulting in varying trajectories for the two teams.

Performance evaluation drew on a combination of subjective assessments by observer analysts of BCT operations, blind review of data products (involving both subjective measures of quality and quantitative measures related to degree and time of completion), and self-assessments by BCT role players in After Action Reviews and via questionnaires.

The final report on the evaluation from the BCBL concluded that the PAL-enhanced systems “clearly ... enhanced the BCT staff’s ability to conduct military operations” (Sanders et al., 2009). The evaluation results showed that each of the experimental hypotheses was validated: the PAL versions of the technologies increased situational awareness and understanding while enabling more tasks to be completed and with higher-quality data products in the same amount of time. Because the assessment focused on the combined effects of procedure learning for CPOF with PAL enhancements to the intelligence analysis tool, assessing the impact of PAL-CPOF in isolation is difficult. Nevertheless, it is possible to identify specific contributions of the procedure learning capabilities to these overall results.

To that point, one requirement for the BCTs was to produce storyboards that described planned BCT response to injected events. Data collected during the evaluation showed that the PAL procedure learning technology

² From the U.S. Joint Staff’s Joint Publication 3-0, Joint Operations: stability operations encompass “various military missions, tasks, and activities conducted outside the United States in coordination with other instruments of national power to maintain or reestablish a safe and secure environment, provide essential governmental services, emergency infrastructure reconstruction, and humanitarian relief.”

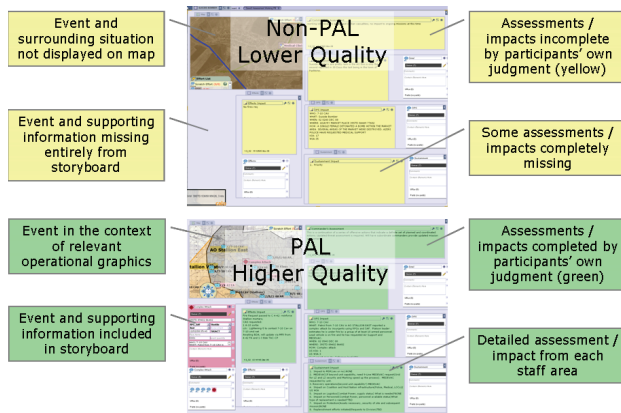


Figure 4. Sample storyboards from the evaluation

enabled teams to recognize significant events four times faster and respond to them more consistently than their counterparts without the technology. Non-PAL teams were unable to keep up with the deluge of activities and entirely missed responding to some critical events, leading to serious downstream consequences. Furthermore, storyboards produced using PAL task learning were of much higher quality, containing much more contextually relevant supporting information and more detailed assessments (see Figure 4). More generally, as noted in (Sanders et al., 2009):

“The military decision making process products ...were deemed to have a higher quality while produced in less time. Furthermore, directions given to subordinate units were more complete and contained more detail than the orders produced without PAL.”

Feedback from the observer analysts and participants further confirmed the value of the procedure learning technology within CPOF:

“When enabled with PAL-CPOF, the BCTs were able to handle routine actions, and had the time to focus their attention on the issues that met the criteria for the commander and staff’s attention. They had more time to think, plan and make sound decisions.”

After the evaluation, the procedure learning technology was demonstrated to the Vice Chief of Staff of the Army, who played an integral role in the initial transitioning of CPOF from research prototype to Army system of record. His enthusiastic response has resulted in exploratory efforts to investigate next steps in transitioning the procedure learning technology into the deployed CPOF system.

Future Directions

Our future work will focus on transitioning the PAL-CPOF procedure learning framework to the Army. Part of this work will involve the typical hardening and validation requirements associated with fielding technology. Beyond that, we will also address feedback obtained from the evaluation participants for improving the usability of the task learning technology to increase its value to the Army. Our focus will be on the following three areas.

Procedure Editing. The PAL-CPOF system used in the evaluation provided no visualization of procedures and no ability to modify them. Participants identified an intuitive editing capability as a key requirement for fielding the procedure learning technology. Companion work has produced an initial stand-alone procedure editor that enables users to view and modify learned procedures (Spaulding et al., 2009). Supported modifications include deleting or adding steps, changing parameters, incorporating fragments of steps from other procedures (with associated reasoning to link the pasted element into its new context), introducing conditions and loops, and composing multiple procedures together. This initial editing capability was developed with significant input from SMEs but has not been evaluated by users. In future work, we will seek to validate and refine the editor to meet user needs, and to integrate it more tightly with CPOF to provide a seamless user interface (UI) experience.

Monitoring and Analysis of Triggers and Procedures. Triggers and procedures together provide a PAL-CPOF user with a tremendous amount of power. However, because of the distributed nature of the resulting process (e.g., triggers can initiate procedures that activate triggers that initiate procedures, and so on), tracking the effects for a large body of triggers and procedures presents challenges for the user. In the extreme, interactions among triggers and procedures could lead to adverse effects such as infinite looping, overwriting of important data, clogging of communications channels, or monopolization of processor resources. We intend to develop visual tools to help users track active triggers and procedures along with complementary analytic tools for detecting possible negative interactions.

Procedure Sharing. Participants in the PAL Double Helix sessions indicated that there would be significant benefit in supporting procedure sharing within and across command centers, both as a way to disseminate best practices and to reduce the slope of the “learning curve” for new soldiers. Building on CPOF’s collaboration architecture, users can easily share procedures by dragging them into a common data repository. To increase the effectiveness of collaboration, we intend to develop a framework for community rating of procedures as well as capabilities to search for procedures by provenance (e.g., who created/revised a procedure, when, frequency/recency of procedure executions, success/failure rates).

Conclusions

We have described our work on transitioning learning by demonstration technology into the CPOF collaborative visualization platform for command and control. Our experience with users over multiple Double Helix sessions and the results of a formal evaluation at the Army BCBL provide evidence that our learning by demonstration technology is both viable and valuable for CPOF users.

The PAL task learning technology enables user extensibility and customizability of software based on the learning by demonstration paradigm. Through the powerful combination of triggers and procedures, CPOF users can significantly reduce time spent on tedious or repetitive tasks and automate routine reactions, freeing them to focus on more cognitively demanding challenges.

Operational benefits of the PAL-CPOF system include

- Reducing reliance on focus-draining manual checks by bringing context to the operator instead of requiring him to hunt for information
- Reducing time-to-respond by eliminating a number of rote/repetitive tasks
- Extending periods of uninterrupted time to focus, think, consider, collaborate, and decide
- Providing economies of scale by focusing on the most frequent tasks, thus significantly reducing operator workload, allowing one operator to perform the work of several
- Supporting the creation of flexible, composable workflows by Army personnel after the software is deployed, thus allowing the PAL-enhanced user community more agility to change processes. This in turn makes users less reliant on costly software developers and cumbersome acquisition processes.

Based on the successes so far and the clear potential seen for the technology by the Army, we are currently looking to transition the task learning technology into an operational setting.

Acknowledgments. This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of DARPA or the Air Force Research Laboratory. The authors thank Hal Dick and Kevin Vallandingham for their significant inputs in shaping the technology, the BCBL for its support in conducting the PAL evaluation, and their many colleagues at SRI and General Dynamics who contributed to the success of this effort.

References

- Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., and Taysom, W. 2007. PLOW: A Collaborative Task Learning Agent. *Proc. 22nd Conference on Artificial Intelligence*, Vancouver, Canada.
- Burstein, M., Laddaga, R., McDonald, D., Benyo, B., Roberston, P., Hussain, T., Brinn, M., and McDermott, D. 2008. POIROT—Integrated Learning of Web Service Procedures. *Proc. AAAI-08*, 1274-1279.
- Cypher, A. (Ed.) 1993. *Watch What I Do: Programming by Demonstration*. Cambridge, MA: MIT Press.
- Eker, S., Lee, T. J., and Gervasio, M. 2009. Iteration Learning by Demonstration. *Proc. of the AAAI 2009 Spring Symposium on Agents that Learn from Human Teachers*.
- Gervasio, M., Lee, T. J., and Eker, S. 2008. Learning Email Procedures for the Desktop. *Proc. AAAI 2008 Workshop on Enhanced Messaging*, Chicago, IL.
- Gervasio, M. T. and Murdock, J. L. 2009. What Were You Thinking? Filling in Missing Dataflow through Inference in Learning from Demonstration. *Proc. 2009 International Conference on Intelligent User Interfaces*.
- Leshed, G., Haber, E., Matthews, T., and Lau, T. 2008. CoScripter: Automating and Sharing How-To Knowledge in the Enterprise. *Proc. CHI*.
- Lieberman, H. 2001. *Your Wish is My Command: Programming by Example*. San Francisco, CA: Morgan Kaufmann.
- Morley, D. and Myers, K. 2004. The SPARK Agent Framework. *Proc. Third International Joint Conference on Autonomous Agents and Multiagent Systems*.
- Myers, K., Berry, P., Blythe, J., Conley, K., Gervasio, M., McGuinness, D., Morley, D., Pfeffer, A., Pollack, M., and Tambe, M. 2007. An Intelligent Personal Assistant for Task and Time Management. *AI Magazine*, 28:2, pp. 47-61.
- Roth, S.F., Kolojejchick, J., Mattis, J., and Goldstein, J. 1994. Interactive Graphic Design Using Automatic Presentation Knowledge. *Proc. SIGCHI'94*, pp. 112-117.
- Roth, S.F., Lucas, P., Senn, J. A., Gomberg, C. C., Burks, M. B., Stroffolino, P. J., Kolojejchick, J., and Dunmire, C. 1996. Visage: A User Interface Environment for Exploring Information. *Proc. Information Visualization*, pp. 3-12.
- Sanders, R., Milligan, R., and Roberts, K. 2009. DARPA Command Post of the Future Enhanced with the Personalized Assistant that Learns 2008 Demonstration. Technical Report, Battle Command Battle Laboratory – Leavenworth.
- Spaulding, A., Blythe, J., Haines, W., and Gervasio, M. 2009. Integrating Task Learning Tools to Support End Users in Real-World Applications. *Proc. International Conference on Intelligent User Interfaces*.