Just in Time Lessons Learned: Timely Retrieval of Operational Vignettes

Laurie Waisel, Wes Regian, Chad Davis, Kamal Gella
Concurrent Technologies Corporation
Johnstown, PA
{waisel, regianj, davisc, gella}@ctc.com

ABSTRACT

The U.S. Military Services have compiled an impressive collection of lessons learned. A warfighter can study only so many lessons learned, and even fewer can be recalled at a critical decision point. This paper addresses the challenge of retrieving the right lesson learned at the right time, and how we are responding to that challenge in developing the Military Analogical Reasoning System (MARS). MARS includes a corpus of operational vignettes (i.e., battle stories) that draws on a comprehensive military ontology for representation of vignette components. The corpus can be searched using a similarity-based retrieval method—structure mapping—to find a vignette from the corpus (the base vignette) that is structurally analogous to a vignette of interest (the target vignette). Calculation of the similarity quotient involves mapping patterns of events between the target and the base. Besides identifying similar vignettes, MARS generates a lesson learned that applies to both of the vignettes. The lesson learned is abstracted from the event pattern(s) that the base and target have in common. A real-time vignette-building capability is being developed to process incoming live data, such as Blue Force Tracker (which tracks the movements of friendly forces). The incoming sensor data are assembled into vignettes-so-far. Since the vignette-so-far describes the evolving situation, the comparison of the vignette-so-far to the library of vignettes results in retrieval of lessons learned relevant to the developing situation; in other words, just-in-time lessons learned. This paper describes the development and capabilities of this research prototype, as well as reviewing related research and recommending steps for further development of a just-in-time lessons learned capability.

ABOUT THE AUTHORS

Laurie Waisel is a Principal Technical Director at Concurrent Technologies Corporation (*CTC)* and Project Manager for the DARPA-funded National Applied Software Engineering Center (NASEC), which conducts research in military decision aids, ontology development, case-based reasoning, and narrative understanding (i.e., stories). She also is Project Manager for the RIPPLE (Reactive Information Propagation and Planning for Lifelike Exercises) suite of applications for designing and executing live training exercises. Dr. Waisel has overseen the development of both programs since their inceptions. Her research interests also include training and exercise support, possibilistic biohazard risk assessment, and epidemiology.

Wes Regian is a Principal Knowledge Engineer at *CTC*. Dr. Regian is responsible for knowledge acquisition and engineering for NASEC. His current project tasks focus on formalizing the cognitive processes whereby field commanders build mental models of evolving battlefield situations. Dr. Regian has 25 years of experience researching and developing technology-based training systems and performance support systems for the federal government.

Chad Davis is a Software Engineer at *CTC*. Mr. Davis works on multiple Government contracts providing application development, testing, and program documentation working on teams that are solving problems, reducing costs, and improving the efficiency of operations.

Kamal Gella is a Principal Director at *CTC*, where he leads a group of software/system engineers in support of decision support, education and training, and case management systems. Mr. Gella has an M.S. in Computer Science from the University of South Carolina and is currently pursuing a Ph.D. in Computer Science from Robert Morris University.

Just in Time Lessons Learned: Timely Retrieval of Operational Vignettes

Laurie Waisel, Wes Regian, Chad Davis, Kamal Gella
Concurrent Technologies Corporation
Johnstown, PA
{waisel, regianj, davisc, gella}@ctc.com

THE PROBLEM WITH LESSONS LEARNED

Commanders cannot easily access lessons learned (LL) during operations, which is when those lessons are most needed. Furthermore, even if LL were easy to access, the commander does not have time to search for the right LL in the midst of an operation (Aha, Weber, Muñoz-Avila, Breslow, & Gupta, 2001).

The commander needs to receive relevant LL when they are needed—that is, at critical decision points. These lessons must be directed to commanders during the course of their normal activities, without burdening them with the tasks of search and retrieval. (Johnson, Birnbaum, Bareiss, & Hinrichs, 2000). We use the concept of Just-In-Time (JIT), which originated in the manufacturing industry (Dyck & Johnson, 1988), to address the military need for the commander *to get the right information in time to make the best decision in a critical situation.*

This paper will describe the ongoing development of the Military Analogical Reasoning System (MARS) (Waisel, 2009; Waisel, DeSmedt, & Regian, 2009). Our goal is to create a system that will rapidly identify stories or vignettes—from a pre-existing corpus of "battle stories"—that will be immediately helpful in the currently evolving situation. The goal of the MARS project is to be able to identify and retrieve similar stories, stories that are similar in ways that matter in a currently evolving situation.

Throughout this paper, we use the terms *story*, *vignette*, *situation*, and *scenario* interchangeably. All of these terms refer to a narrative account of a military operation. The account may be of an actual operation that has taken place—a historical story—or it may be of a notional operation that has been developed for training purposes—a training scenario. We use the term *significantly similar* to describe a pair of stories with important structural and semantic similarities.

A NEW APPROACH TO LESSONS LEARNED

We begin with an example of a military story.

Consider this story, named BOXES: "An informant reports that he saw heavy wooden boxes being unloaded at a house in the Baghdad neighborhood of Shula. Company C raids the house and confiscates 100 AK-47s and 10 RPG-7s."

How would we find stories that bear significant similarity to BOXES, even if they look different on the surface?

Consider a second story (PACKAGES): "Members of a foot patrol saw multiple large packages being delivered to a house in Mosul. Company A raids the house and confiscates 300 cases of cigarettes, stolen and slated for sale on the black market." For reasons which will be explained later in the paper, MARS would return this story as a potential match.

What are the significant similarities between these two stories? Let's look at the events as they might be represented by a military operations language. In BOXES, the events are Transfer, Report, Raid, and Confiscate. In PACKAGES, we have Transfer, Observe, Raid, and Confiscate. In both stories, there was a transfer of material objects, a report on or a direct observation of the transfer, a raid on the location where the transfer was observed, and confiscation of contraband. Had we searched on keywords such as boxes, AK47s, Baghdad, weapons, etc., we would not have found the cigarette delivery story. If we had searched on the words raid or confiscate, we would have gotten many stories about raids and confiscations that had nothing to do with the transfer of material objects. Searching on event patterns rather than on keywords allows us to identify stories that are similar to each other in important structural and semantic ways. The LL is then derived from the pattern of matching events that the stories share.

The new approach referred to in the title of this section contains the following innovations: (1) rather than storing LL, MARS stores the vignettes from which the LL can be derived in real time when needed; (2) during a developing situation, incoming operational data can be captured live and a developing story inferred from those data; and (3) an immediately relevant LL can be derived by finding event patterns that the two significantly similar stories have in common.

In the next section, we delve into the concept of Just-In-Time Lessons Learned (JITLL). We examine first the component concepts JIT and LL, and then the merging of those concepts into JITLL.

Just-In-Time Lessons Learned (JITLL)

Just-In-Time (JIT)

JIT technology originated in the field of manufacturing. By the 1970s and 1980s U.S. manufacturers had noticed that Japanese firms far outranked U.S. firms in quality and productivity. Industrial engineers studied how Japanese firms had achieved such success and encountered an interdependent set of tenets that collectively are known as Kanban (Dyck, *et al.*, 1988). Kanban is largely based on two principles: minimization of inventory and Jidoka. The Japanese word Jidoka has multiple translations, including "automation with a human touch" and "quality at the source." These two Kanban principles have been thoroughly integrated into the philosophy, standards, and procedures of manufacturing and quality control in the United States. The inventory minimization principle became known as JIT manufacturing. JIT is a "pull" system; it minimizes inventory by manufacturing components only when they are required to fill customer orders. In other words, incoming customer orders cause the components needed for filling those orders to be pulled into inventory. For the JITLL, we want to "pull" operational data from the environment so we can generate a LL that is relevant to what is happening right now.

Use of the JIT concept has spread far beyond manufacturing. Software engineering has JIT software compilation (Aycock, 2003). Prognostic systems can be designed to perform maintenance tasks just prior to failure (Ball, et al., 2008). Educators use feedback from out-of-class coursework to drive JIT teaching in the classroom; students' responses to assignments determine how new material is presented in the classroom (Bailey & Forbes, 2005; Davis, 2009; Novak, Gavrin, Christian, & Patterson, 1999). Just-in-time training (JITT) is used extensively in corporate training programs (Mosher, 2005; Perkins, et al., 2007; Morrison & Jaworski, 2008; Dutton, 2009) because traditional training methods do not reach users at the time that they need the information. JITT for the military is needed to keep Department of Defense (DOD) training on the cutting edge (Joint Chiefs of Staff, 2000).

Lessons Learned (LL)

Now let's step away from JIT and look at LL. Harvesting and disseminating LL are essential elements of military training (R. O. Weber & Aha, 2002, 2003). The DOD's definition of LL is "… a technique, procedure, or practical workaround that enable[s] a task to be accomplished to standard based on an identified deficiency or shortcoming" (Joint Chiefs of Staff, 2000, p. GL-3). In other words, a LL may be thought of as a contingency plan: what to do under a certain set of circumstances to overcome an obstacle or resolve a problem. The definition of LL used by the American, European, and Japanese Space Agencies addresses many of the contextual factors that add value to LL: being based on experience; being inclusive of both successes and failures; having a real or assumed significant impact on operations; being technically verifiable (and verified); and being applicable to a specific design, process or decision that improves the likelihood of future success (R. Weber, Aha, & Becerra-Fernandez, 2001). LL systems are built to preserve crucial knowledge resident in an organization's experts or in organizational culture. The space agencies' definition helps distinguish LL from other organizational knowledge artifacts. Incident reports, for example, report only on failures. Alerts focus on broad concerns about safety, policy, and other industry-wide issues. Best practices and corporate memories are not necessarily based on experience.

A retired Marine Corps General told one of the authors about a lesson he learned in Haiti in the late 1980s. Riots had been occurring frequently and in no discernable geographic pattern. U.S. personnel, after trying for weeks to figure out how to predict the riots, eventually realized that prior to a riot, tires were piled up at the location where the riot would be held, and then set on fire during the riot. So the LL was: To predict where a riot may occur, look for piles of tires in the street. This LL is based on experience, had a significant impact on operations, and was applicable to decisions and procedures that addressed stability operations.

Weber *et al.* (2001) have identified the stages of the LL lifecycle process as lesson collection, lesson verification, lesson storage, and lesson dissemination. The organizational goal of the process is to improve processes and performance. Honeywell's Military Segment Vice President Jerry Wellman (2007) has categorized the capture and deployment of LL into four approaches: culture, old pros, processes, and archives. *Culture* is the implicitly understood but not directly expressed set of behaviors, operating principles, and social norms that are known by members of the culture. Culture is tacit knowledge; that is, knowledge which is intuitive, unarticulated, and very possibly nonverbalizable (Hedlund, 1994). *Old pros* are veterans who have accumulated years of experiential knowledge about the organization's products, processes, environment, and capabilities. *Processes* are LL that have solidified into policies, procedures, and standards. *Archives* for capture, retrieval, and dissemination of LL often are developed when organizations become aware that valuable LL are being overlooked.

Although Wellman's article is oriented toward a corporate audience, his military background is evident; each of his four approaches has a direct correlate in the military world. In boot camp, recruits absorb military culture—the tacit knowledge—in addition to learning doctrine and procedures. Wellman's description of old pros is synonymous with the definition of military subject matter experts. Wellman's processes and military doctrine are one and the same. Finally, the military has many organizations across the services devoted exclusively to archiving LL.

One of the largest and best-known military LL archives is the Center for Army Lessons Learned (CALL) (Center for Army Lessons Learned, 2008). CALL has a public archive (CALL Public Archives, n.d.), which allows users to search four different libraries of LL: Combined Arms Research Library, Military History Institute papers, Military Review English Edition, and Military Review Spanish Edition. Each library can be expanded into a hierarchy of topics and subtopics. A separate password-protected archive is limited primarily to military personnel.

Just-In-Time Lessons Learned (JITLL)

The concepts of JIT and LL have been merged to create the concept of JITLL. The use of JITLL is well-established in the Department of Energy's (DOE) Office of Health, Safety and Security (DOE, 2008; Grant, 2006; Wu, 2006), making it possible to provide immediate feedback and timely guidance on safety issues. The U.S. Army regards JITLL as an essential element in producing competent future military leaders (Horey, et al., 2004). In 2001, Aha, Weber, Muñoz-Avila, *et al.*, used case-based reasoning to develop a method for integrating LL with organizational decision processes. In the following two years, Weber and Aha developed an intelligent military Lesson Elicitation Tool (2002) and explored methods for the timely delivery of military LL (2003).

Weber *et al.* (2001) note that in order to identify LL in a timely manner, they must be pushed to the right person at the right time, and that a software-controlled process is needed to identify immediately relevant LL and push them to the right person. The authors of this paper believe that what is needed is a system that both pushes and pulls. The system needs to pull contextual data in order to determine the right LL, and then that LL needs to be pushed to the user.

JITLL is related to just-in-time training (JITT), which is used extensively in corporate training programs (Mosher, 2005; Perkins, *et al.*, 2007; Morrison, *et al.*, 2008; Dutton, 2009), because traditional training methods do not reach users at the time that they need the information.

In MARS, we envisioned the following steps for a JITLL system:

1. Record stories that embed LL.
2. Capture (pull) real-time data in a Tactical Operations Center (TOC) and assemble data into a "story-so-far."
3. Analyze stories from the corpus in real time to find storie(s) analogous to the evolving story-so-far.
4. Push significantly similar stories to user for selection of best match to story-so-far.
5. Extract a LL common to both the story-so-far and the story selected from the corpus.
6. Push this LL to the user.

The following points are worth noting:

- Unlike other LL systems, MARS collects and accumulates stories, not LL.
- MARS combines pulling live data from the current situation, pushing analogous stories to the user for review, and allowing the user to select the story that seems most relevant. Only then is a LL generated and pushed back to the user.
- What MARS is really comparing and analyzing are *episodes* from stories rather than the stories in their entirety.

An episode is a sequence of events within a story that embodies a meaningful chunk of action. MARS compares the episodes from a pair of stories to determine if the stories are significantly similar *based on that particular episode*. We might speak of significantly similar episodes rather than significantly similar stories but for this crucial distinction between episodes and stories: while a LL system could deal with episodes alone, it is the episode *within the context of its story* that cues the human user to identify the most relevant analogous story. We will elaborate on the method of identifying and analyzing the significantly similar episodes in the sections on structure mapping.

Recalling the point made by Weber *et al.* (2001) about the need to push LL to the right user a the right time, the second bullet point illustrates the value of a push-pull LL system. Without the pull aspect, the system has no information about the current situation except for what is input by the user—an activity that we want to minimize.

The significance of collecting stories rather than LL is that each lesson generated is specific to the grouping of the story-so-far and a story from the corpus. Why use the grouping of story-so-far and story to generate the LL? First, because the story-so-far may have enough information to identify significantly similar stories in the corpus, but not enough information to generate a useful LL. Furthermore, the point of generalizing a LL from a pair of stories is to capitalize on and encapsulate the relationship pattern common to both stories. The story-so-far might have one relationship pattern in common with one story from the corpus, and a different relationship pattern in common with another story from the corpus.

In the next section, we discuss the research behind the implementation of the MARS JITLL system, and the techniques by which the implementation was carried out.

ANALOGICAL REASONING

At the heart of our method for finding significant similarities between stories—and thereby generating LL—is the process of *analogical reasoning*. The methodology we use to implement analogical reasoning is called *case-based reasoning* (CBR), and the type of CBR we employ is called *structure mapping*. Finally, the model we use to execute structure mapping is called *Many Are Called / Few Are Chosen (MAC/FAC)*. This section discusses analogical reasoning, CBR, structure mapping, MAC/FAC, and their applications in MARS.

Analogical reasoning is the process of mapping between two things, termed analogs, to determine what one analog—the base—can tell us about the other, the target (Gentner, 1983). Two things—situations, processes, entities—are analogous if they share a common pattern of relationships, even if the elements comprising the pattern are different for each analog. Analogical reasoning uses the information given for base and target to generate plausible inferences about the target. Analogical reasoning is therefore a type of inductive reasoning (Holyoak, 2005).

Analogical learning, related to analogical reasoning, can be viewed as a combination of both inductive and deductive reasoning. In learning by analogy, the learner acquires a new concept by modifying a known similar concept. This process can be decomposed into two steps, one inductive and one deductive: first the learner determines general characteristics of the new concept by drawing inferences (inductive reasoning) from supplied facts and observations; then, the learner uses deductive reasoning to derive the features expected of the new concept (Michalski, 1986). The "goodness" of the inference may be assessed by how well the features deduced from an induced model of the new concept match the features observed in the new concept. This inductive/deductive model of analogical learning is an example of a mental model, the study of which has produced its own large body of research (See, for example, Nersessian, 2008; Nersessian, in press; Harmon & Nersessian, 2008; Cowan, 1986; Gentner, 1983; Johnson-Laird, 1983; Lohse, 1991; Norman, 1983; Polk, 1996). Visualization of mental models has also been explored (Crapo, Waisel, Wallace, & Willemain, 2002). Delving into mental models is beyond the scope of this paper, so we return now to analogical reasoning.

Implementation of analogical reasoning can take many forms. Structure mapping, used in MARS, relies on structural constraints alone. By contrast, the Analogical Mapping by Constraint Satisfaction (ACME) model applies multiple types of constraints—structural, semantic and pragmatic—in a cooperative algorithm to identify analog pairs. A cooperative algorithm is a procedure that simultaneously satisfies each of a set of interacting constraints (Holyoak, 2005; Holyoak & Thagard, 1989). The Learning and Inference with Schemas and Analogies (LISA) model uses multiple types of constraints as ACME does. Unlike ACME, LISA includes a semantic analysis, associating propositions (predicates and their arguments) with semantic primitives (Hummel & Holyoak, 1997).

Developing Analogical Reasoning for MARS

The analogical reasoning in MARS was developed in stages. We began with what we call *simple story comparison*, which answers the question "What does this story tell me about that story?" Following Gentner, we refer to "that story" as the target story, and to "this story" as the base story. The comparison is described as simple because it compares only two stories, both stored rather than live. The next stage of development, *multiple story comparison,* enabled searches through a (potentially large) corpus of stories to answer the question "What can this corpus of base stories tell me about the target story?"

Finally, we move on to *live story comparison*, searching the corpus for stories that are similar to a *story-so-far*—a story being built in real time as data become available to the TOC. Live story comparison answers the question, "What do these base stories tell me about this unfolding target story?" To summarize, simple story comparison compares one story with another and determines if there are significant similarities between them. Multiple story comparison compares a single target story with multiple base stories. Live story comparison is the same as multiple story comparison, except the target story is a live story-so-far being inferred in real time from incoming data. All three types of analogical reasoning are accomplished by drawing analogies from structured, formal representations of stories.

Case-Based Reasoning (CBR)

Case-based reasoning (CBR) is a method for implementing analogical reasoning. CBR makes use of previous situations known as *cases* to solve a new problem. Cases are useful to a new problem if the situations they describe are similar to the circumstances of the new problem. CBR may suggest a way to solve the new problem, recommend a way to adapt the case to the new situation, or warn of possible failures.

A case is roughly synonymous with a situation, although a situation must meet the following three requirements in order to qualify as a case: the characterization of the situation must include the situation's context; the situation must represent an experience; and, the situation must teach a LL fundamental to achieving the goals of the reasoner (Kolodner, 1993).

Like analogical reasoning, CBR entails reasoning by example—mapping the elements of a known thing, the base case, to the elements of an unknown thing, the target case. In MARS, the base case is the story from the corpus, and the target case is the story-so-far that has been assembled from the live operational data.

Why use CBR to compare military stories? Expert systems, which use reasoning methods similar to CBR, have long been used for solving medical and engineering problems. An expert system simulates the problem-solving procedure of an expert and then applies that problem-solving procedure to a new problem from the same domain. Military operations analysts know that expert systems could be useful to commanders, but expert systems have a disadvantage: they give an answer rather than guiding the user through a decision making process. Systems that perform the latter function are called decision support systems. CBR is often used to develop decision support systems for domains in which an automated answer is undesirable. Dupuy (1988) and Goodman (1988) laid the groundwork for using CBR to understand, plan, and learn from military situations.

The difficulties in doing CBR with military history are legion. They include the "complexity of the multifarious, interacting, interrelated problems" that face a combat commander; the fact that every battle, no matter how similar it appears to another, has its own unique characteristics; the idiosyncratic unpredictable behavior that humans are prone to; and the lack of any living experts familiar with all the varieties of combat circumstances (Dupuy, 1988, p. 125).

As we implemented CBR for MARS, we addressed (but do not claim to have surmounted) some of these obstacles with our comprehensive military ontology. Most actions important to commanders are included in the ontology's hierarchy, thus helping tame the specter of unpredictable behavior. In addition, a comprehensive military ontology may soften the sharp-edged uniqueness of each situation by providing a framework for classifying the set of events and relationships that comprise a situation. By using CBR, we have chosen to trade off the richness of the full human experience (via text narrative) for the ability to represent situational events in a structured way that accommodates powerful analysis.

Structure Mapping

The type of CBR methodology we use in MARS is called structure mapping. Gentner's (1983) structure mapping theory asserts that "analogy is characterized by the mapping of *relations* between objects, rather than *attributes* of objects, from base to target…." (p. 168, emphasis added). Gentner also notes that "psychologically, the representation must be chosen to model the way people think about the domain" (p. 157). These declarations by Gentner formed the core of our effort to develop a system for identifying military vignettes that are significantly similar to each other.

Many Are Called / Few Are Chosen (MAC/FAC) for Structure Mapping

Gentner and Forbus's (1995) MAC/FAC is a model for similarity-based reminding that employs structure mapping. MAC/FAC attempts to capture the cognitive event of being reminded of something. Recall that the objective of MARS is to be able to retrieve stories from a corpus that are similar to a live situation that is currently developing. MARS utilizes the MAC/FAC model in order to answer the following question: if I were the story corpus, which story would this real-time evolving situation remind me of?

In MAC/FAC, there is a trade-off between computationally cheap but not especially discriminating, and computationally expensive with finer-grained selection and deeper reasoning. First, the fast non-structural MAC algorithm returns a subset of items worthy of further investigation, and then the slower and more in-depth FAC algorithm does a detailed examination of that subset using structure mapping.

The current implementation of MARS does “All Are Called” (AAC) rather than MAC. The corpus is still small and military operational knowledge is a closed domain; therefore, we have not found it necessary to use the MAC algorithm. MAC is used to reduce the number of potential stories of interest that will go through the FAC procedure. As the story corpus grows, however, we will introduce a *weak AI method* (Coppin, 2004; Michalewicz, 1994)—a method that makes few assumptions about the problem domain and therefore has wide applicability—to perform MAC.

MARS Structure Mapping General Method

MARS’ structure mapping method looks for similarities between stories in the patterns of the relationships between events in each story. The algorithm incorporates a priority value for each event, the priority indicating the criticality of the event to the story’s outcome or LL. The similarity metric is based on the analysis and mapping of high- and medium-priority events within each story. Consider the following vignette, called FOX1: “In Viet Nam, a US tank platoon is moving toward Objective Fox, which is a tunnel complex being used by the Viet Cong to store weapons. The platoon encounters a minefield in its path, and the commander decides to go around the minefield. This decision brings the platoon into a valley with a large hill overlooking the valley. The platoon is then ambushed by a Viet Cong platoon positioned atop the overlooking hill.”

The important events in FOX1 can be represented in the following manner:

CAUSE (DEPLOY (MINE, ROUTEA), DIVERT (TANKS))

Now consider a second vignette, FOX2: “In Iraq, a foot patrol conducting reconnaissance moves toward Objective Fox, which is an insurgent safe house. The unit has a Chemical Agent Monitor, which can detect biological and chemical hazards. It alerts the commander to the presence of gas in the soldiers’ path, and the commander decides to go around the hazard. This decision brings the foot patrol down a street with several two- and three-story buildings that appear deserted. The soldiers are then ambushed by two insurgents positioned atop one of the buildings with rockets and machine guns.”

The important events in FOX2 can be represented this way:

CAUSE (DEPLOY (GAS, ROUTEB), DIVERT (SOLDIERS))

The structure mapping algorithm identifies these two vignettes as potentially analogous because both contain the following string of events: CAUSE, DEPLOY, DIVERT.

MARS Structure Mapping Algorithm

After identifying a pair of potentially analogous stories, MARS calculates the similarities as shown in Table 1. To illustrate the structure mapping algorithm in MARS, we will look at two simplified stories: MINE and GAS. The MINE story has the following events and arguments: Move To Contact() → Discover(Minefield) → Disrupt(Wedge Formation) → Encounter Near Ambush() → React to Enemy Direct Fire(). The GAS story has a slightly different set of events and arguments: Move To Contact() → Discover(Razor Wire) → Disrupt (Slant Formation) → Encounter Near Ambush() → React to Enemy Direct Fire().

Table 1 illustrates the similarity algorithm and walks the reader through an example. The Most Recent Common Ancestor (MRCA) is the first node that two classes have in common when traversing up the hierarchy from those classes. Figure 1 depicts two parts of a notional hierarchy illustrating the relations that determine the result of step 2.1 in Table 1. In the part on the left (blue), the MRCA of Mine and Gas is one level up: Obstacle. In the part on the right (orange), reaching the MRCA for Slant and Wedge requires going two levels up the hierarchy, to Tactical Formation.

Table 1. The similarity algorithm and an example.

ALGORITHM STEPS	SIMILARITY CALCULATIONS USING ALGORITHM STEPS		
1. Identify Potentially Matching Stories			
1.1. Find sequences of high- and medium-priority events in each story 1.2. Look for sequences (runs) of events that are identical in base and target 1.3. Any two stories that contain identical sequences (runs) of three or more high- or medium-priority events are considered a potential match	**Event Sequences of modified MINE and GAS stories** **MINE:** Discover(Mine) → Disrupt(Wedge Formation) → Near Ambush() → React to Enemy Fire() **GAS:** Discover(Gas) → Disrupt(Slant formation) → Near Ambush() → React to Enemy Fire()		
2. Calculate similarity between stories for the arguments and effects of each event in the run			
	Discover(Obstacle) → Disrupt	Disrupt(Tactical Formation) → Near Ambush	Near Ambush() → React to Enemy Fire
2.1. Argument similarity for each event: one level to MRCA = 0.75; two levels to MRCA = 0.5 = `scoreEvArg`	0.75 because arguments are one level from MRCA	0.5 because arguments are two levels from MRCA	0 because event has no arguments
2.2. Causal similarity for each event = `scoreEvCau`	1 because following event same in both stories	1 because following event same in both stories	1 because following event same in both stories
2.3. Add `scoreEvArg` and `scoreEvCau` = `scoreEvRaw`	0.75 + 1 = 1.75	0.5 + 1 = 1.5	0 + 1 = 1
2.4. Weight each event's `scoreEvRaw` by 1.0 for high-priority events; by 0.75 for medium-priority events = `scoreEvWtd`	1.75 * 1 = 1.75	1.5* 1 = 1.5	1 * 0.75 = 0.75
2.5. Divide each event's `scoreEvWtd` by the number of arguments in that event plus 1 = `scoreEv`	1.75 / 2 = 0.875	1.5/ 2 = 0.75	0.75/ 1 = 0.75
3. Calculate the similarity between stories from the similarities for the component events			
3.1. Number of points that would be earned if arguments of events in each story were identical across stories = `scoreEvMax`	[(1+1) * 1] / 2 = 1	[(1+1) * 1] / 2 = 1	[(0+1) * 0.75] / 1 = 0.75
3.2. Sum `scoreEvMax` over all events = `scoreEvMaxSum`	1 + 1 + 0.75 = 2.75		
3.3. Sum `scoreEv` over all events = `scoreEvSum`	0.875 + 0.75 + 0.75 = 2.375		
3.4. Divide `scoreEvSum` by `scoreEvMaxSum` = `scoreFINAL`	2.375 / 2.75 = 0.864 = 86.4% similarity		

The algorithm first looks for matching sequences of events in the two candidate stories. Having found such a sequence, it then calculates component similarities for each event in the sequence. Finally, the sum of the component similarities is divided by the sum of the highest possible component similarities to obtain the similarity quotient, which in this example is 86.4%.

.

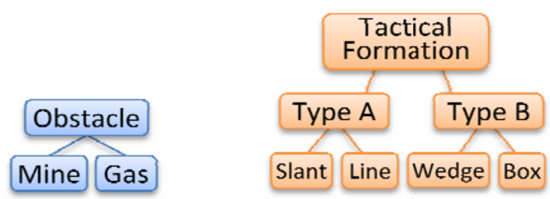

Figure 1. Finding the Most Recent Common Ancestor

Equation 1 is used for calculating both the final score for each matching event (`scoreEv`) and the maximum possible score for each event (`scoreEvMax`); n is the number of arguments in that event.

$$\frac{\left(scoreEvArg + scoreEvCau\right)\times weight}{n+1} \qquad (1)$$

The difference between the calculations to get `scoreEv` and `scoreEvMax` is that for `scoreEv`, events with differing arguments are weighted accordingly, and for `scoreEvMax`, the arguments are assumed to be identical, so the weight is always 1.

The algorithm has been implemented with the Java SimPack Library (Java SimPack, 2008) a library designed for measuring intra-ontology similarity. SimPack offers multiple methods for measuring similarities in ontologies. Besides the ontology distance method that MARS uses, SimPack has information-theoretic measures, vector-space measures, and full-text similarity measures, all of which are based on well-known research into text and concept similarity.

INSIDE MARS

Story Representation for the Military Domain

In order to work in the structure mapping algorithm, the analog stories needed to be represented so that patterns of relationships could be mapped between stories. Also, per Gentner (1983, pp. 157, quoted earlier), we required that the language of the representation be intuitive for commanders. Accordingly, a story in MARS comprises a series of events as military actions, including the five Ws information: who did it, what it was, when it happened, where it happened, and why it happened. We added a sixth W representing the object of the action: to whom it was done. The patterns that were created and mapped comprise sequences of three or more of the same events.

To represent stories in commanders' language, we turned to the gold standards for military nomenclature. The Universal Joint Task List (UJTL) is a comprehensive hierarchical listing of the tasks that can be performed by the Joint Staff, Services, combatant commands, Joint organizations, and combat support agencies that are answerable to the Chairman of the Joint Chiefs of Staff (JCS, 2008). The UJTL is used not only by the U.S. Armed Forces, but also by several other countries and international military organizations such as NATO. These operational tasks become events or actions in MARS.

We had to represent objects as well as tasks. The Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) is a NATO-approved set of data elements, entities, and relations that describe the information exchange requirements within military operations (DOD, 2006). In MARS, the JC3IEDM elements become objects and entities.

We developed two hierarchical data models for MARS, one for events, using the UJTL, and one for entities, using the JC3IEDM. These data models became the backbone of the comprehensive military ontology developed later.

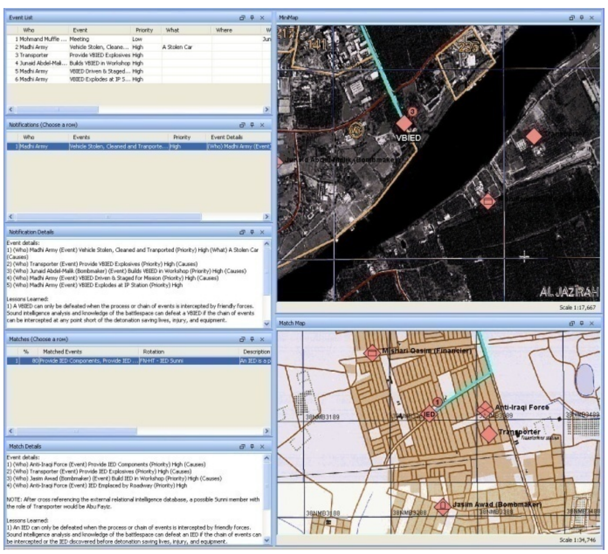

Figure 2. The story-matching interface.

MARS Interface

Figure 2 depicts the main interface for MARS, in which one story is selected as target, and multiple base stories can be examined. On the left side of the screen are panels containing (top to bottom) the Event List, Notifications, Notification Details, Matches, and Match Details. The right side shows maps for both the target story (top) and the base story (bottom). After the user identifies the target story (or after a target story is automatically acquired from message traffic as a story-so-far), MARS displays the target story's events and properties in the Event List (top left).

Table 2 contains some of the information that appears in the Event List. Each row in the Notifications panel displays a sequence of story-critical events in the target that can be compared to critical events in a base story from the corpus. After the user selects one of the rows, the Notification Details panel displays the event list, LL, and narrative description for the story that contains the selected event sequence. The Matches panel displays episodes from different stories that contain the sequence of events selected in the Notifications panel. Selecting one of the match rows causes the details and map for that story to be displayed.

Table 2. Event list and properties.

Event List from Minefield Fix Vignette					
#	Event	Priority	What	Causes	When Who Where
1	Move to Contact	Med		Discover	0040-A/63 AA- Rte A
2	Discover	Med	Mine Field	Disrupt	0207-A/63 AA-Rte A
3	Disrupt	Med	Wedge	Discover	0338-A/63 AA-Rte A
4	Discover	High	Hidden Mine Fields	Fix	0452-A/63 AA-Rte A
5	Fix	High		Breach	0605 -A/63 AA-Rte A
6	Breach	High		React to Enemy	0725-A/63 AA-Rte A
7	React to Enemy Direct Fire	Med		Medical Evacuation	0832-A/63 AA-Rte A
8	Medical Evacuation	Low		Move to Contact	1030-A/63 AA-Rte A
9	Move to Contact	Low			1210-A/63 AA- Rte A

MARS Story Input

Stories, or situations, can be entered into MARS in one of several ways. The interface for manual input is shown in Figure 3. Data can also be imported from a database or captured live by parsing incoming message data.

In the story input interface (Figure 3), the user can select from a palette of actors (i.e., units) and events (i.e., military operations) to build a story. The main sections of the interface are the map (top left), timeline (bottom left), palettes (top right), and properties (bottom right). The palettes are supported on the back end by a hierarchical representation of military operations (and objects) built on the UJTL and the JC3IEDM. Properties of a selected event or object are displayed in the properties window (bottom right).

Figure 4 shows the top level of the hierarchy of events available to the user for insertion into the story. A small part of the full expansion of the hierarchy is shown in Figure 5.

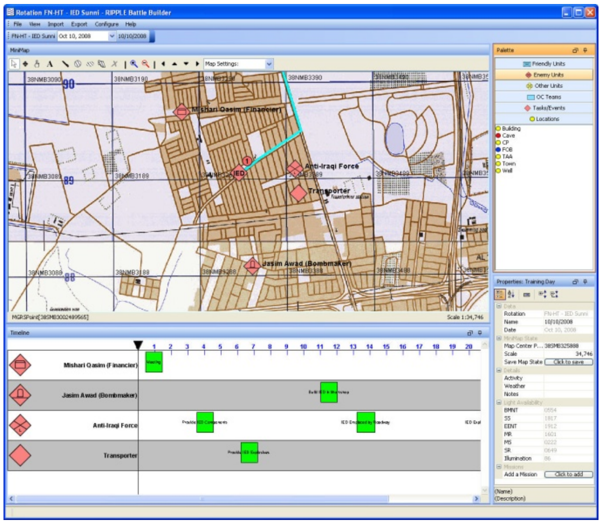

Figure 3. The story input interface.

Figure 4. The top level of the hierarchy.

Architecture

Figure 6 is a diagram of the MARS architecture. It depicts how the Structure Mapping Engine interacts with external data from the TOC and the stories corpus, as well as the Ontology and the Lessons Learned Generator.

ONTOLOGY

Ontologies are developed for many reasons, from categorizing websites to categorizing products to biology taxonomies (Gella, J. Wesley Regian, Waisel, DeSmedt, & Kovacs, 2009). An ontology is a formal representation of a set of concepts in a domain and the relationships among those concepts. It defines a common vocabulary for researchers in that area. A concept or abstraction hierarchy is similar to an ontology, but an ontology is a more complete representation of the knowledge in a domain. Differences between an abstraction hierarchy and an ontology include these: an ontology may define properties as well as relationships; may be paired with a set of class instances (i.e., actual data that are categorized by the abstractions defined in the ontology); and is not limited to hierarchical relationships. Noy and McGuinness (2001) have written an excellent introduction to ontology development.

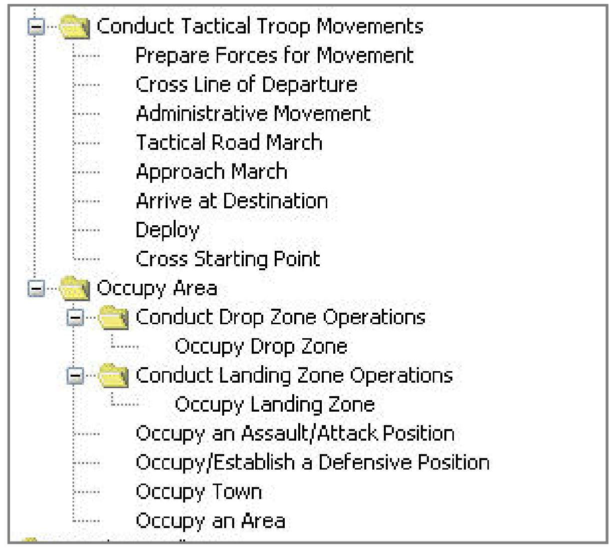

Figure 5. A small part of the large hierarchy of events.

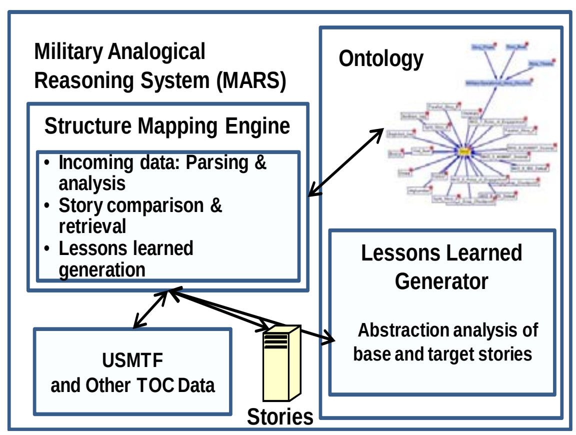

Figure 6. MARS architecture.

Comprehensive OWL Military Ontology (COMO)

As described earlier, the UJTL and JC3EIDM form the backbone of our comprehensive OWL Military Ontology (COMO). OWL is the widely-used Web Ontology Language. Beginning with the doctrinal military action and object hierarchies, we added concepts and properties for stability and support operations and human terrain activities. Among other innovations, we developed a

Roles concept, which allowed us to identify the role that an object played in a given situation: a hammer can be a tool or a weapon, depending on context. In MARS, we primarily make use of COMO's abstraction hierarchy; however, in related prototypes, we utilize concept properties and instantiations of the ontology (a pairing of the ontology with instance data for analysis of a specific situation).

Figure 7 and Figure 8 show two very small fragments of COMO's concept hierarchy. On the left in each figure is a hierarchical representation, and on the right is a hyper-tree visualization. Figure 7 illustrates how the roles concept is used in the ontology. Figure 8 contains the military operational story structure.

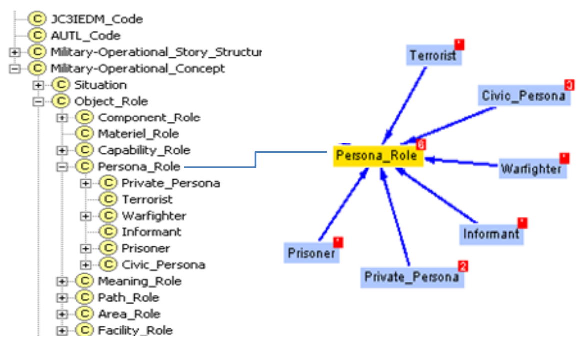

Figure 7. How the roles concept group is built.

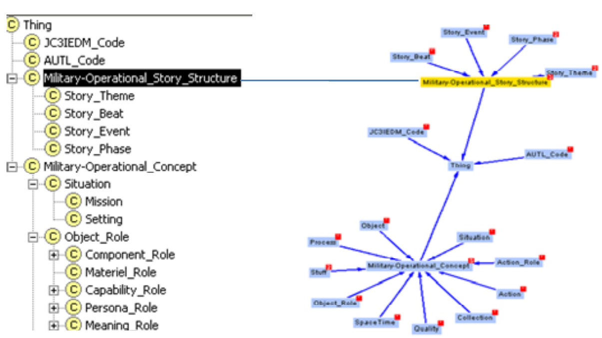

Figure 8. The military operational story structure.

INNOVATIONS AND CONCLUSION

In this section, we describe the technical innovations in MARS and conclude with a summary of the paper, a brief mention of challenges we have overcome and challenges we still face, and plans for future work.

Innovations

Comprehensive Military Ontology

COMO is the first military ontology that incorporates the UJTL and JC3IEDM. COMO's frame architecture enables us to enhance the ontology's realism by adding complex properties and relations. COMO's broad applicability and comprehensiveness have resulted in its being used in several different prototypes. In DARPA's URGENT (Urban Reasoning and Geospatial Exploitation Technology) program, COMO was used to calculate answers to tactical questions (e.g., "what's the shortest route from A to B that will keep me under concealment?") based on a labeled 3D model of an area. A *CTC* Independent Research and Development (IR&D) project used COMO to develop the capability for MARS to search external databases, including live feeds, for missing information.

Live Story Capture and Comparison

We are developing the capability to capture live data from TOC message traffic. The data are parsed and constructed into a story-so-far. While building a structured situation representation of live incoming data is a difficult problem, we have the advantage of being able to leverage the highly structured formats typical of incoming TOC data. We have developed a parser for U.S. Message Traffic Format (USMTF), focusing initially on messages addressing Nuclear, Biological and Chemical (NBC) situations. Each time an event is added to a story-so-far based on analysis of incoming data, that version of the story-so-far will be the target for a similarity sweep through the corpus of base stories. As the situation evolves, the contents of the Notifications panel will change as differing or growing sequences of critical events emerge. As the Notifications change, the Matches panel will also change, since the groups of critical events in the target story will be evolving.

Harvesting Immediately Relevant Lessons Learned

By combining the story matching capability with the live story capture capability, we can harvest LL that are pertinent to the evolving situation. MARS accomplishes this by doing a semantic analysis on COMO's abstraction hierarchy to produce a LL that is relevant to both stories. Figure 9 illustrates how a LL is derived and harvested for the FOX1 and FOX2 stories.

The following paragraph describes how CAUSE(DEPLOY, DIVERT) is turned into conversational form. In this example, lowercase letters represent what has been added to the pseudocode by the narrative rule base.

The LL CAUSE(DEPLOY, DIVERT) is a Resource Description Framework (RDF) expression of the form (PREDICATE (SUBJECT, OBJECT)). To read and understand this kind of expression, we unpack it into SUBJECT OBJECT PREDICATE. So CAUSE(DEPLOY, DIVERT) would be expressed in narrative form as OBSTACLE DEPLOY ROUTE. Our rule base defaults to past tense and a narrative perspective of first person plural, so applying the rule base to the RDF expression yields "[an OBSTACLE was DEPLOYed on our ROUTE]." By the same token, the more complex expression (CAUSE (DEPLOY (OBSTACLE, ROUTE)), DIVERT) becomes "an OBSTACLE was DEPLOYed on our ROUTE] AND CAUSEd us to DIVERT." And, finally, the expression (ENEMY (CAUSE (DEPLOY (OBSTACLE, ROUTE)), DIVERT (FORCES)) AMBUSH) becomes "an OBSTACLE was DEPLOYed on our ROUTE AND CAUSEd us to DIVERT AND then the ENEMY AMBUSHed us."

The analysis that leads to the LL CAUSE(DEPLOY, DIVERT) is explained in detail in the discussion of structure mapping. Methods for transforming the pseudocode form of the LL—CAUSE(DEPLOY, DIVERT)—to the narrative form—"Beware of any obstacle that causes you to change your route, because in a similar situation, this led to an ambush"—are well-established. With additional natural language processing and the use of rule bases to add articles, tense, and narrative perspective, the LL can be expressed in a more conversational tone. The hard part of generating the LL common to two stories is identifying and structuring the key concepts of relationships.

Conclusion

In this paper, we have presented a technology for harvesting JITLL for the commander during an evolving situation. Incoming TOC data are parsed into a structured situation-so-far that is compared with other stored situations. The algorithm for calculating similarity between situations is founded on a well-documented method for analyzing intra-ontology similarities. Analysis of the ontology's hierarchy with respect to the events in the target and base stories yields a LL common to both. Live situation awareness data transformed into a structured representation will enable identification of LL relevant to the currently evolving situation. Live derivation of JITLL will increase efficiency, enhance the ability to understand decision factors, and augment human reasoning abilities.

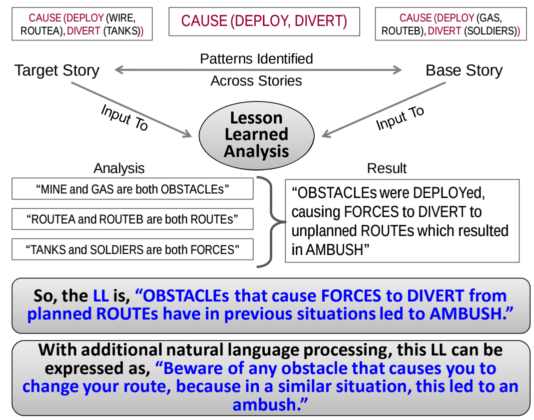

Figure 9. How lessons learned are harvested.

Challenges that we have addressed so far include deciding on a formal representation system for stories, deciding how to do the comparison between stories, and building a military ontology. Challenges that we still face include how to chunk individual stories from a stream of live data, exploring how to take in data that are not as well-structured as USMTF, building the corpus of stories, and user testing.

MARS is intended to be used by commanders and staffs at the battalion or brigade level as a real-time decision aid during a developing situation. MARS is intended to be used in the TOC, not in the field. To thoroughly test MARS for its intended use, there are three different experiments that need to be done. First, we need to test whether subject matter experts' (SME) assessment of similarity between military concepts has been captured accurately in the ontology. Second, we need to find out how close MARS' assessment of similarity between stories is to SME assessment of same. Finally, we need to test the validity of the LL extracted from the pairs of stories. We intend to model the first experiment on the experiment done by the developers of SimPack (Bernstein, Kaufmann, Kiefer, & Bürki, 2006).

Future work includes conducting these experiments, completing the live data capture feature, building up the corpus, and expanding COMO.

ACKNOWLEDGEMENTS

This work has been supported in part by DARPA Grant HR0011-07-1-0001. Many of the accomplishments reported in this paper would not have been possible without this valuable support from DARPA.

REFERENCES

Aha, D. W., Weber, R., Muñoz-Avila, H., Breslow, L. A., & Gupta, K. M. (2001). Bridging the Lesson Distribution Gap. In B. Nebel (Ed.) *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (pp. 987-992). San Francisco, CA: Morgan Kaufmann.

Aycock, J. (2003). A Brief History of Just-in-Time. *ACM Computing Surveys (CSUR) 35*(2), 97-113.

Bailey, T., & Forbes, J. (2005). Just-in-Time Teaching for CS0. *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (pp. 366-370). New York: ACM.

Ball, J., et al. (2008). AAAI Fall Symposium Reports. *AI Magazine, 29*(1), 99-104.

Bernstein, A., Kaufmann, E., Kiefer, C., & Bürki, C. (2006). SimPack: A Generic Java Library for Similarity Measures in Ontologies (Technical Report). Zurich, Switzerland: Univ. of Zurich, Dept. of Informatics. Retrieved June 19, 2009, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.8152&rep=rep1&type=url&i=0

CALL Public Archives (n.d.). [Online Database Portal]. Accessed July 9, 2009, at http://calldp.leavenworth.army.mil/call_pub.html

Center for Army Lessons Learned (2008). [Web Page]. Accessed July 9, 2009, at http://usacac.army.mil/cac2/call/archives.asp

Coppin, B. (2004). *Artificial Intelligence Illuminated*. Sudbury, MA: Jones and Bartlett Publishers.

Cowan, D. A. (1986). Developing a Process Model of Problem Recognition. *Academy of Management Review, 11*(4), 763-776.

Crapo, A. W., Waisel, L. B., Wallace, W. A., & Willemain, T. R. (2002). Visualization and Modeling for Intelligent Systems. In C. T. Leondes (Ed.) *Intelligent Systems: Implementation Techniques* (Vol. 1, pp. 53-85). Boston, MA: CRC Press.

Davis, J. (2009). Experiences with Just-in-Time Teaching in Systems and Design Courses. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (pp. 71-75). New York: ACM Press.

DOD (1999). Inhibited Red Fuming Nitric Acid Information Paper, Glossary. Retrieved June 24, 2009, from http://www.gulflink.osd.mil/irfna/irfna_taba.htm

DOD (2006). *Department of Defense Instruction Number 8260.03*. Retrieved June 11, 2009, from http://www.dtic.mil/whs/directives/corres/pdf/826003p.pdf

DOE (2008). Tank Farm Assessment Project Monthly Report for June 2008. (No. A-08-AMTF-TANKFARM-O 17). Richland, WA. Retrieved May 29, 2009, from http://www.hanford.gov/orp/uploadfiles/A-08-AMTF-TANKFARM-017.pdf

Dupuy, T. N. (1988). Military History and Case-Based Reasoning. In J. Kolodner & M. Goodman (Eds.) *Proceedings of the DARPA Workshop on Case-Based Reasoning* (pp. 125-135). San Mateo, CA: Morgan Kaufmann.

Dutton, G. (2009). Training Tech Check. *Training, 46*(1), 24-26.

Dyck, H., & Johnson, R. A. (1988). Transforming a Traditional Manufacturing System into a Just-in-Time System with Kanban. In M. A. Abrams, P. L. Haigh & J. C. Comfort (Eds.) *Proceedings of the 20th Winter Simulation Conference* (pp. 616-623). San Diego, CA: IEEE Publishing.

Gella, K., J. Wesley Regian, Waisel, L., DeSmedt, B., & Kovacs, P. J. (2009). Effective Knowledge Management Using an Ontology-Based Approach [Manuscript submitted for publication].

Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science, 7*, 155-170.

Gentner, D., & Forbus, K. (1995). MAC/FAC: A Model of Similarity-Based Retrieval. *Cognitive Science*, (19), 141-205.

Goodman, M. (1988). CBR in Battle Planning. In J. Kolodner & M. Goodman (Eds.) *Proceedings of the DARPA Workshop on Case-Based Reasoning* (pp. 264-269). San Mateo, CA: Morgan Kaufmann.

Grant, G. (2006). Just in Time Lessons Learned. Presented at the *Integrated Safety Management Best Practices Workshop*, Aurora, CO, September 12-13. Retrieved May 29, 2009, at http://www.hss.energy.gov/healthsafety/ism/workshop/Sep-06_Presentations/RichHiggins.ppt

Harmon, E., & Nersessian, N. J. (2008). Cognitive Partnerships on the Bench Top: Designing to Support Scientific Researchers. In G. Marsden, I. Ladeira & P. Kotzé (Eds.) *Proceedings of the 7th ACM Conference on Designing Interactive Systems DIS08* (pp. 119-128). New York: ACM.

Hedlund, G. (1994). A Model of Knowledge Management and the N-Form Corporation. *Strategic Management Journal, 15*(S2), 73-90.

Holyoak, K. J. (2005). Analogy. In K. J. Holyoak & R. G. Morrison (Eds.) *The Cambridge Handbook of Thinking and Reasoning* (pp. 117-142). New York: Cambridge Univ. Press.

Holyoak, K. J., & Thagard, P. (1989). Analogical Mapping by Constraint Satisfaction. *Cognitive Science. 13*(5), 295-355.

Horey, J., et al. (2004). Competency-Based Future Leadership Requirements. (Tech. Rpt. No. 1148): U.S. Army Research Institute for the Behavioral and Social Sciences. Retrieved May 29, 2009, from http://www.au.af.mil/au/awc/awcgate/army/tr1148.doc

Hummel, J. E., & Holyoak, K. J. (1997). Distributed Representations of Structure: A Theory of Analogical Access and Mapping. *Psychological Review, 104*(3), 427-466.

Java SimPack (2008). [Java Library]. Santa Clara: Sun Microsystems.

Johnson-Laird (1983). *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge, MA: Harvard University Press.

Johnson, C., Birnbaum, L., Bareiss, R., & Hinrichs, T. (2000). War Stories: Harnessing Organization Memories to Support Task Performance. *Intelligence: New Visions of AI in Practice, 11*(1), 17-31.

Joint Chiefs of Staff (2000). Joint Lessons Learned Program: Instruction. (CJCSI 3150.25A). Retrieved May 29, 2009, from http://www.dtic.mil/doctrine/jel/cjcsd/cjcsi/3150_25a.pdf

Joint Chiefs of Staff (2008). Universal Joint Task Manual. (CJCSM 3500.04E). Retrieved June 11, 2009, from http://www.dtic.mil/doctrine/training/cjcsm3500_04e.pdf

Kolodner, J. L. (1993). *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

Lohse, J. (1991). A Cognitive Model for the Perception and Understanding of Graphs. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology (*pp. 137-144). New York: ACM.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs* (2nd ed.). Berlin: Springer-Verlag.

Michalski, R. S. (1986). Concept Learning. In S. C. Shapiro (Ed.) *Encyclopedia of Artificial Intelligence* (1st ed., Vol. 1, pp. 185-194). New York: John Wiley & Sons.

Morrison, M., & Jaworski, B. (2008). Just-in-Time Executive Development. *Chief Learning Officer, 7*(11), 30-56.

Mosher, B. (2005). E-Reference: The Real Just-in-Time Training. *Chief Learning Officer, 4*(11), 17.

Nersessian, N. J. (2008). Modeling in Conceptual Change. In S. Vosniadou (Ed.) *International Handbook of Research on Conceptual Change* (pp. 391-416). New York: Routledge-Taylor & Francis.

Nersessian, N. J. (in press). Model-Based Reasoning in Interdisciplinary Engineering. In A. W. M. Meijers, D. M. Gabbay, P. Thagard & J. Woods (Eds.) *Handbook of the Philosophy of Technology and Engineering Sciences*. Amsterdam: Elsevier.

Norman, D. A. (1983). Some Observations on Mental Models. In D. Gentner & A. L. Stevens (Eds.) *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Novak, G., Gavrin, A., Christian, W., & Patterson, E. (1999). *Just-in-Time Teaching: Blending Active Learning with Web Technology*. Upper Saddle River, NJ: Prentice Hall-Pearson.

Noy, N. F., & McGuinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. (Stanford Medical Informatics Technical Report No. SMI-2001-0880): Stanford Knowledge Systems Laboratory. Retrieved April 20, 2009, from http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html

Perkins, T., Doyle, D., Wisniewski, J., Kaiser, S., Vosler, L., & Amoo, J. (2007). Great Business Training Is Just Enough, Just in Time, and Just for Me. *Community College Journal of Research & Practice, 31*(6), 537-539.

Polk, T. (1996). Reasoning Matters: Mental Models and Soar. In D. M. Steier & T. M. Mitchell (Eds.) *Mind Matters: A Tribute to Allen Newell* (pp. 401-406). Mahwah, NJ: Lawrence Erlbaum Associates.

Waisel, L. (2009). A Military Analogical Reasoning System (MARS) for Understanding Stories [Powerpoint slides]. *DARPA Workshop on Experience-Based Narrative Memory*.

Waisel, L., DeSmedt, B., & Regian, W. (2009). Representation and Analysis of Formalized Stories [White paper]. Paper presented at the *DARPA Workshop on Experience-Based Narrative Memory*, Arlington, VA, February 23-24.

Weber, R., Aha, D. W., & Becerra-Fernandez, I. (2001). Intelligent Lessons Learned Systems. *Expert Systems with Applications, 20*(1), 17-34.

Weber, R. O., & Aha, D. W. (2002). Intelligent Elicitation of Military Lessons. In Y. Gil & D. Leake (Eds.) *Proceedings of the 7th International Conference on Intelligent User Interfaces (*pp. 226 - 227). New York: ACM.

Weber, R. O., & Aha, D. W. (2003). Intelligent Delivery of Military Lessons Learned. *Decision Support Systems, 34*(3), 287-304.

Wellman, J. (2007). Lessons Learned About Lessons Learned. *Organization Development Journal, 25*(3), P65-P72.

Wu, C. (2006). ISM Vision and Workshop Objective. Presented at the *Department of Energy - Office of Health, Safety and Security: Integrated Safety Management Workshop*, Albuquerque, NM, April 24-27. Retrieved May 29, 2009, at https://hss.doe.gov/HealthSafety/ism/workshop/Apr-06_Presentations/Tue-AM1-Wu.ppt

