Learning by Demonstration for a Collaborative Planning Environment

Karen Myers, Jake Kolojejchick, Carl Angiolillo, Tim Cummings, Tom Garvey, Matt Gaston, Melinda Gervasio, Will Haines, Chris Jones, Kellie Keifer, Janette Knittel, David Morley, William Ommert, Scott Potter

■ *Learning by demonstration technology has long held the promise to empower nonprogrammers to customize and extend software. We describe the deployment of a learning by demonstration capability to support user creation of automated procedures in a collaborative planning environment that is used widely by the U.S. Army. This technology, which has been in operational use since the summer of 2010, has helped to reduce user work loads by automating repetitive and time-consuming tasks. The technology has also provided the unexpected benefit of enabling standardization of products and processes.*

Learning by demonstration technology has long held the promise to empower nonprogrammers to customize and extend software. Recent technical advances have enabled its use for automating increasingly complex tasks (Allen et al. 2007; Blythe et al. 2008; Burstein et al. 2008; Leshed et al. 2008; Cypher et al. 2010). However, fielded applications of the technology have been limited to macro recording capabilities, which can only reproduce the exact behavior demonstrated by the user.

This article describes the successful deployment of learning by demonstration technology that goes well beyond macro recording by enabling end users to create parameterized procedures that automate general classes of repetitive or time-consuming tasks. This task-learning technology originated as a research system within DARPA's Personalized Assistant That Learns (PAL) program, where it was focused on automating tasks within a desktop environment (Gervasio, Lee, and Eker 2008; Eker, Lee, and Gervasio 2009; Gervasio and Murdock 2009).

From those research roots, PAL task learning has evolved into a fielded capability within the Command Post of the Future (CPOF) — a military command and control (C2) system used extensively by the U.S. Army. The CPOF software is part of the Army's Battle Command System, and as such is standard equipment for virtually every Army unit. Since its inception in 2004, thousands of CPOF systems have been deployed.

CPOF is a geospatial visualization environment that enables multiple users to collaborate in developing situational awareness and planning military operations. Much of CPOF's power comes from its generality, providing tremendous flexibility for handling a wide range of missions. The flip side of this flexibility, however, is that CPOF provides few built-in processes to support specific work flows. As a result, CPOF can require significant user interaction to complete tasks (that is, it is "click intensive").

Task learning provides tremendous value for CPOF by enabling individual users and collective command staffs to create customized, automated information-management schemes tailored to individual preferences and the staff's standard operating procedures, without needing software engineers for extensive recoding. Task learning can reduce work load and stress, can enable managing more tasks with better effectiveness, and can facilitate consideration of more options, resulting in better decisions.

In Learning by Demonstration to Support Military Planning and Decision Making (Garvey et al. 2009), we described the core task-learning technology, its integration into an initial PAL-CPOF prototype, and an extensive exercise-based evaluation of the prototype conducted by the U.S. Army in December, 2008. At this evaluation, users overwhelmingly endorsed the capabilities provided by task learning. Users stressed that task learning "saves time and that time equals lives." The results led to a recommendation by the Army to fully incorporate PAL task learning into CPOF, with the objective of deploying it for operational use.

That objective has been realized: PAL task learning has been integrated into the mainline CPOF system and its incremental fielding throughout the U.S. Army was begun in the summer of 2010. For CPOF users, task learning speeds time-critical processing, eases task loads, reduces errors in repetitive tasks, and facilitates standardization of operations.

This article describes our experiences in transitioning the initial PAL-CPOF prototype to the field, summarizing valuable lessons learned along the way. As one would expect, the deployment process involved refining and hardening the prototype capabilities along with significant systems integration work. In addition, we continuously engaged with users to ensure the technology's usability and utility. This engagement included working closely with a U.S. Army unit that was preparing for deployment, to help personnel incorporate task learning into their operational processes.

One unexpected outcome of the interactions with the unit was an expanded value proposition for task learning, moving beyond the original motivation of automating time-consuming processes to further include standardization of processes and products. We collaborated extensively with the unit to develop a comprehensive library of learned procedures that capture critical work flows for their daily operations. Interestingly, the unit made fundamental operational changes to take greater advantage of the automation enabled by task learning.

We begin with an overview of CPOF followed by a summary of the PAL task learning. We then describe the process of getting to deployment, covering technical challenges encountered, unit engagement activities, and an Army-led assessment of the technology. Next, we discuss the fielding of the technology, including trade-offs made to ensure deployability, the impact of the deployed technology, and lessons learned. We close with a summary of ongoing work to deploy additional functionality and to broaden the user base for task learning in CPOF.

Command Post of the Future

Command Post of the Future (see figure 1) is a state-of-the-art command and control visualization and collaboration system. CPOF originated in a DARPA program focused on advanced user interface design for C2 environments. It grew out of a need to enable distributed command posts to process greater amounts of information and to collaborate effectively on operations. CPOF is built on the CoMotion platform, which was derived from visualization research on SAGE (Roth et al. 1994) and Visage (Roth et al. 1996).

Three design concepts lie at the heart of CoMotion: information centricity, direct manipulation, and deep collaboration.

Information Centricity. CPOF is organized round the principle of direct interaction with information. In any C2 environment, the ability to incorporate new information dynamically is critical to the success of an operation. CPOF uses geospatial, temporal, tabular, and quantitative visualizations specifically tailored to accommodate information in the C2 domain. Though many prior visual analytics tools operate on static data dumps, CPOF's "live" visualizations continually update in response to changes sourced from users' interactions or from underlying data feeds. CPOF is highly composable, permitting users to author new information directly in visualizations or to create composite work products by assembling multiple visualizations in a single container "product."

Direct Manipulation. CPOF makes heavy use of drag-and-drop and other direct manipulation gestures to afford users content management, editing, and view control operations. By employing a small set of interactions with great consistency, simplicity and predictability emerge.

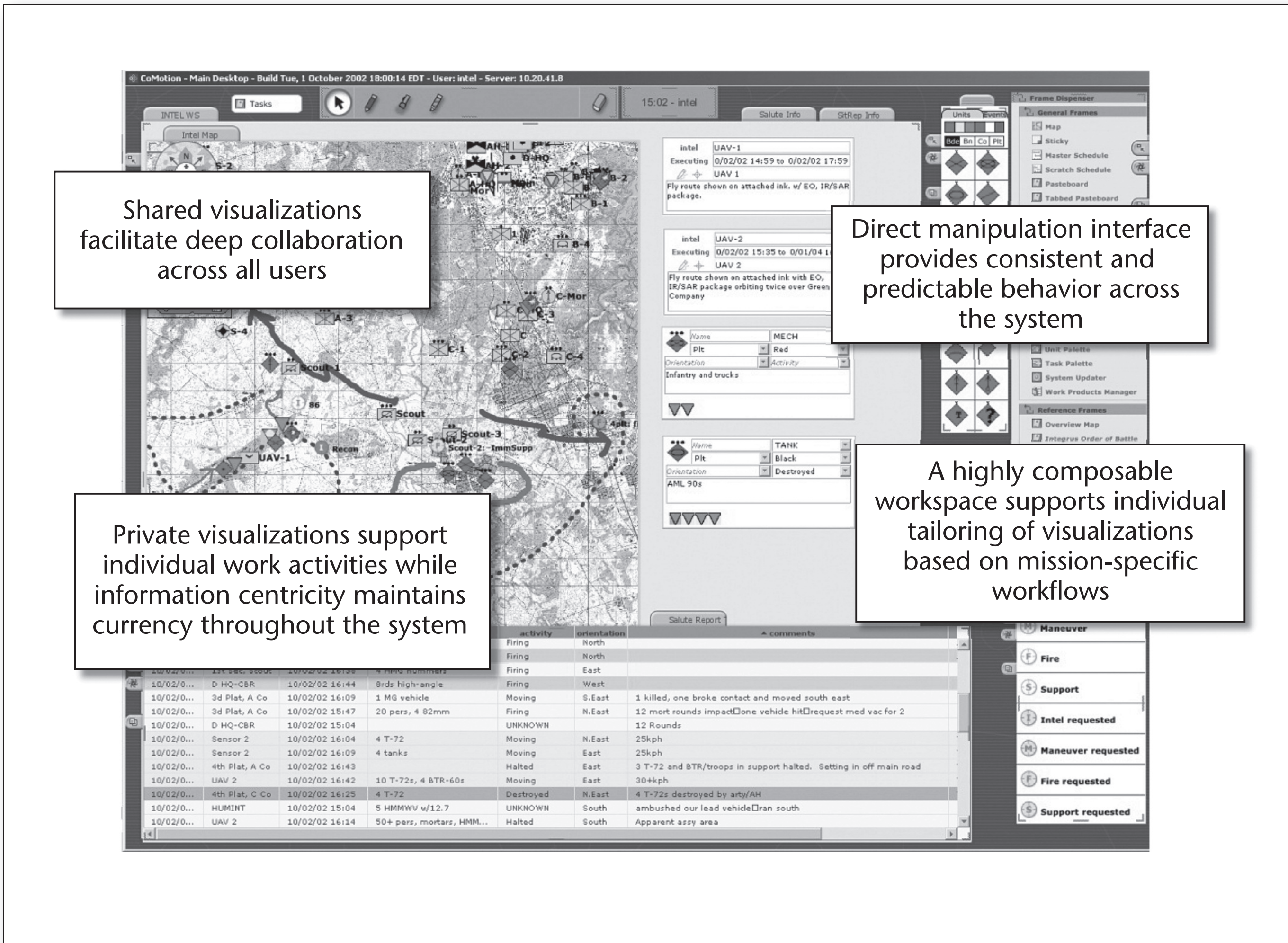

Figure 1. A CPOF Display for a Notional Mission, Showing Geospatial and Tabular Displays of Information along with Palettes for Quick Drag-and-Drop Insertion of Information.

Deep Collaboration. CPOF offers a deep collaboration capability, beyond pixel sharing and chat. Any visualization or composite product in CPOF supports simultaneous interaction by every user with access to it, supporting the collaborative creation of plans and analysis products. Leveraging an "over the shoulder" metaphor, sharing in CPOF happens as a natural side effect of user activities, providing shared visibility among distributed team members (just as sharing occurred naturally among colocated users in command posts prior to CPOF).

CPOF is used daily at hundreds of distributed command posts and forward operational bases. The software spans organizational echelons from corps to battalion, with users in functional areas that include intelligence, operations planning, civil affairs, engineering, and ground and aviation units. CPOF is used extensively to support C2 operations for tasks covering information collection and vetting, situation understanding, daily briefings, mission planning, and retrospective analysis. A detailed description of CPOF's operational utility is provided in Croser (2006).

The CPOF interface enables users to compose and share many "simultaneous but separate" products tailored to task and area of responsibility. One such product is a storyboard, an example of which is depicted in figure 2. A storyboard is typically created in response to a significant event, such as a downed aircraft. While the content and layout vary for different classes of events, storyboards typically provide a summary of key details, relevant operational graphics, impact analyses, and recommended actions. Other products created within CPOF include information briefings, common operating pictures, resource-level estimates, and fragmentary orders.

Users can collaborate synchronously in CPOF by interacting with a set of shared products. The live-

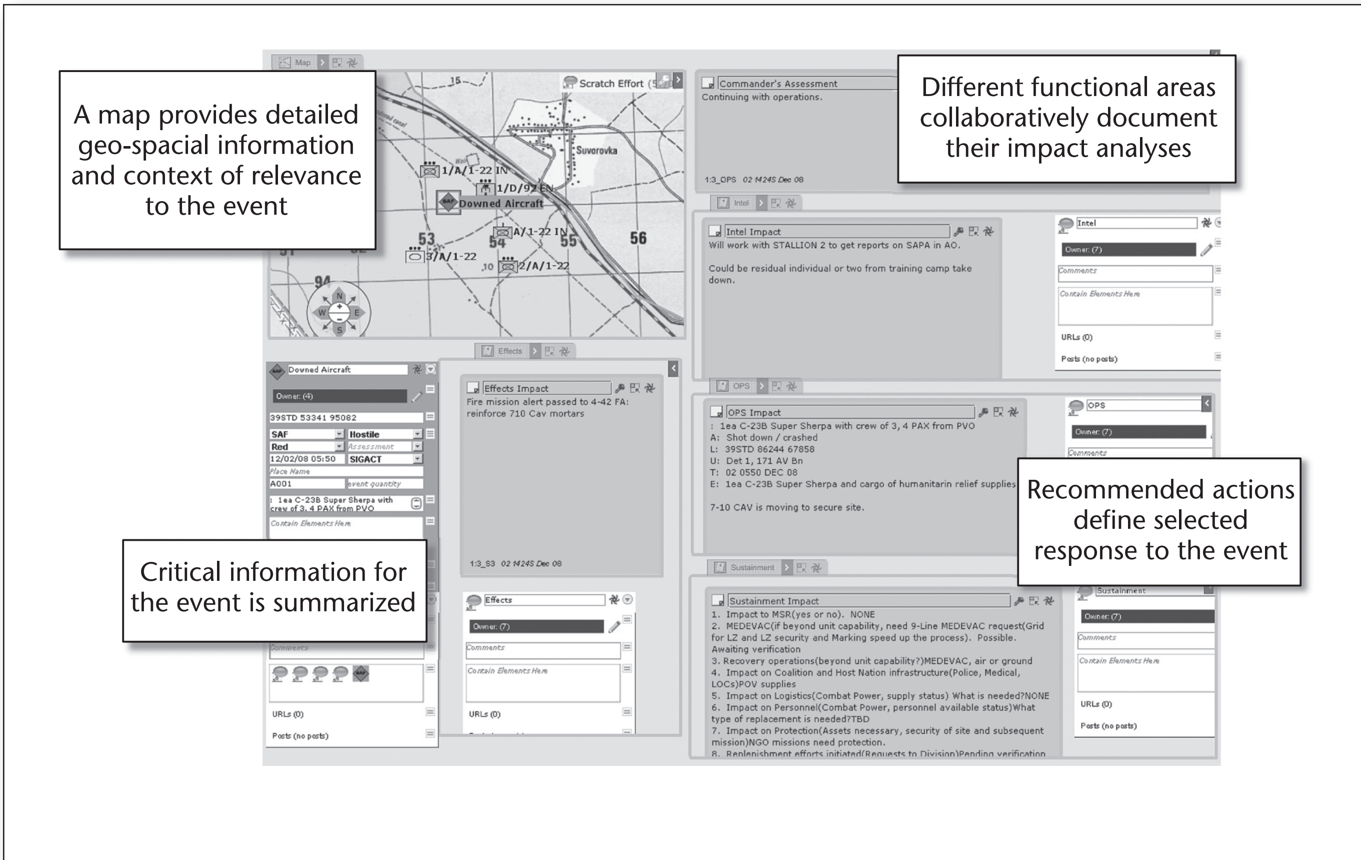

Figure 2. A Sample CPOF Storyboard.

ness of visualizations ensures data updates flow rapidly to users, and the "over the shoulder" metaphor allows one user to monitor another's work, thus minimizing interruptions with requests for information. Collaboration group size spans the spectrum from individual (analysis, event tracking) to small group (mission planning) to organization wide (update briefings, mission rehearsals).

Each major deployment typically has several data partitions (each a CPOF repository) that allow the U.S. Army to manage risk and redundancy for the overall system. Each major deployment has dedicated field support representatives and senior trainers residing with the unit on their bases to administer the system and support its users.

Task Learning in PAL-CPOF

The simplicity of interaction with CPOF affords users great flexibility in creating rich visualizations over a wide variety of data. But this powerful capability often comes at the cost of time-consuming, click-intensive processes. Task learning addresses this problem by letting users automate repetitive processes, allowing them to focus on more cognitively demanding tasks. For example, the outline for the storyboard depicted in figure 2 can take many minutes to create manually, thus slowing down the process of responding to time-critical events. Task learning can be used to create a procedure that generates a storyboard template automatically when invoked, enabling the soldier to focus immediately on content for the storyboard rather than having to devote precious time to rote visualization construction.

In the paper by Garvey et al. (2009), we described the initial integration of task learning into CPOF. Here, we provide a brief overview of the learning technology, the procedure execution and editing functionality, and the rules and work flows that together constitute the task-learning capability added to CPOF.

LAPDOG: Learning by Demonstration

The LAPDOG system (Gervasio, Lee, and Eker 2008; Eker, Lee, and Gervasio 2009; Gervasio and Murdock 2009), depicted in figure 3, provides the learning by demonstration capabilities from the PAL program that support task learning within CPOF. Given a trace consisting of a sequence of

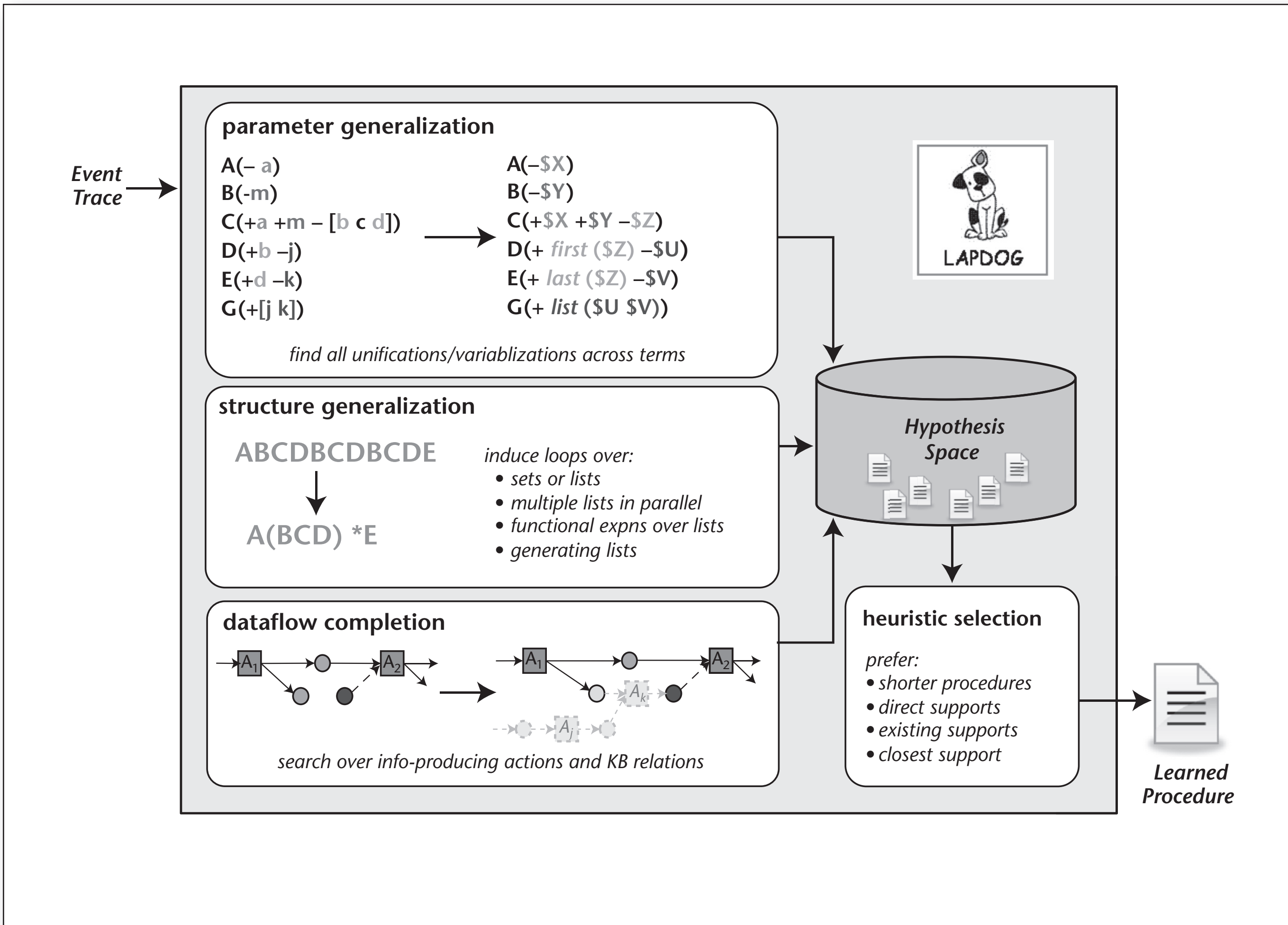

Figure 3. Learning Procedures by Demonstration in LAPDOG.

actions performed by a user to achieve a particular task, LAPDOG generalizes the trace into a procedure for achieving similar tasks in the future.

An *action model* provides the semantic description of the demonstrable actions in an application, together with mechanisms for the instrumentation and automation of those actions. An early challenge in the development of the CPOF action model was finding the right level of abstraction. Capturing actions at the level of primitive data changes (for example, create an entity, move a contained entity to the front/back of its container) allowed for a very compact action model and greatly simplified instrumentation but led to lengthy procedures with minimal generalization and poor comprehensibility. To support an action model at the level humans typically describe their actions (for example, dispense a frame, center a map on an entity) required investing in a translation layer to convert lower-level events into corresponding higher-level actions. The resulting action model thus comes at a higher cost of instrumentation but enables more successful generalization and increased understandability for visualization and editing. The current action model covers most, although not all, of the core user activities in CPOF.

LAPDOG employs a *data flow model* in which actions are characterized by their inputs and outputs, with outputs serving as inputs to succeeding actions. LAPDOG generalizes demonstrations into procedures in two ways. First, it performs *parameter generalization,* unifying action outputs and inputs and replacing constants with variables or expressions over variables to capture the data flow in the demonstration. Second, it performs *structure generalization,* inducing loops over collections of objects by generalizing repeated sequences of actions.

LAPDOG's loop induction algorithm supports advanced features such as the ability to learn loops that iterate over ordered or unordered collections of objects (lists and sets), involve expressions over those collections, process multiple lists simultaneously, and accumulate outputs. Although a powerful generalization technique, loop induction has been surprisingly difficult to see fully realized in CPOF for two primary reasons: (1) iteration is a difficult concept for end users, and (2) identifying collections in CPOF currently involves idiosyncratic gestures in the user interface.

LAPDOG features the ability to infer action sequences for *completing data flow* in situations where the linkage between outputs and inputs is implicit. For example, in preparing a text message based on an incident report, the user will typically use information from particular fields in the report. Because instrumentation will not capture this linkage, without data flow completion the learned procedure would require the message text to be input by the user, thus losing the implicit relations between the report fields and portions of the message. However, by inferring these relations, LAPDOG can instead induce a parameterized message template for use in preparing text messages for the procedure. This inference of implicit data flow improves LAPDOG's ability to generalize across the parameters of procedure actions, reducing the inputs required for a learned procedure and so simplifying its use.

LAPDOG was not specifically designed to learn from a single example but that has become its primary mode of use in CPOF since users have generally been unwilling to provide multiple demonstrations. LAPDOG generalizes from a single example with the aid of heuristics for filtering the set of alternative hypotheses. Specifically, it prefers more recent or more direct supports, action outputs over procedure inputs, existing procedure inputs over new ones, and loops over straight-line alternatives.

Procedure Execution and Editing

In the initial PAL-CPOF prototype, the SPARK agent framework (Morley and Myers 2004) was used to execute learned procedures. SPARK is a feature-rich system designed to support the sophisticated control and reasoning mechanisms required by practical agent systems (for example, Morley, Myers, and Yorke-Smith 2006; Yorke-Smith et al. 2009). The execution of learned procedures within the deployed system is provided by a lightweight version of SPARK focused specifically on procedure execution.

Because many CPOF users have no programming experience, it is important to provide comprehensible visualizations that communicate what a learned procedure does and to assist users in making valid edits. This is particularly important in collaborative environments such as CPOF, to allow users to benefit from being able to modify and reuse procedures created by others. On the PAL project, we developed technology for procedure visualization and assisted editing (Spaulding et al. 2009). Our approach, refined through feedback from numerous user engagements, augments the action model with a flexible metadata capability to improve visualization and editing. It also employs procedure analysis techniques to detect editing problems and suggest fixes.

Composing Higher-Level Work Flows

Learned procedures provide significant value by automating time-consuming functions in CPOF, such as the creation of storyboard templates described earlier. Our earlier interactions with the U.S. Army revealed that the value of task learning was greatly enhanced by introducing rules to enable automatic invocation of learned procedures based on some triggering event. Rules were used extensively in the original PAL-CPOF prototype to facilitate rapid response to a range of operational events. Three types of rules were supported: time rules, data rules, and area rules. Time rules invoke procedures at absolute or relative times or with a specific frequency, data rules invoke procedures based on particular changes to properties of objects in the system, and area rules invoke procedures based on particular changes to predefined areas on a map. For example, a data rule could be created that invokes a procedure for creating a storyboard template tailored to a particular event class when an event of that type is recorded in the system.

Taking this concept a step further, as shown in figure 4, it is possible to compose collections of related rules and procedures into more complex work flows that automate larger chunks of functionality. Work flows open the door to automating critical responses to significant events in accord with standard operating procedures (SOPs). For example, PAL work flows have been created that automatically construct and arrange work spaces, notify selected staff of events, track assets, and prepare simple reports for the user's approval and dissemination. As discussed below, work flows proved instrumental to our deployment strategy but introduced some additional challenges.

Given that work flows encompass multiple procedures and rules that can interact in potentially subtle ways, creating complex work flows requires a level of sophistication not met by most CPOF users. Successful management of work flows required us to consider classes of users with different capabilities and responsibilities, A significant part of our design effort leading up to deployment was devoted to creating tools and best practices to facilitate understanding and management of an ecosystem of interacting procedures and rules by users of differing skill levels.

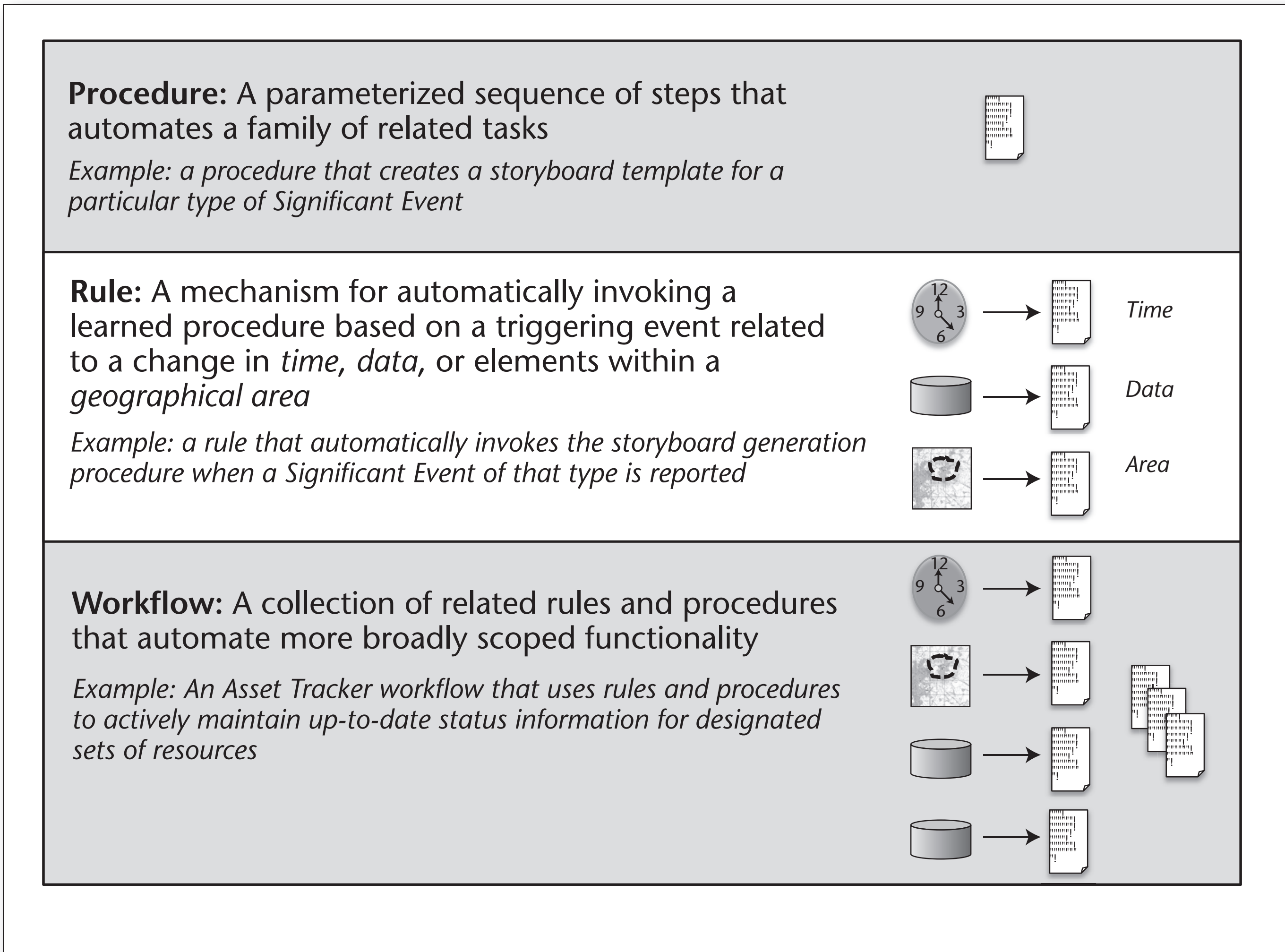

Figure 4. PAL Automation Elements: Procedures, Rules, and Work Flows.

Getting to Deployment

Our efforts to deploy the PAL task-learning technology within CPOF involved two main thrusts. One focused on the hardening and refinement of the task-learning capabilities, along with their integration into the mainline CPOF system. The second focused on user engagement to ensure the operational effectiveness of the technology. This section describes key challenges that arose on the technical side, along with our user engagement efforts and their impact on the development and deployment processes.

Technical Challenges

Transitioning the task-learning technology into CPOF presented numerous software-engineering challenges, independent of the learning capability itself. The collaborative nature of CPOF requires integration to be sensitive to problems inherent to distributed, multiuser systems. CPOF is a fielded platform but also continues to evolve over time. Finally, the data that CPOF manages is dynamic and voluminous.

Deployment of the technology also raised a number of challenges more directly linked to task learning. One was developing an expressive yet sustainable action model. To address this challenge, we evolved our action model framework to be extensible so that it could both grow to accommodate changes in CPOF and enable procedures from prior versions to be upgraded easily to new versions.

High-volume, concurrent procedure execution presented a second challenge. The PAL task-learning technology was developed originally as part of an intelligent desktop assistant, for which execution typically involved small numbers of procedures executed one (or a few) at a time under explicit user control. During exploration of the use

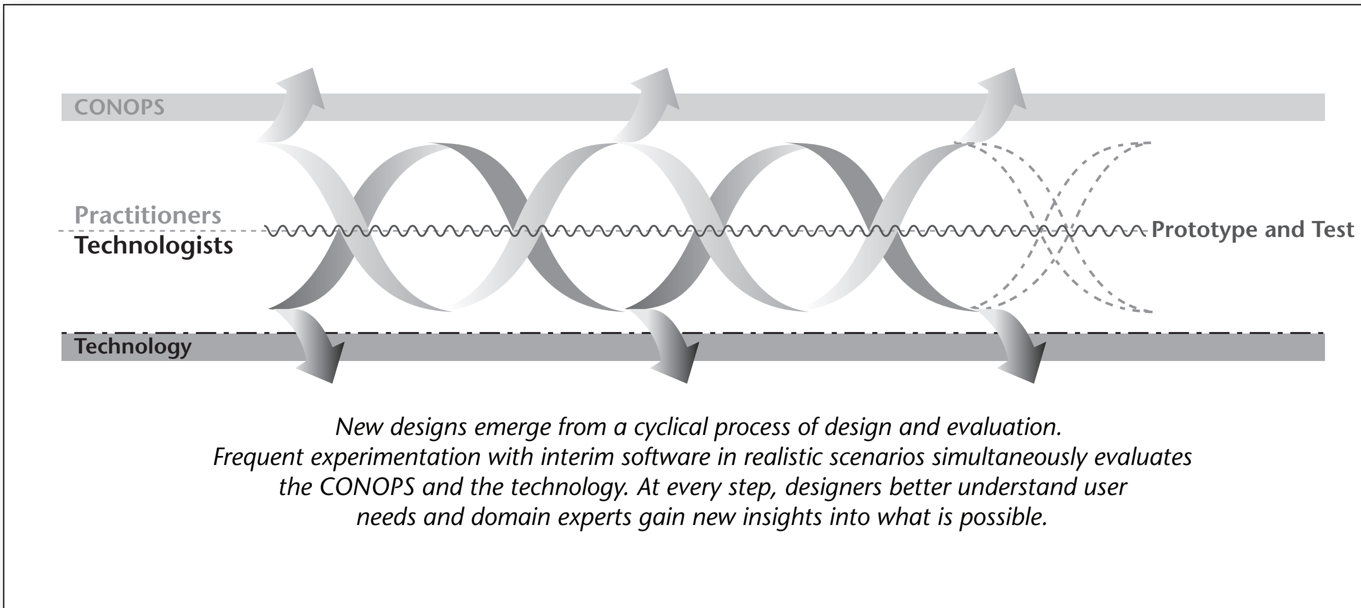

Figure 5. CPOF'S Double Helix Process: Building Common Ground between Technology Developers and Domain Experts.

of the task-learning technology within CPOF, the automatic invocation of procedures through rules took on an increasingly significant role. In the deployed version of the system, rule-based invocation can lead to hundreds of procedures executing simultaneously, which was well beyond the design parameters for the desktop assistant. Addressing concurrency with high volumes of procedure execution required a complete reimplementation of the PAL-CPOF communications infrastructure to eliminate timing problems triggered by these higher than anticipated invocation rates.

Another significant challenge was the difficulty of porting PAL products (procedures or work flows) between units, which have data elements specific to their needs. Task learning generates parameters based on the data elements referenced in a demonstration. Procedures easily fit into user work processes because dependencies on the unit-specific data partition are encoded as default parameters. However, ensuring that a PAL product works for other units requires "rewiring" those defaults to the appropriate local data sources. For example, one key data element within a CPOF repository is the SigActs table, which records the significant actions of relevance to a particular unit. A unit linked to a different CPOF repository would have its own local SigActs table. For a procedure taught within the context of one repository to work on a second repository, default references to the SigActs table would need to be reset accordingly. For complex products, this localization process proved to be particularly demanding, requiring deep knowledge of both the author's intent and the unit's data configuration. For the initial deployment, we relied on human experts to perform this task.

User Engagement Prior to Fielding

Our transition plan emphasized continuous user interactions to develop and refine the task-learning concept in CPOF. To this end, we conducted a series of user-centered design exercises with the deploying unit. These exercises were conducted using the Double Helix methodology that was applied successfully to the development and deployment of the initial CPOF system.

The name Double Helix derives from simultaneous technology refinement based on insights from user interactions and refinement of the Concept of Operations (CONOPS) for its use within the operational domain (see figure 5). In the Double Helix methodology, technologists work with users in realistic settings to uncover technology issues, understand the operational "sweet spot" for the capability, and identify opportunities for its use to solve relevant operational problems.

Our Double Helix exercises yielded several insights that informed our deployment effort.

Initial demonstrations of simple PAL capabilities facilitated user acceptance of the technology. In general, introducing new technology into a large-scale, distributed operational environment requires strict feature specifications and configuration control. In addition, deployment typically requires safety and encapsulation for users with limited training while placing unfortunate restrictions on more experienced users. Such requirements run counter to the spirit of end-user task learning and could be expected to reduce user acceptance of such technology. To mitigate this risk, our engagement team introduced PAL by initially creating small procedures that automated routine, repetitive tasks rather than complex work flows. By first understanding these simple examples, users were more likely to embrace the technology and to conceive of more complex support that PAL could provide. This approach proved successful in enabling low-risk, progressive adoption of the technology.

Creation of a library of automation products built by expert users greatly increased adoption of the technology. During user engagements, we identified CONOPS common to multiple units that could be supported by generic, but sophisticated, PAL work flows. Expert users were able to automate several of these "best practice" work flows using task learning. This led to the introduction of a PAL Library as a way to disseminate these more advanced work flows.

The PAL Library significantly strengthened the capability deployment process. First, it conveyed to users the "art of the possible" (that is, what can be done with the PAL technology). Prior to development of the library, users often had limited understanding of how PAL products could help them. Afterward, there was a marked increase in capability expectations among the user base. Second, it provided complete interactive products with more compelling functionality than individual procedures. Third, it enabled users to reap the benefits of PAL without having to understand how to create procedures and rules. Finally, the library created an effective conduit to introduce enhanced capabilities into the field.

Incremental integration of PAL capabilities into operational SOPs was essential for successful technology transition. Initial attempts to automate collaborative processes monolithically failed because the changes imposed on the users were too substantial. Based on these early experiences, we switched to a more gradual approach that would enable PAL work flows to be adopted incrementally into existing SOPs. Our adoption path consisted of the following steps: (1) an isolated work flow for a single user, (2) an individual work flow shared by multiple users, (3) a set of individual work flows within a PAL-initialized organizational process, and (4) a set of collaborative work flows supporting multiple users. Following this path facilitated incremental adoption rather than an all-or-nothing change, which led to increased user acceptance for the technology. In this way, adoption could be "contagious." This approach enabled the use of task learning to expand beyond automating current SOPs to providing higher-order decision support that would not otherwise have been practical.

NTC Assessment

The U.S. Army performed an assessment of the PAL software midway through the project to determine whether to proceed with deployment. The assessment targeted both the operational utility of the PAL capabilities and system performance metrics (for example, bandwidth utilization, server/client performance, software quality). To enable realistic operating conditions, the Army chose to perform the assessment in conjunction with the unit's standard three-week rotation at the National Training Center (NTC) at Ft. Irwin in May of 2010.

The purpose of the NTC rotation was to prepare the unit for operational deployment through a variety of training exercises, culminating in a "final exam" executing a realistic mission overseen by an experienced team of observer/controllers. The unit was provided with an engineering release of a version of CPOF that included the learning technology and the PAL Library that it had helped to develop in the months leading up to the event. The unit had the freedom to use PAL capabilities to the extent that it found them helpful for completing their assigned mission; however, evaluation of the PAL capabilities was not an explicit objective for the soldiers.

Assessment of the PAL technology was done through a series of questionnaires that were administered to members of the unit and the observers/controllers. These questionnaires solicited feedback on both the usability of the PAL technology and the extent to which it enabled the unit to conduct its operations more effectively. Based on the responses to these questionnaires, the Army concluded that the PAL technology significantly improved CPOF operations throughout the rotation, in certain situations reducing hours of work to minutes. A report summarizing the assessment stated that PAL capabilities "improve a unit's capability to efficiently execute their roles, functions and missions through better collaboration and information sharing." It was also determined that the PAL capabilities did not negatively affect the system performance metrics of interest. Given these results, the Army decided to proceed with the full integration of PAL task learning into CPOF and to allow the unit to deploy with the PAL capabilities.

Fielding

The original plan for deployment involved a roughly two-year effort split over two phases. The first phase was to begin integration into the mainline CPOF system while concurrently developing new functionality (informed by user engagement) that would further enhance the value of task learning within CPOF. The second phase was to focus on hardening and integration, with the objective of delivering the final capability for operational deployment.

Early in the effort, the U.S. Army requested that the deployment cycle be accelerated in order to get the task-learning technology into the hands of users faster. This expedited fielding led to a requirement for a hardened capability that was fully integrated into CPOF roughly nine months after the start of the effort. Not all of the planned technical functionality could be sufficiently hardened and integrated to meet this aggressive new schedule. The end result was that capabilities would be limited in the first release, with additional functionality planned for incorporation into subsequent releases of CPOF.

Functionality Trade-offs

Two main considerations determined what functionality to make available in the first deployment.

One consideration was the potential for users to teach and execute procedures with deleterious side effects, such as compromising critical data or interfering with system operation.

A second consideration was the need for training to take full advantage of the capabilities afforded by the PAL technologies. Our Double Helix interactions introduced us to the real world of deployment, redeployment, and assignment rotation, wherein the pool of users available for training changed frequently, never really reaching the critical mass of skill needed to fully understand CPOF and PAL. Indeed, many of the users with whom we interacted obtained the bulk of their CPOF knowledge through informal, on-the-job training in the form of interactions with more skilled colleagues. This sort of informal training rarely affords the trainee the opportunity to create a mental model sufficient to understand the consequences of end-user programming.

Based on these considerations, we decided not to make the learning by demonstration capability directly available to all users of CPOF in the initial deployment. Rather, a typical user could instead access prebuilt procedures and work flows constructed by a small team of well-trained library builders. A typical user has permission to run any available library product (work flow or procedure) but not to extend or modify library products, or even create stand-alone procedures that address their individual needs. Library builders have access to the core learning by demonstration capability in order to extend and modify the library in theater. In our initial deployment, the library builders are field support and training personnel, who are more readily trained and have greater facility with programming constructs, making it easier for them to demonstrate and manipulate procedures. It is important to note, however, that these library builders are typically not professional software engineers and gain a great deal of flexibility and power from the task-learning capability.

Imposing this restriction was a difficult decision to make, as the team is fully committed to making the learning technology available to all CPOF users. Ultimately, however, we decided that it would be advisable to roll out the technology incrementally to provide some of the benefits of the learning capability while minimizing associated risks. As discussed in the Ongoing Work section below, we are continuing to extend the range of the capabilities that are available to all users of the system and are on track to deploy those in future releases of CPOF.

Operational Deployment

For approximately six months beginning in late summer of 2010, a team deployed first to Iraq and then to Afghanistan to upgrade CPOF to a version containing the PAL learning technology. The upgrade team consisted of senior trainers to conduct classes and teach best practices, software developers to oversee the upgrade and collect performance and debugging information, and a unit-engagement team to help tailor and extend the PAL Library using task learning. Over several months, the upgrade team visited each of the CPOF installations and spent 2–5 weeks conducting training classes, upgrading the repository, and providing one-on-one support.

Despite the military's reputation as a regimented organization, its processes can be surprisingly decentralized; each unit has a unique operating environment and is given a fair degree of leeway to accomplish its missions. Further, each commander sets his own operating procedures and reporting requirements. The engagement team first collected information from the unit about its local processes, then used this data to customize the generic, previously developed library work flows to the needs of the individual unit. The engagement team iterated with the commanders and CPOF operators to teach them how to customize and use these work flows, and helped with further refinements. Concurrently, the team trained the field support representatives to provide ongoing support for the new capabilities.

There was substantial variability in adoption levels for the task-learning technology. One unit that had just arrived in theater was adapting the processes left behind by the outgoing unit and welcomed the assistance to refine and automate their processes. In cases like this, the engagement team observed substantial adoption of PAL work flows and a high level of enthusiasm for the new capabilities (with one unit giving the team a standing ovation and high-fives after a demonstration of the capability). In contrast, a unit that was close to rotating out of theater had little interest in taking on the risk of new processes and tools.

One indicator of the degree of adoption of PAL was the number of modification requests received. In a few cases, the engagement team had little contact with the unit after customizing its PAL work flows and conducting training; in other cases, the team received a steady stream of requests for changes and new features. The ability to receive a requested enhancement in days to weeks rather than the typical 18-month software release cycle for CPOF generated tremendous excitement, opening the door for units to adapt CPOF on the fly to in-theater dynamics.[1]

In several instances, the engagement team worked with users who had minimal CPOF experience. This inexperience was a disadvantage during training because it meant spending a substantial amount of time on basic system usage. However, this inexperience also provided a substantial advantage — these users drew little distinction between native CPOF capabilities and PAL enhancements. This allowed them to leapfrog their more experienced peers: rather than first learn a manual process and then understand the PAL procedures to automate that process, these users gravitated straight to using the automation.

One of the biggest lessons learned was the inverse correlation between the user-perceived complexity of a PAL work flow and the degree to which it was adopted. For example, the "storyboard creator" work flow allowed an individual soldier to create a visual summary of a significant event in only a few clicks. This work flow fit well within existing processes and produced storyboards that looked similar to those that the soldiers created manually. As such, soldiers could incorporate this work flow into existing processes with minimal change. This incremental adoption allowed a few soldiers to try the new tools, while others took a wait-and-see approach. Adoption spread as word-of-mouth recommendation overcame initial reluctance.

On the other hand, a powerful but complex SigAct (that is, significant action) management work flow saw substantially less adoption. This may have been partially due to the modeled processes being overly specific to the unit that we worked with prior to deployment. Worse, its benefits were dependent on having all of the participants in the SigACT reporting process switch to the new work flow at once. Beyond the original unit, there was little enthusiasm for moving everyone to this new process simultaneously. Based on these findings, the engagement team made modifications to the SigACT work flow so that soldiers can use portions of it without requiring a specific prescribed reporting structure.

As units embraced the PAL technology, some began to formulate new automated work flows based on their individual operational needs. Many of these work flows centered on automating existing processes. However, a handful, such as an asset tracker, suggested entirely new unit SOPs (for example, making a transition from manually editing text tables to tracking and recording asset status automatically), thus showing evidence of users modifying their behavior to more fully take advantage of the technology.

Ongoing Work

As noted above, concerns about safety, usability, and training led to our decision to restrict access to task learning in the initial deployment. To address these concerns, we have delivered several capabilities for incorporation into the next release of CPOF and are continuing to develop others for a third and final planned integration phase. We expect that incorporating these mechanisms will enable responsible extension of the task-learning capability to all users of CPOF.

One strategy has been the introduction of safeguards to limit inappropriate procedure executions. These safeguards range from permissions and type checking for identifying procedures that are expected to fail during execution, to identifying loops with inordinately large numbers of iterations at execution time (which could signal a misguided application of the procedure). A second strategy has been additional support for navigating the library, making it easier for users to locate work flows of interest and instantiate them to work in their local environments.

Finally, we have built an interactive editor within CPOF that enables users to visualize and modify learned procedures. The editor, derived conceptually from the original PAL prototype, is expected to increase understanding, and hence acceptance, of the task-learning technology among basic users while providing advanced users with the means to adapt learned procedures quickly rather than demonstrating them again from scratch to incorporate changes. The ability to modify and reuse procedures created by others is particularly valuable in CPOF, where sharing is inherent to the underlying collaborative user experience.

Conclusions

The PAL learning by demonstration technology has seen wide adoption within CPOF, a U.S. Army system of record employed daily by thousands of soldiers as an integral part of their mission operations. The open-ended composability of CPOF makes it an ideal target for the end-user automation enabled by the learning by demonstration technology. Although integration into CPOF presented numerous technical and operational challenges, the result is a customizable system that reduces time spent on tedious tasks while also facilitating standardization of best practices. By freeing soldiers to focus on higher-level tasks, PAL significantly improves mission-execution capabilities.

Our experiences in fielding the task-learning technology to the U.S. Army offers guidance for future such efforts. User acceptance is greatly facilitated by a deep understanding of the processes that the technology will support, flexibility to allow incremental adoption, and ongoing user engagement that includes technology demonstrations to highlight "the art of the possible."

Acknowledgments

This material is based upon work supported by the United States Air Force and DARPA under Contracts No. FA8750-05-C-0033 and FA8750-09-D-0183. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force and DARPA.

The authors thank Dr. Robert Kohout (DARPA), LTC Richard Hornstein (Army TBC), and Andy Harned (Army TBC) for their leadership and support, and BG(R) Pat O'Neal, MG(R) Thomas Garrett, COL Reginald Allen, COL Justin Gubler, LTC Michael Spears, LTC Brian Gruchacz, MAJ Charles Ayers, MAJ Anthony Whitfield, and Marshall Reed, for their assistance in aligning the task learning technology with user needs.

The authors also thank the following individuals for their contributions to system development, library building, and user engagement: Chick Agnew, Susan Alsue, Naomi Anderson, Stuart Beltson, Sonya Berkovich, Sharon Booth, Kåre Digernes, Patrick Dominic-Savio, Vanessa Fine, Eric Gardner, Dave Goldstein, Daragh Hartnett, Tim Holland, David Jacobs, Melinda Janosko, Chris Joslin, Suzanne Kafantaris, Eric Kaun, Anthony Kim, Warren Knouff, Tom Kilsdonk, Tom Lee, Pat Martin, Tom Martin, John Parise, Seth Porter, Perry Rajnovic, Andrew Riggs, Shahin Saadati, Joan Saz, Dennis Sigel, Rob Smoker, David Stevens, Jim Straub, Rebecca Torbochkin, and Rob Umberger.

Approved for public release, distribution unlimited.

Note

1. The nine-month deployment of the PAL capability was enabled by the army's desire to transition the technology as soon as possible, which led to an opportunistic incorporation of the technology into a previously planned release.

References

Allen, J.; Chambers, N.; Ferguson, G.; Galescu, L.; Jung, H.; Swift, M.; and Taysom, W. 2007. PLOW: A Collaborative Task Learning Agent. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Blythe, J.; Kapoor, D.; Knoblock, C. A.; Lerman, K.; and Minton, S. 2008. Information Integration for the Masses. *Journal of Universal Computer Scienc*e 14(11): 1811–1837.

Burstein, M.; Laddaga, R.; McDonald, D.; Benyo, B.; Cox, M.; Roberston, P.; Hussain, T.; Brinn, M.; and McDermott, D. 2008. POIROT — Integrated Learning of Web Service Procedures. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence,* 1274–1279. Menlo Park, CA: AAAI Press.

Croser, C. 2006. Commanding the Future: Command and Control in a Networked Environment. *Defense and Security Analysis* 22(2): 197–202.

Cypher, A.; Dontcheva, M.; Lau, T.; and Nichols, J., eds. 2010. *No Code Required: Giving Users Tools to Transform the Web*. Amsterdam: Morgan Kaufmann.

Eker, S.; Lee, T.; and Gervasio, M. 2009. Iteration Learning by Demonstration. In *Agents That Learn from Human Teachers: Papers from the 2009 AAAI Spring Symposium*. Technical Report SS-09-01. Menlo Park, CA: AAAI Press.

Garvey, T.; Gervasio, M.; Lee, T.; Myers, K.; Angiolillo, C.; Gaston, M.; Knittel, J.; and Kolojejchick, J. 2009. Learning by Demonstration to Support Military Planning and Decision Making. In *Proceedings of the of the 21st Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Gervasio, M.; Lee, T.; and Eker, S. 2008. Learning Email Procedures for the Desktop. In *Enhanced Messaging: Papers from the 2008 AAAI Workshop*. Technical Report WS-08-04. Menlo Park, CA: AAAI Press.

Gervasio, M., and Murdock, J. 2009. What Were You Thinking? Filling in Missing Dataflow through Inference in Learning from Demonstration. In *Proceedings of the 2009 International Conference on Intelligent User Interfaces*. New York: Association for Computing Machinery.

Leshed, G.; Haber, E.; Matthews, T.; and Lau, T. 2008. CoScripter: Automating and Sharing How-to Knowledge in the Enterprise. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI)*. New York: Association for Computing Machinery.

Morley, D., and Myers, K. 2004. The SPARK Agent Framework. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*. Los Alamitos, CA: IEEE Computer Society.

Morley, D.; Myers, K.; and Yorke-Smith, N. 2006. Continuous Refinement of Resource Estimates. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*. New York: Association for Computing Machinery.

Roth, S. F.; Kolojejchick, J.; Mattis, J.; and Goldstein, J. 1994. Interactive Graphic Design Using Automatic Presentation Knowledge. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI 1994),* 112–117. New York: Association for Computing Machinery.

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. In *Proceedings of the IEEE Symposium on Information Visualization,* 3–12. Piscataway, NJ: Institue of Electrical and Electronics Engineers.

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 – Ft. Leavenworth. Washington, DC: US Army.

Spaulding, A.; Blythe, J.; Haines, W.; and Gervasio, M. 2009. From Geek to Sleek: Integrating Task Learning Tools to Support End Users in Real-World Applications. In *Proceedings of the 2009 International Conference on Intelligent User Interfaces*. New York: Association for Computing Machinery.

Yorke-Smith, N.; Saadati, S.; Myers, K.; and Morley, D. 2009. Like an Intuitive and Courteous Butler: A Proactive Personal Agent for Task Management. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Karen Myers is a principal scientist in the AI Center at SRI International, where she directs a research group focused on planning and agent technologies. Her current interests lie with the development of mixed-initiative AI systems that enable humans and machines to solve problems cooperatively. She has a Ph.D. in computer science from Stanford University.

Jake Kolojejchick is an entrepreneur and innovator focused on creating effective, natural decision-making environments that allow diverse, distributed teams to cope with highly uncertain, information-intensive domains. He is currently chief scientist at General Dynamics C4 Systems, pursuing research in mixed-initiative systems, cyber situational awareness, and autonomy in sensor systems.

Carl Angiolillo is a user experience designer specializing in making complicated things seem simple. He has an M.S. in human-computer interaction from Carnegie Mellon University where he programmed intelligent tutoring systems and has previous experience in software development, teaching, and theatrical set design.

Tim Cummings is a staff software engineer with General Dynamics C4 Systems. He has a B.S. in computer science from Brigham Young University and has 12 years of industry experience on a broad range of commercial and military software projects.

Tom Garvey is a principal scientist and associate director of SRI International's AI Center where he oversees groups developing technologies in planning, human-centered control, agents, robotics, knowledge acquisition, learning, and command and control. He has a Ph.D. in electrical engineering and computer science from Stanford University.

Matt Gaston is the director of the Carnegie Mellon University (CMU) Cyber Innovation Center and an adjunct associate professor at the CMU Institute for Software Research. Prior to joining Carnegie Mellon, Gaston was the director of research for General Dynamics C4 Systems and before that spent nearly 10 years at the National Security Agency.

Melinda Gervasio is a senior computer scientist in the AI Center at SRI International, where she works on integrated intelligent systems that utilize machine learning, automated reasoning, and human interaction, with a particular focus on learning procedural tasks. She has a Ph.D. in computer science from the University of Illinois at Urbana-Champaign.

Will Haines is an interaction designer in the AI Center at SRI International, where he designs usable interfaces for end-user programming systems. His interests lie in deploying advanced technology into real-life operational settings. He has degrees in computer science and human-computer interaction from Carnegie Mellon University.

Chris Jones is a senior software engineer at SRI International. His interests include concurrency, networks, and multiagent architectures. He has a B.S. in computational biochemistry from Montana State University.

Kellie Keifer is the operations director for the AI Center at SRI International and the program manager for the PAL Military Transition effort. Keifer has extensive experience in the transition of research technology into operational systems, including projects in air traffic management prior to joining SRI. She has a master's degree in computer science from the University of Illinois, Urbana-Champaign.

Janette Knittel is the lead product concept designer for General Dynamics C4 Systems | Viz. Knittel is an expert in designing and developing systems for soldiers, commanders, and analysts in a battle command environment. She has a bachelor of fine arts degree from Carnegie Mellon University.

David Morley is long-time consulting research scientist for SRI International and a former SRI employee. His interests are in practical applications of agent systems, especially those that involve human interaction and that deal with failure and change. He has a Ph.D. in computer science from the University of Melbourne, Australia.

William Ommert is a principal engineer with General Dynamics C4 Systems | Viz where he leads research and production software development efforts focused in the areas of visualization and system interoperability. He has a B.S. in mathematics and computer science from Carnegie Mellon University and has previous experience in robotic systems design.

Scott Potter earned a Ph.D. in cognitive systems engineering from the Ohio State University in 1994 where his research focused on representing intelligent system information for fault management. He is currently a senior systems engineer at General Dynamics C4 Systems | Viz, where he designs visualization-centered decision-support systems for complex, dynamic, high-risk environments.