Intelligent Lessons Learned Systems[1]

Rosina Weber[1], David W. Aha[2], Irma Becerra-Fernandez[3]

[1]Department of Computer Science, University of Wyoming, Laramie, WY 82071
[2]Navy Center for Applied Research in AI, Naval Research Laboratory, Washington, DC 20375
[3]Florida International University, Decision Sciences and Information Systems, Miami, FL 33199
{weber,aha}@aic.nrl.navy.mil, becferi@fiu.edu

Abstract

Lessons learned processes have been deployed in commercial, government, and military organizations since the late 1980s to capture, store, disseminate, and share experiential working knowledge. However, recent studies have shown that software systems for supporting lesson dissemination do not effectively promote knowledge sharing. We found that the problems with these systems are related to their textual representation for lessons and that they are not incorporated into the processes they are intended to support. In this article, we survey lessons learned processes and systems, detail their capabilities and limitations, examine lessons learned system design issues, and identify how artificial intelligence technologies can contribute to knowledge management solutions for these systems.

Keywords: Lessons learned systems, Knowledge management, Artificial Intelligence, Case-based reasoning

1 Introduction

Lessons learned (LL) systems have been deployed in many military, commercial, and government organizations to disseminate validated experiential lessons. They support organizational LL processes, and implement a knowledge management (KM) approach for collecting, storing, disseminating, and reusing experiential working knowledge that, when applied, can significantly benefit targeted organizational processes (Davenport & Prusak, 1998). Recent studies (Fisher et al., 1998; Weber et al., 2000) have identified that, in spite of significant investments in these systems, their ability to promote knowledge sharing is limited.

Several Navy officers and contractors inspired us to investigate this topic, explaining that, while large repositories of lessons exist, their information is not being used. To gain further insight into LL systems, we reviewed relevant literature on LL processes and systems (*e.g.*, van Heijst, van der Spek, & Kruizinga, 1996; Fisher et al., 1998; SELLS, 1999; 2000; Secchi, 1999; Aha & Weber, 2000), and interviewed members of organizations that have implemented LL systems, including the Joint Center for Lessons Learned (JCLL) of the Joint Warfighting Center, the Department of Energy (DOE), the Naval Air Warfare Center, the RECALL group at NASA's

[1]Accepted in the special issue on Expert Systems/AI and Knowledge Management for the Expert Systems With Applications Journal. This special issue (Guest Editors: Yogesh Malhotra and Jay Liebowitz) will be published as Vol. 20, No. 1, to be published in January 2001.

Goddard Space Flight Center, the Navy Facilities Engineering Command, the Construction Industry Institute, and the Air Force Center for Knowledge Sharing. We also spoke with many intended users of LL systems. Based on these interviews, we learned that today's standalone LL systems are infrequently used.

To better understand the underlying issues, we developed a categorization framework for LL systems to investigate their characteristics and those of the processes that they represent. After introducing this subject in Section 2 and briefly summarizing LL system characteristics in Section 3, we detail this framework in Section 4, and provide example LL systems in each category. After examining representations for lessons learned in Section 5, we examine in Section 6 how artificial intelligence (AI) technologies may improve the design and effectiveness of LL systems.

As with any new field of research, investigations on the effective design of LL systems will identify many issues that demand further research and development. In this article, we define some of these research issues by categorizing LL systems, and by establishing some future directions, as well as observing potential contributions from AI. Hyperlinks to several of the LL systems surveyed are at www.aic.nrl.navy.mil/~aha/lessons.

2 Lessons learned definitions

Lessons learned were originally conceived of as guidelines, tips, or checklists of what went right or wrong in a particular event (Stewart, 1997). The Canadian Army LL Centre and the Secretary of the US Army for Research, Development, and Acquisition, among others, still abide by this notion. Today, this concept has evolved because organizations working towards improving the results obtained from LL systems have adopted acceptance criteria for lessons (*e.g.*, they have to be validated for correctness and should impact organizational behavior).

Several other definitions, emphasizing overlapping but non-identical criteria, are currently being used to define lessons and their processes. For example, some authors distinguish lessons from lessons learned. Bartlett (1999) proposes that a lesson learned is the change resulting from applying a lesson that significantly improves a targeted process. Similarly, Siegel (2000) argues that stored lessons are "identified lessons" rather than "lessons learned" in that they are records of potentially valuable experiences that have not (yet) necessarily been applied by others.

The DOE's Society for Effective Lessons Learned Sharing (SELLS) organization, which is perhaps the most mature organization (organizes semi-annual workshops) of its type in the USA, originally defined a LL as a *"good work practice or innovative approach that is captured and shared to promote repeat application. A LL may also be an adverse work practice or experience that is captured and shared to avoid recurrence"* (DOE, 1999). At their Spring 2000 Meeting, SELLS members discussed the following new standard definition: *"A lessons learned is the knowledge acquired from an innovation or an adverse experience that causes a worker or an organization to improve a process or activity to work safer, more efficiently, or with higher quality"* (Bickford, 2000a). Thus, definitions for lessons learned are still evolving.

The United States Air Force promotes a particularly intuitive definition (www.afkm.wpafb.af.mil):

> *"A lesson learned is a recorded experience of value; a conclusion drawn from analysis of feedback information on past and/or current programs, policies, systems and processes. Lessons may show successes or innovative techniques, or they may show deficiencies or problems to be avoided. A lesson may be:*
>
> > *1. An informal policy or procedure;*
> >
> > *2. Something you want to repeat;*
> >
> > *3. A solution to a problem, or a corrective action;*
> >
> > *4. How to avoid repeating an error;*
> >
> > *5. Something you never want to do (again)"*

However, the most complete definition for lessons learned is the one currently used by the American, European, and Japanese Space Agencies:

> *"A lesson learned is a knowledge or understanding gained by experience. The experience may be positive, as in a successful test or mission, or negative, as in a mishap or failure. Successes are also considered sources of lessons learned. A lesson must be significant in that it has a real or assumed impact on operations; valid in that is factually and technically correct; and applicable in that it identifies a specific design, process, or decision that reduces or eliminates the potential for failures and mishaps, or reinforces a positive result."* (Secchi et al., 1999)

This definition clarifies the guiding criteria needed for reusing lessons, and how reuse should focus on processes that a lesson can impact. This could lead us to another definition – one that focuses on how lesson dissemination improves a targeted process effectively. However, this definition may exclude some of today's LL systems. In Section 3, we examine the features of LL systems and use them to generate a categorization framework. Thus, instead of offering one all-encompassing definition, we hope to guide the reader to the most important issues that should be considered when designing LL systems under a given set of conditions.

Organizations that use LL systems describe different purposes for their use, including avoiding wasting resources (*e.g.*, a focus of the Air Force Air Combat Command Center's LL systems), protecting the safety of their workers (*e.g.*, a focus of the DOE's Corporate LL process), and to *"learn and live, otherwise die"* (*e.g.*, the Center for Army Lessons Learned (CALL)). Nevertheless, the underlying motivation is to help attain an organization's goals, regardless of their type.

3 Lessons learned systems

LL systems are motivated by the KM need to preserve an organization's knowledge that is commonly lost when experts become unavailable through job changes or retirement. The goal of LL systems is to capture and provide lessons that can benefit employees who encounter situations that closely resemble a previous experience in a similar situation. In this context, several proposed KM strategies employ different knowledge artifacts such as lessons learned, best practices, incident reports, and alerts. Lessons learned are usually described with respect to their origin (*i.e.*, whether they originate from an experience), application (*e.g.*, a task, decision, or process), orientation (*i.e.*, whether they are designed to support an organization or an entire industry), and results (*i.e.*, whether they relate to successes or failures). Table 1 contrasts some typical knowledge artifacts using these attributes. The following paragraphs refine these distinctions.

Table 1: Distinguishing some knowledge management artifacts.

Knowledge artifacts	Originates from experiences?	Describes a complete process?	Describes failures?	Describes successes?	Orientation
Lessons learned	yes	no	yes	yes	organization
Incident reports	yes	no	yes	no	organization
Alerts	yes	no	yes	no	industry
Corporate memories	possibly	possibly	yes	yes	organization
Best practices	possibly	yes	no	yes	industry

Incident reports: These describe an unsuccessful experience – an incident – and lists arguments that explain the incident without posing recommendations. This is the typical content of systems concerning safety and accident investigation. For example, the DOE disseminates lessons on their accident investigations, through the WWW, due to the extreme importance of these reports.

Alerts: These knowledge artifacts also each originate from a negative experience. They are reports of problems experienced with a particular technology or a part that is applicable to organizations in the same industry (Secchi, 1999). Alert systems manage repositories of alerts that are organized by a set of related organizations that share the same technology and suppliers. Some organizations use the same communication process to disseminate both lessons and alerts, which can be used as sources for creating lessons.

Corporate memories: This generic concept is not attached to a specific definition, although some attempts have been made to define (Stein, 1995) and even to classify corporate memories (Kühn & Abecker, 1997). A corporate (or *organizational*) memory is a repository of artifacts that are available to enhance the performance of knowledge-intensive work processes. Lessons learned, alerts, incident reports, data warehouses, corporate (*e.g.*, videotaped) stories, and best practices are instances of corporate memories.

Best practices: These are descriptions of previously successful ideas that are applicable to organizational processes. They usually emerge from reengineered generic processes (O'Leary 1999). They differ from lessons in that they capture only successful stories, are not necessarily derived from specific experiences, and they are intended to tailor entire organizational strategies.

LL systems that intermix lessons with other types of knowledge artifacts, including either the ones we mentioned here or others that are not easily reused (*e.g.*, reports, general information), can complicate the process of finding relevant lessons, and thus motivate the design of LL systems that focus exclusively on lessons. It is also possible to represent multiple lessons in a single database entry. However, in addition to possibly confusing users, this can cause several other problems, including complicating lesson verification, automated lesson reuse, the collection of reuse statistics, the representation of a lesson's result, and the prevention of duplicate lessons. These are compelling reasons to include only one lesson per database entry.

Another perspective on LL definitions stresses characteristics of knowledge representations and systems. For example, LL systems are not focused on a single task; they address multiple tasks in the same system. Thus, we can distinguish lessons in the context of knowledge representations (*e.g.*, cases, rules), identifying affinities and differences between them.

Cases: These are conceptually similar to lessons; both denote knowledge gained from experience and can be used to disseminate domain knowledge. However, while a library in a case-based reasoning (CBR) system is organized and indexed to accomplish a specific task (Kolodner, 1993), a LL database is not committed to only one particular task. Instead, it is tailored for an organization's members who can benefit from reusing its data for a variety of tasks, depending on the lesson content available. The two assumptions necessary to use CBR are also valid for lessons (*i.e.*, problems are expected to recur, and similar problems are solved using similar solutions (Leake, 1996)).

Rules: Although a lesson, like a rule, associates a set of precedents (conditions) with a consequent (suggestion), the suggestion may be instantiated differently depending on the context in which it is applied. Lesson reuse is more demanding than rule reuse because lessons require the user to recognize how to apply the lesson's suggestion for a given problem-solving context. Thus, lessons are tailored for use by field experts, and domain-specific knowledge is required for their reuse. Furthermore, lessons support partial matching (*i.e.*, of their conditions) during reuse, which differs from traditional rule-based approaches that require perfect matching. Rules (and cases) also require that their interrelations be considered during authoring, which is not necessary for lessons.

A complete and efficient KM strategy requires an organization to populate its corporate memory with lessons, best practices, and sector specific alerts. Some sectors may also benefit from maintaining benchmarked repositories (Mahe et al., 1996) and memories of operations. The strategy grounding the implementation of organizational memories should always be oriented to reuse. Section 4 surveys LL systems, while Section 5 describes how lessons can be represented to encourage reuse.

4 Surveying lessons learned systems

LL systems are ubiquitous. We located over forty LL systems on the WWW that are maintained by various government and other organizations (Aha & Weber, 1999). Existing systems for lesson dissemination are usually built using standalone retrieval tools that support variants of hierarchical browsing and keyword search.

LL systems have been the subject of a few recent workshops and surveys. SELLS has held workshops since 1996 (*e.g.*, SELLS, 1999). Also in 1996, the International Conference on Practical Aspects of Knowledge Management held a small workshop on *The Lessons Learned Cycle: Implementing a Knowledge Pump in your Organisation* (Reimer, 1996). In 1999, the European Space Agency (ESA) sponsored the workshop *Alerts and Lessons Learned: An effective way to prevent failures and problems* (Secchi, 1999), which included contributions that discussed implementations of LL systems for the ESA, Alenia Aerospazio, the Centre National d'Etudes Spatiales (CNES), and National Space Development Agency of Japan (NASDA).

The most ambitious investigation of LL processes was performed by the Construction Industry Institute's Modeling LL Research Team (Fisher et al., 1998). They surveyed 2400 organizations, characterized the 145 initial responses as describing 50 distinct LL processes, and performed follow-up, detailed investigations with 25 organizations. They concluded that there was strong evidence of weak dissemination processes, and few companies performed a costs/benefits analysis on the impact of their LL process. Secchi et al. (1999) describe the results of a similar survey, focusing on the space industry, in which only 4 of the 40 organizations that responded were using a computerized LL system. In both surveys, none of the responding organizations implemented a LL process that proactively 'pushed' lessons to potentially interested customers in the lesson dissemination sub-process. This lack of emphasis on active lessons dissemination is probably because software was not used to control the process(es) targeted by the lessons, or elicited lessons were immediately incorporated into the targeted process (*e.g.*, into the organization's best practice manuals, or by requiring project members to read through project-relevant lessons prior to initiating a new project). This is not feasible in a military context, where the doctrine-updating process is rigorous and slow, and where archived lessons are needed to store crucial information that has not yet been accepted into doctrine, or is too specific (or otherwise inappropriate) for inclusion into doctrine.

LL systems, in general, poorly serve their intended goal of promoting knowledge reuse and sharing. Two reasons are paramount for this failure. First, the selected *representations* of lessons are typically inadequate. That is, they are not usually designed to facilitate reuse by lesson dissemination software, either because they do not clearly identify the process to which the lesson contribution applies, or its pre-conditions for application. A primary contributing factor to this problem is that most lessons are described as a set of free-text fields. Second, these systems are typically not integrated into an organization's decision-making process, which is the primary requirement for an AI solution to successfully contribute to KM activities (Reimer, 1998; Aha et al., 1999). These observations prompted our decision to examine LL systems in more detail, seeking to identify their distinguishing characteristics and to encourage the development of LL dissemination systems that successfully address these two issues.

We created a two-part categorization framework for LL systems. Section 4.1 refers to the categories of the processes that LL systems are designed to support, while Section 4.2 refers to system categories themselves.

4.1 Categorizing lessons learned processes

LL systems exist to support organizational processes. Based on a survey of organizations that deploy and utilize LL systems, we have identified the essential components of a generic LL process (Figure 1). Flowcharts describing LL processes abound; almost all organizations produce them to communicate how lessons are to be acquired, validated, and disseminated (*e.g.*, Fisher et al., 1998; SELLS, 1999; Secchi, 1999). As an organizational process, it involves both human and technological issues. We limit our research scope to the technological issues.

The primary LL sub-processes are: collect, verify, store, disseminate, and reuse.

Collect: This sub-process has been performed in four different ways, and we propose two additional lesson collection methods. Table 2 then presents a summary.

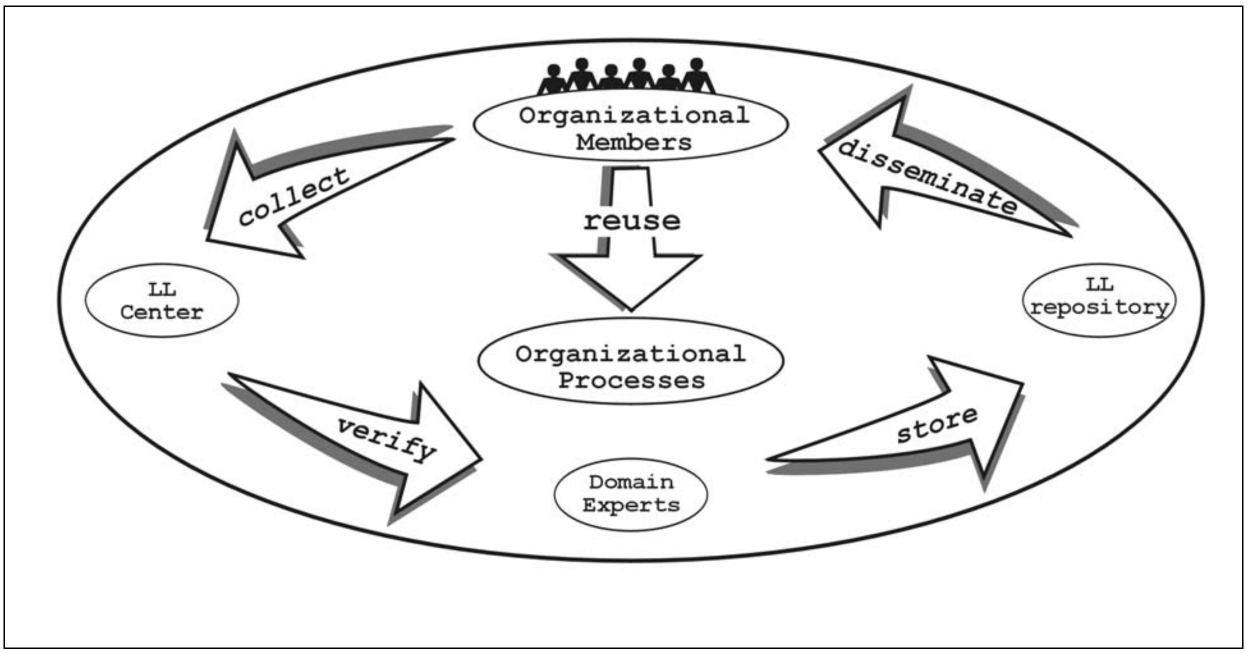

Figure 1: A generic lesson learned process.

Passive collection. Organization members submit their own lessons using a form (e.g., online) in 2/3 of the organizations surveyed. For example, CALL has an excellent passive collection form with online help and examples.

Reactive collection. Members are interviewed to collect lessons (e.g., Nemoto et al., 1999; Tautz et al., 2000; Vandeville & Shaikh, 1999).

After action collection. This approach is typically used by military organizations to collect lessons after missions, and has been adopted by J.M. Huber (Beebe, 2000) and the ESA (Secchi et al., 1999). Different organizations can benefit from lesson collection during or near the completion of a project (Vandeville & Shaikh, 1999).

Proactive collection. In this case, lessons are captured while problems are solved, as in the military active collection method (see below). However, lessons can also be automatically collected. CALVIN (Leake et al., 2000) employs an example of this method, in which the user can override the system's suggested lessons.

Table 2: The lesson collection strategies employed by surveyed organizations.

Passive	Accident Investigation LL, Air Combat Command Center for LL, AFCKS, Berkeley Lab LL Program, CALL, DOE Corporate LL Collections, US DOE Office of Environmental Management (EM) LL database, Federal Transit Administration LL Program, Idaho National Engineering and Environmental Laboratory, JCLL, Lawrence Livermore National Laboratory (LLNL), Reusable Experience with Case-Based Reasoning for Automating LL (RECALL), US Army Medical LL (AMEDD), Navy Lessons Learned System (NLLS), Project Hanford LL, Automated LL Collection And Retrieval System (ALLCARS), DOE's Environment, Safety and Health (ESH) LL Program at the Los Alamos National Laboratory, Xerox's Eureka system
Reactive	The COIN best practices system (Tautz et al., 2000), NASDA
After action report	Alenia Aerospazio Space Division, Canadian Army LL Centre, ESA's LL system, JCLL, Marine Corps LL System, NLLS
Proactive	CALVIN (Leake et al., 2000)
Active (scan)	ESA LL, Lockheed Martin LL, Project Hanford LL, ESH LL Program
Active (military)	CALL, JCLL, NLLS

Active collection. At least two methods are called active. Active scan attempts to find lessons in documents and in communications among organization's members (Knight & Aha, 2000). In contrast, the military active collect method (Tulak, 1999), used by military organizations, is well directed and thus more promising: problems demanding lessons are identified and a collection event is planned to obtain relevant lessons. This involves four phases: mission analysis and planning, deployment and unit link-up, collection operations, and redeployment and report writing.

Interactive collection. Weber et al. (2000) proposed a dynamic intelligent elicitation system for resolving ambiguities in real time by interacting with the lesson's author and relevant information sources.

Verify: A team of experts usually performs this sub-process, which focuses on validating lessons for correctness, redundancy, consistency, and relevance. In military organizations, verification categorizes lessons according to task lists (e.g., the Universal Naval Task List (OPNAVINST, 1996)). In LL systems designed for training purposes, verification can be used to combine and adapt complementary or incomplete lessons.

Store: This sub-process addresses issues related to the representation (e.g., level of abstraction) and indexing of lessons, formatting, and the repository's framework. Lesson representations can be structured, semi-structured, or in different media (e.g., text, video, audio) (e.g., Johnson et al. (2000) focus on video clips in which experts provide relevant stories). Task-relevant representations, such as the DOE's categorization by safety priority, are also often used.

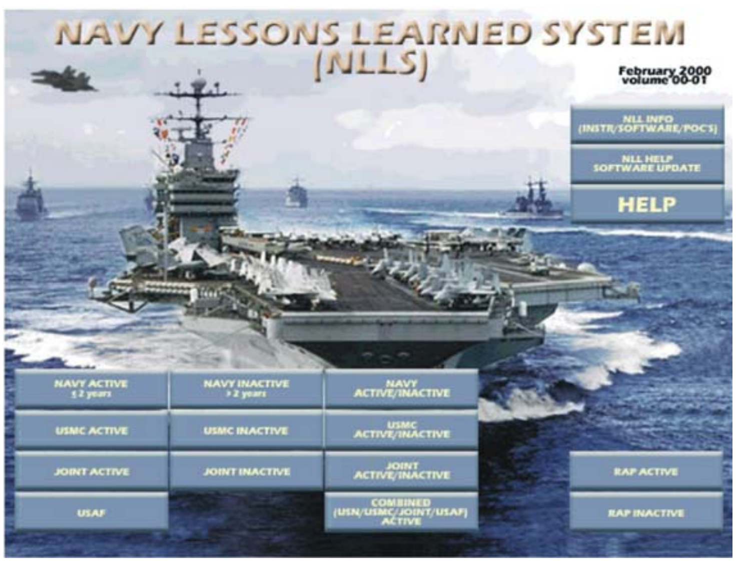

Figure 2. The user interface for the Navy Lessons Learned System.

Disseminate: The dissemination sub-process may be the most important with respect to promoting lesson reuse. We have identified five dissemination methods, which are detailed below. Table 3 then provides a summary.

Passive dissemination. Users search for lessons in a (usually) standalone retrieval tool. The system remains passive. Although this is the most traditional form of dissemination, it is ineffective. Figure 2 shows the top-level interface for the (unclassified) Navy Lessons Learned System (NLLS), whose February 2000 version combines approximately 49,000 lessons learned from four services. Although impressive in its interface and contents, it is, like most other LL dissemination systems, limited in that it implements a passive dissemination approach.

Active casting: In this method, adopted by the DOE and the Canadian Army, lessons are broadcast to potential users via a dedicated list server. Recently, the Air Force Center for Knowledge Sharing (AFCKS) has adopted a similar approach in which user profiles are collected to ensure that lessons, when received, are disseminated to users whose profiles (i.e., interests) match the lesson's content.

Broadcasting. Bulletins are sent to everybody in the organization, as is done in some LL organizations (e.g., CALL). Another form of broadcasting is performed by the NLLS, which sends CD-ROMs containing the NLLS databases to many Navy organizations.

Active dissemination: Users are dynamically notified of relevant lessons in the context of their decision-making process, as exemplified by systems described by Weber et al. (2000) and Leake et al. (2000).

Proactive dissemination: The system builds a model of the user's interface events to predict when to prompt users with relevant lessons. This approach is used by Microsoft (Gery, 1995) and was used by Johnson et al. (2000) in the Air Campaign Planning Advisor (ACPA) to disseminate videotaped stories. We discuss ACPA further in Section 6.2.2.

Reactive dissemination: When users realize they need additional knowledge, they can invoke a help system to obtain relevant lessons and related information. This is used in the Microsoft Office Suite and in ACPA.

Table 3: The dissemination sub-processes employed by surveyed organizations.

Passive	Air Combat Command Center for LL, AFCKS, Berkeley Lab LL Program, CALL, EM LL database, ESA LL, JCLL, LLNL, Lockheed Martin LL, RECALL, AMEDD, NLLS, NAWCAD's Center for Automated Lessons Learned (NAWCAD/CALL), Project Hanford LL, ALLCARS, Alenia Aerospazio Space Division, Eureka
Active casting	Accident Investigation LL, CALL, DOE Corporate LL Collections, LLNL, Lockheed Martin LL, Project Hanford LL, ESH LL Program
Broadcasting	Canadian Army LL Centre, DOE Corporate LL Collections, Federal Transit Administration LL Program, Marine Corps LL System, NLLS
Active	None deployed; proposed by Weber et al. (2000) and Leake et al. (2000)
Proactive	None deployed; proposed in ACPA (Johnson et al., 2000)
Reactive	None deployed; proposed in ACPA (Johnson et al., 2000)

Among the small number of organizations we have surveyed (about 40), at least 17 use passive dissemination. This method makes several strong assumptions regarding its users (*e.g.*, that the user knows about the existence of the LL systems, knows where to find it, has the skills to use it or time to learn how to use it, knows how to interpret its results). These are too demanding. Alternative methods have been used by a smaller number of LL organizations such as active scan and broadcasting, whereas active, proactive, and reactive methods have only been implemented in research prototypes.

Reuse: The choice of whether to reuse a lesson's recommendation is made by the user. Automatic reuse can only be conceived in the context of an embedded architecture, which is rare (e.g., ACPA (Johnson et al., 2000), ALDS (Weber et al., 2000), and CALVIN (Leake et al., 2000)). We have identified three categories of reuse sub-processes:

Browsable recommendation: The system simply displays a retrieved lesson's recommendation, as is done in most LL tools.

Executable recommendation: Users can optionally execute a retrieved lesson's recommendation (Weber et al., 2000). This capability requires embedding the reuse process in a decision support software tool.

Outcome reuse: This involves recording the outcome of using a lesson, which can help to identify a lesson's utility. For example, in Lockheed Martin's Oak Ridge LL system, LL coordinators are expected to identify actions taken or planned relative to given lessons. Comments on the outcome observed after reuse may not demand substantial time in comparison to the potential benefits (e.g., identifying useless lessons for subsequent removal).

Using artificial intelligence techniques can potentially enhance LL sub-processes. For example, Sary & Mackey (1995) used conversational case retrieval to improve recall and precision for a passive dissemination sub-process. We discuss this further in Section 6.

4.2 Categorizing lessons learned systems

Besides the characteristics identified by the different methods employed in each of the sub-processes, we have identified a set of other characteristics to infer trends in the design of LL systems. This categorization for LL systems is based on the system's content, role, orientation, duration, organization type, architecture, representation (*i.e.*, attributes and format), confidentiality, and size. We selected a subset of the organizations surveyed to illustrate this categorization framework.

Some trends stand out based on the number of examples in certain categories. Because LL systems have the reputation for being under-utilized, we attempt to identify some reasons to both explain and address this problem by highlighting relevant trends.

Content: Because lessons are not the only KM artifacts designed for reuse, some organizations will use similar collect, verify, store, disseminate, and reuse sub-processes for objects such as incident reports or alerts. *Pure* LL systems only manipulate lessons; *hybrid* systems also include other objects (*e.g.*, the DOE Corporate Lessons Learned Collections also store alerts and incident reports).

Table 4: Content of lessons learned systems.

Pure	Air Combat Command Center for LL, AFCKS, JCLL, RECALL, Marine Corps LL System, AMEDD, Air Force Center for Knowledge Management, Eureka
Hybrid	Accident Investigation LL, Canadian Army LL Centre, Berkeley Lab LL Program, CALL, DOE Corporate LL Collections, EM LL database, Federal Transit Administration LL Program, Idaho National Engineering and Environmental Laboratory, ESA LL, LLNL, Lockheed Martin LL, NLLS, NAWCAD/CALL, Project Hanford LL, ESH LL Program

A high percentage of organizations use hybrid repositories (Table 4). This decision may be related to their low effectiveness, given that reuse is enhanced by using homogeneous lessons, which are more amenable to computational processing. We suggest designing knowledge artifact repositories that clearly distinguish lessons from other artifacts.

Role: LL systems differ according to the nature of the processes (*roles*) and users they are designed to support. For example, military personnel execute planning processes (*i.e.*, tasks are part of plans with established goals, usually in a multi-person, distributed context). In contrast, technicians are users whose technical processes often require applying domain-specific expertise for diagnosis and troubleshooting. This distinction has motivated us to create two categories of roles (Table 5). Due to their distinctive nature, they require different LL system requirements (*e.g.*, for lesson dissemination, representation, and verification). Using this perspective, storing lessons with different roles (both planning and technical) can negatively impact system effectiveness. If these two types of lessons are stored separately, then the resulting homogeneity should simplify lesson retrieval.

Table 5: Roles for lessons learned systems.

Planning	Air Combat Command Center for LL, AFCKS, AMEDD, Canadian Army LL Centre, JCLL, Marine Corps LL System, NAWCAD/CALL, NLLS
Technical	Accident Investigation LL, Alenia Aerospazio Space Division, Berkeley Lab LL Program, DOE Corporate LL Collections, EM LL database, Federal Transit Administration LL Program, ESA LL, Eureka, Project Hanford LL
Both	Air Force Center for Knowledge Management, CNES, ESH LL Program, Idaho National Engineering and Environmental Laboratory, LLNL, Lockheed Martin LL, RECALL

Orientation: Typically, LL systems are implemented to support one organization, and they should be built in accordance with that organization's goals (Table 6). Some LL systems are built to support a group of organizations (*e.g.*, the European Space Agency maintains a system for its community), while others have a task-specific scope (*e.g.*, CALVIN (Leake et al., 2000) was designed to collect and share lessons on which information sources to search for a given task). Most of the LL systems that we surveyed are specific to a particular organization; only five share lessons for and across an entire corporation.

Table 6: Orientation of lessons learned systems.

Corporate-wide LL systems	Accident Investigation LL, DOE Corporate LL Collections, JCLL, NAWCAD/CALL, Air Force Center for Knowledge Management LL
Organizational LL systems	Air Combat Command Center for LL, AFCKS, Canadian Army LL Centre, EM LL database (DOE), CALL, Federal Transit Administration LL Program, ESA LL, LLNL, Lockheed Martin LL, RECALL, Marine Corps LL System, AMEDD, NLLS, Project Hanford LL, ESH LL Program, Alenia Aerospazio Space Division, Eureka, NASDA, CNES

Duration: Most LL systems are permanent, although temporary ones may be created due to a temporary job or event (*e.g.*, a temporary LL system was created to support the Army Y2K Project Office).

Organization type: We distinguish organizations as either *adaptable*, in which case they can quickly incorporate lessons learned in their processes, or *rigid*, in which case they use doctrine that is only slowly updated. Adaptable organizations do not necessarily need to maintain a *permanent* lesson repository because lessons, once incorporated into these organizations' processes, have already been learned/reused. In contrast, rigid organizations (*e.g.*, military organizations) have a greater need to maintain lesson repositories because they may exist for a long time prior to the incorporation of lesson knowledge into doctrine, or lessons may not be deemed sufficiently general for inclusion into doctrine. Organization type can greatly influence lesson representation and LL processes.

Architecture: LL systems can be standalone or embedded in a targeted process. Embedded systems can use an active, proactive, or reactive dissemination sub-process (Johnson et al., 2000; Weber et al., 2000). Embedded LL systems can alternatively be accessed via a link in the decision support tool (Bickford, 2000b).

Attributes and Format: Most LL databases (~90%) include both textual and non-textual attributes. Lessons are initially collected in text format and then supplemented with fields to provide structure.

Confidentiality: Lessons can be classified, unclassified, or restricted. For example, the USAF's Center for Knowledge Sharing provides Internet access to unclassified lessons and SIPRNET (Secret Internet Protocol Router Network) access to classified lessons. The Internet site also provides access to classified lesson *titles*, which simplifies finding these lessons on the corresponding SIPRNET site.

Table 7: Size of (unclassified) lessons learned system repositories.

< 100	Accident Investigation LL, Air Combat Command Center for LL, AMEDD, Berkeley Lab LL Program, Federal Transit Administration LL Program, Idaho National Engineering and Environmental Laboratory
100-1,000	DOE Corporate LL Collections, EM LL database, Lawrence Livermore National Laboratory, Lockheed Martin LL, Project Hanford LL, RECALL
1000-5,000	ALLCARS, JCLL, Marine Corps LL System, NLLS, NAWCAD/CALL
5,000-10,000	CALL
30,000+	AFCKSLL, ESH LL Program, Eureka (Xerox)

Size: Military organizations are all grouped within the intermediary ranges from 1,000 to 10,000 lessons (Table 7), and these are relatively old organizations: the Joint Center for Lessons Learned (JCLL) (1988), the Marine Corps LL System (1994), the Navy Lessons Learned System (1991), the Combined Automated Lessons Learned system of the Navy Air Warfare Center, Aircraft Division (1994), the ALLCARS system supported by the USAF Center for Knowledge Management, and the Center for Army Lessons Learned (1995). In fact, the JCLL has culled its database from 13,000 to 2,000 lessons to remove "lessons" that were not validated, irrelevant lessons, and redundancies. (This highlights the importance of tightly integrating verification with lesson collection.)

LL systems can be distinguished into two main groups. The first group is composed of military organizations that employ mostly pure LL systems with lessons that target planning processes. The second group is composed of technical LL systems with hybrid repositories, which is typical of DOE organizations and its contractors. We discuss these groups in more detail below.

4.2.1 Pure planning systems group

Most military organizations (*e.g.*, the AFCKS, JCLL, NAWCAD/CALL) maintain pure planning LL systems. It is typical of military organizations and their personnel to envision their processes as plans. The after action report is the most frequently used method for collection, followed by the (military) active collection method, which is used when a collection event is planned to extract lessons for a predefined set of problems. Dissemination is primarily passive and broadcast; all military organizations have passive standalone online repositories, and many also broadcast lesson repositories on a CD-ROM.

4.2.2 Hybrid technical systems group

The Department of Energy has a society for LL (SELLS) with over ninety members. Lessons shared within the DOE typically address technical processes. Because their main concern is safety, these LL systems are also used to deliver alerts, incident reports, and general information. The main collection process is active scan, focusing on project reports. Each LL site has a coordinator who is responsible for collecting, verifying, and storing lessons, along with publicizing the availability of these lessons.

4.2.3 Other potential groups

The utility of corporate LL systems indicates that, even when lessons are individually collected, they can benefit from being reused in other organizations that are in the same industry, share the same interest, and, most importantly, are not competitors. NAWCAD/CALL is a good example; it compiles aircraft related lessons from the aviation branches of several organizations (*e.g.*, Air Force, FAA, Navy).

The ESLL (ESA LL system (Secchi et al., 1999)) focuses on corporate lesson sharing for the ESA and its contractors, but it is also intended to be shared with other space agencies such as CNES and Alenia Aerospazio. One of the goals of the 1999 ESA sponsored workshop was the mutual exchange of lessons in the space community. Space agencies are a group in which *ad hoc* solutions can be suggested since lessons can reach a large range of organizations.

Section 4 discussed a categorization framework for LL systems, but did not address computational issues on how to represent lessons to promote reuse. The following section focuses on lesson representation.

5 Representing lessons learned

AI approaches typically represent knowledge artifacts using representations (*e.g.*, cases, rules, concept maps) that support computational reasoning. Given our interest in using AI technology to enhance LL processes, we examined LL repositories to identify patterns that facilitate their conversion into structured and homogeneous representations. Not surprisingly, we found that lesson authors do not all employ the same lesson format. This complicates our task because we wish to create a single, structured, stereotypical, and effective lesson representation. We also found that most LL repositories include a monolithic "lesson content" text field, and that there is a trend to isolate the "*recommendation*" (or "*suggestion*") field.

Table 8 summarizes some representational issues for 20 LL systems in which we had available a sufficient number of lessons to detect a style in writing. We also identified the format of attributes and the lessons' role for each repository.

Use of conditionals: All but one of these 20 repositories contained lessons that are written in statements that something "should" be done (*i.e.*, "*when these conditions take place, you should apply this lesson*").

Author's Identity: About 25% of these repositories do not disclose the author's identity. CALL's lesson insertion process is the only one that provides the users with an explicit option for revealing their identity while submitting lessons. This is an important organizational issue that highlights personnel concerns; after the author's identity is disclosed, the information may be used for purposes other than sharing knowledge (*e.g.*, job evaluation). Anonymity might contribute to obtaining more realistic lessons.

Attributes: Lessons learned centers predictably develop lesson databases, which typically use a combination of textual and non-textual attributes to store lessons. One exception is the Canadian Army's LL Centre, which maintains text summaries of lessons but no database. Another exception is NASA-Goddard's RECALL system, which uses an interactive CBR approach to improve lesson retrieval. This involves representing lessons structured as *cases*, which are stored in a combination of text and <question,answer> pairs. Representing lessons as cases deserves further consideration because the benefits of potentially improved retrieval performance have to be balanced against an increase in knowledge engineering efforts. We discuss this further in Section 5.2.

Table 8: Characteristics of lesson representations among lessons learned repositories.

#	Organization's name	Lessons Expressed with Conditional	Includes Author's Identity?	Text or Non-text Attributes?	Technical or planning role?
1	Accident Investigation LL (DOE)	yes	yes	T & NT	T
2	Air Combat Command Center for LL	yes	yes	T & NT	P
3	AFCKS	yes	yes	T & NT	P
4	Canadian Army LL Centre	yes	**no**	**T**	P
5	NAWCAD's Combined Automated LL	yes	yes	T & NT	T
6	Berkeley Lab LL Program	yes	yes	T & NT	T
7	CALL	yes	**optional**	T & NT	P
8	DOE Corporate LL Collections	yes	yes	T & NT	T
9	EM LL database	yes	yes	T & NT	T
10	Federal Transit Administration LL Program	yes	yes	T & NT	T
11	Idaho National Engineering & Environmental Laboratory	yes	yes	T & NT	B
12	The ALLCARS LL database	yes	**no**	T & NT	P & T
13	Joint Center for LL	yes	yes	T & NT	P
14	LLNL	**no**	**no**	T & NT	B
15	Lockheed Martin LL	yes	yes	T & NT	T & P
16	RECALL	yes	**no**	**T & Q/A pairs**	B
17	Marine Corps LL System	yes	yes	T & NT	P
18	Medical LL (AMEDD)	yes	**no**	T & NT	P
19	Navy LL System (NLLS)	yes	yes	T & NT	P
20	Project Hanford LL Database (in AJHA)	yes	yes	T & NT	T

Legend: T = "textual", NT = "non textual", T (role) = "technical", P = "planning", B= "both"

This brief overview of lesson representations suggests three issues to resolve when designing LL systems: the style for expressing lessons, whether to require the author's identity, and whether to use textual attributes. Because lessons differ substantially according to the role that they serve, we advocate a different representation for each role, as described in Sections 5.1 and 5.2.

5.1 Representing planning lessons

Planning lessons teach something in the context of executing a plan, where the content of the lesson (*i.e.*, its *contribution*) will modify the way that a task is performed, thus changing an evolving plan. This highlights the important role that a task must play in providing a planning lesson's context as well as the lesson's contribution. We define a planning lesson as follows, provide intuitive examples, and then define its six highlighted (boldfaced) components.

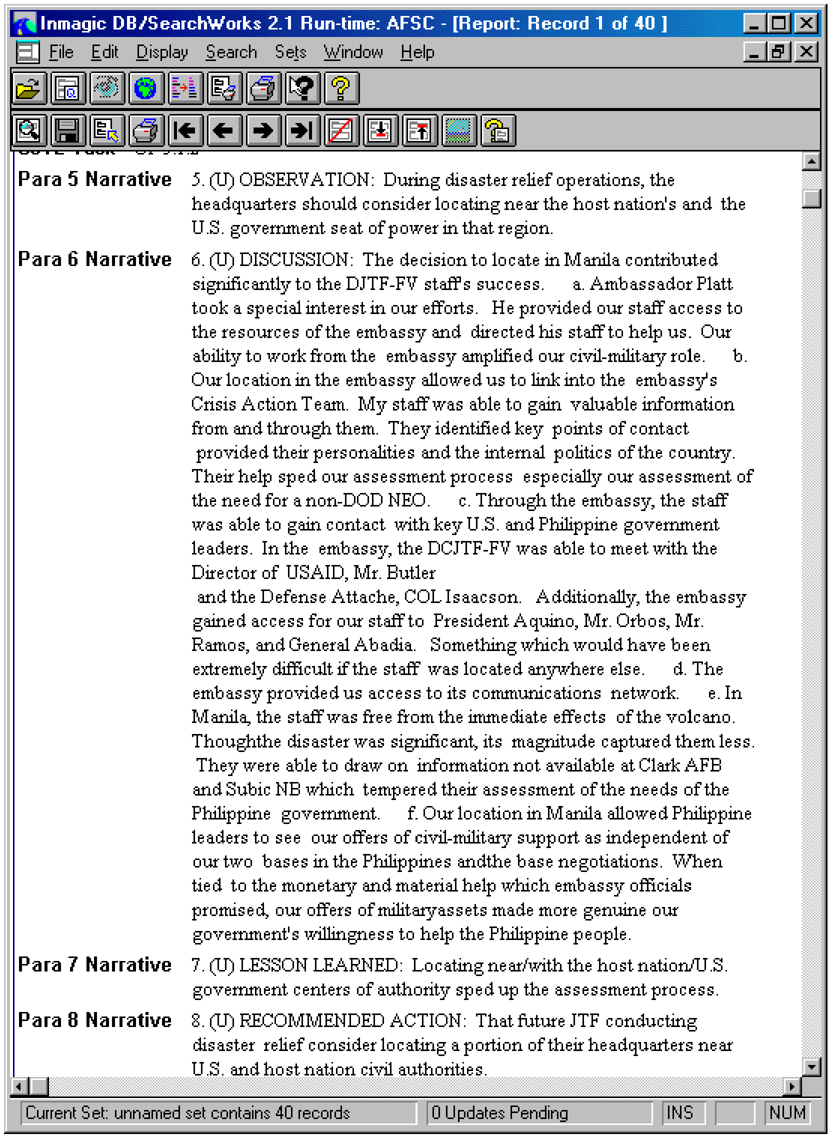

Figure 3: Example of a successful lesson.

Definition: A **planning lesson** concerns the application of an **originating action**, under a given set of **conditions**, that, when combined with a **contribution**, yields a **result,** which can be positive or negative. Lessons also contain a **suggestion** that defines, when performing an **applicable task** under similar conditions, how to reuse this lesson.

We assume the *result* was observed when the lesson was collected, regardless of the amount of elapsed time since the originating action. Within this context, the *reuse* of a planning lesson can be stated as follows.

For lessons corresponding to successes: Under similar conditions, repeat the originating action to ensure that the lesson contribution will cause a similar result.

For example, Figure 3 displays an unclassified, *positive* (*i.e.*, originating from a success) lesson for disaster relief planning from the Joint Center for Lessons Learned. In particular, the *originating action* concerned the decision for where to locate the Joint Task Force (JTF) headquarters, under the context (*conditions*) of a civilian disaster relief effort in a foreign nation where there is a U.S. government seat of power. The *contribution* of this lesson specifies locating the JTF headquarters near the U.S. government seat of power. The *result* is positive; the text stresses the success of the location (*i.e.*, the decision to locate in Manila contributed significantly to the DJTF-FV staff's success). The *suggestion* is to situate a portion of the JTF headquarters near U.S. and host nation civil authorities when the *task* is to locate the JTF's headquarters.

For lessons corresponding to failures (or accidents): Under similar conditions, avoid repeating the originating action to prevent the contribution from causing a similar result.

Figure 4 displays another unclassified lesson, this time based on a failure. The *originating action* concerns implementing a "triple registration process." The *contribution* concerns the use of this process for processing evacuees. The result is inferred to be negative from the statement "time consuming and evacuee discomfort." The suggestion is to "locate an Immigration and Naturalization Service (INS) screening station at the initial evacuation processing site. Evacuees are required to clear INS procedures prior to reporting to the evacuation processing center." The applicable task is evacuee registration.

With these examples in mind, we now define our representation's six components for planning lessons:

Definition: An **originating action** is the action that occurred that caused a lesson to be learned.

Definition: Conditions define the context (*e.g.*, weather variables, or an organizational process) in which applying an **originating action**, combined with a **contribution**, will yield a specific **result**.

Even when a few conditions are absent, a lesson may still be valid. If none of the conditions hold, then it is very likely that the lesson is not valid.

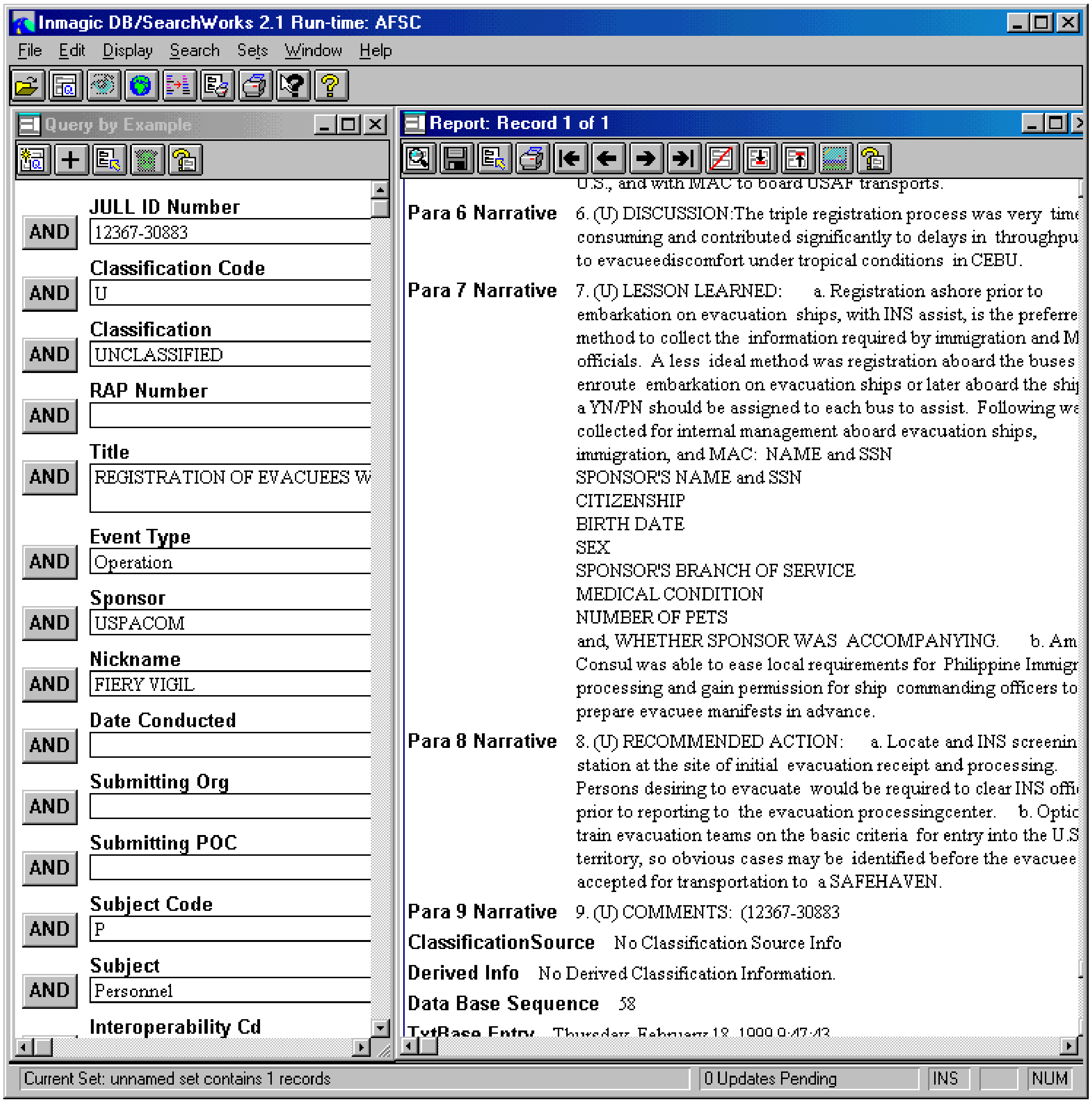

Figure 4: Example of a failure lesson.

Definition: The lesson **contribution** is the component that, when combined with the **originating action**, yields the **result**. It captures the causality of the lesson. The lesson contribution frequently has to be interpreted by domain experts.

A contribution in a planning lesson can be, for example, a method, a resource, the inclusion of an element onto a checklist, or the review of a relevant document.

Definition: The **result** is a consequence, caused by the **contribution** when the **originating action** was applied under the **conditions,** that can significantly impact a plan's performance (i.e., either positively or negatively).

Definition: The **suggestion** is a recommended response or action. It is an interpretation of the lesson that either promotes the reuse of a contribution (for positive lessons), or recommends avoiding the contribution (for negative lessons).

Lesson representations in most LL repositories include a component named *recommended action* or *recommendation* rather than *suggestion*. We use *suggestion* here, having been influenced by Sampson's (1999) finding that, by using this name, users feel this gives them more freedom of choice in their decision-making processes.

Definition: An **applicable task** of a lesson is the task (*i.e*., or applicable decision or process) that the lesson, when reused, can positively impact.

The information recorded in a lesson requires both an applicable task and an originating action because they can differ. Usually, we learn something from observing the result of applying an action, in which either the action or the conditions in which it is applied differ from previous experience. The lesson contribution provides a causal explanation for the result. Typically, the applicable task is the same as the originating action. However, these can differ; it is possible to learn lessons that are applicable to other tasks (*e.g*., finding petroleum when digging for water).

This representation, which reduces a planning lesson's content to a minimum, facilitates lesson reuse. We wish to minimize the amount of useless information represented in a lesson, and simultaneously target the lesson to a specific audience that can benefit from its reuse.

In the context of locally collected lessons, we can assume "If there is a lesson, there is at least one task in which a lesson is applicable." However, it is probable that expert users in a given domain, who are capable of analogical reasoning, find multiple applicable tasks for a given lesson. Therefore, users can submit additional applicable tasks for a single lesson and in doing so enhance opportunities for its reuse.

This subsection focused on a generic representation for planning lessons. Several representation techniques, popular in artificial intelligence research, are applicable to lessons learned systems designed to support planning. For example, we are examining the use of hierarchical task networks for plan representation (Weber et al., 2000).

5.2 Representing technical lessons

Technical lessons are typically the result of a technician's experiences. Lessons are usually related to a portion of a large technical system that is known by all technicians familiar with its domain (*e.g.*, flight engineers). Technical work is not delivered through plans, but through jobs or projects. In this context, technical lessons refer to problems, their causes, and their solutions.

Not surprisingly, technical repositories are typically represented using either <problem, solution> pairs (*e.g.*, trouble tickets in the HVAC LL system (Watson, 2000)) or <problem, cause, solution> tuples (*e.g.*, EUREKA's tips (Everett & Bobrow, 2000)). This latter representation is particularly effective because it can be used to provide explanations to the user. This is important for technical domains, as exemplified in the Project Hanford LL system (Bickford, 2000b), whose lessons often include the expression *root cause*.

Xerox's EUREKA LL system is somewhat unique in that it is completely self-sustained by its intended users; no lessons learned organization was responsible for creating it. There are some significant benefits to this approach. For example, each lesson includes the author's identification, which has generated prestige among the company's technicians, and is an effective motivator for tip contribution. Also, Xerox technicians have proactively extended lesson representations with additional information (*i.e.*, media attachments such as charts, pictures, and videos), which can further promote lesson reuse.

NASA Goddard Space Flight Center's RECALL LL system represents lessons as *cases* (Sary & Mackey, 1995), which allow users to search for them using a constrained dialogue approach known as *conversational case-based reasoning* (Aha & Breslow, 1997). This involves structuring the lesson's subject by using <question, answer> pairs. RECALL incrementally focuses the user's search on a decreasing number of cases each time the user selects and answers a question that distinguishes the top-ranking lessons. However, the lesson contribution (*i.e.*, the contribution this lesson offers to an activity) is represented only through these pairs, and is not stored explicitly for subsequent analysis.

Representing lessons using cases has both benefits and drawbacks. For example, using structured case representations could be used, as in RECALL, in an attempt to increase retrieval performance. An extension of this approach, use for planning, might store technical lessons separately from planning lessons, which can enhance clarity and provide a greater degree of freedom in their representation. However, as with other representations popular in artificial intelligence, cases require significant knowledge engineering effort.

Another innovation for representing lessons is used in the evolving LL system of the Project Hanford Site of the Department of Energy (Bickford, 2000b). Several attributes were introduced for its lessons collected in 2000 that were not included in lessons acquired in previous years. For example, this includes *estimated savings/cost avoidance.* Using an attribute for estimated savings or cost avoidance could help to determine the effectiveness of LL systems (*i.e*., highlight metrics for performance analysis), given information on when a given lesson was (successfully) applied.

The database used in NAWCAD's CALL system exemplifies how lessons from different organizations but in the same context can share the same repository. Although it contains only technical aviation lessons, it contains repositories for the FAA, Navy, and Air Force. Because its domain is somewhat restricted, it was possible to identify a *limited* set of "impact areas" (Table 9) that lessons can target, which can promote lesson reuse. That is, this impact field can serve as a primary index for partitioning the database, and thus simplify retrieval. Table 9 shows some of the 43 impact areas used in NAWCAD's CALL system.

Table 9: Some impact areas used for lessons in NAWCAD's CALL system.

Reliability
Technical Publications
Design Engineering
Survivability
Maintainability
Supply Support
Design Engineering
Maintenance Engineering
Life Cycle Cost

Although useful for administrative purposes, the impact areas used in NAWCAD's CALL system are not optimal, and could benefit from an AI perspective. In particular, it categorizes problems by organizational departments, which is not ideal for promoting lesson reuse. A more useful approach would categorize problems using a model-based framework (*e.g*., of an aircraft), although this would require locating each lesson in the context of a functioning model. For instance, lessons about life cycle cost, design engineering, and technical publications about an aircraft's wings should all be closely grouped rather than distributed according to the organization responsible for producing them.

The National Security Agency's (NSA's) LL database uses another lesson categorization that is not designed to promote reuse (O'Leary, 1998). It contains three types of lessons: *informational* (*e.g.*, how an NSA employee's duties could be changed during times of emergencies), *successful* (*e.g.*, capture effective responses to a crisis), and *problem* (*i.e.*, describe examples of actions that failed and potential ways to resolve them). When searching for lessons, users are required to know the categorization of the lesson they need prior to their search. This complicates the retrieval of pertinent lessons. We suggest that LL systems categorize lessons by their contribution rather than, or at least in addition to, the type of experience from which they were derived (*e.g.*, success, failure).

Sophisticated causal and domain modeling approaches from AI research can be used to benefit lesson representation and reuse. Among the few commercial tools, REASON™ by Decision Systems Inc., is unique in that it allows users to build a causal model for technical problems, representing a potentially beneficial framework to represent lessons (*i.e.*, this can facilitate both lesson elicitation and explanation). Alternatively, models that describe a domain's objects and relations (*e.g.*, concept maps (Leake et al., 2000)) can provide a user with a systemic view (Senge, 1990), allowing users to browse the "neighborhood" of their problem in their search for reusable knowledge artifacts. In addition, users can use these tools to communicate their interests, so that they can be automatically informed of relevant lessons that are made available at a later date.

Choices in lesson representation can affect reuse frequency. However, some choices require substantial knowledge engineering effort, and each organization must weigh the tradeoffs. In the following section, we discuss other AI techniques that can potentially increase lesson reuse in LL systems.

6 Incorporating AI in lessons learned systems

In the previous sections we introduced and summarized LL processes and systems. Although well-known in KM research, this is a relatively new applications area in AI. This has motivated us to organize a workshop on this subject (Aha & Weber, 2000) and, in doing so, to learn about potential trends in how AI may be applicable to increase the effectiveness of LL systems.

AI techniques can clearly address two key problems in designing LL systems, relating to lesson representation and system architecture, respectively. In Section 6.1, we use our categorization of lessons learned processes (Section 4) to briefly note how AI techniques can apply to various sub-processes. Then, in Section 6.2, we focus on the benefits of embedding LL systems in decision support systems (*e.g.*, opportunities for proactive lesson dissemination so as to increase the frequency of lesson reuse).

6.1 Intelligent lessons learned processes

In Section 4 we introduced a categorization of LL processes that helps identify essential issues in the design of intelligent LL systems. Table 10 lists categories, problems, and potential AI techniques for their solution, which we discuss in the following paragraphs.

Table 10: Potential uses of artificial intelligence for lessons learned sub-processes.

Category/Sub-process	Issue/Problem	Some Potentially Applicable AI Techniques
Interactive collection	Lesson elicitation from authors	Interactive CBR, query-driven simulation
Active scan collection	Lesson extraction from documents	Textual CBR, information extraction, query-driven simulation
Verification	Contrast, analyze	Approximate reasoning
Lesson storage	Indexing, modeling	Weighted features, objects, frames, concept maps
Passive and active dissemination	Retrieval	Case retrieval, latent semantic analysis, ontologies
Broadcasting dissemination	Summarization	Summarization filters
Executable lesson reuse	Plan adaptation	Case-based planning

Interactive collection: An intelligent computer system can be designed to interactively elicit lessons from lesson providers. If the lesson is to be represented by a fixed set of attributes, then a simple form could be used to collect this information. However, if the set of attributes required for describing each lesson varies greatly among lessons, then an interactive CBR method (*e.g.*, Aha & Breslow, 1997) may be useful for guiding the lesson author through the elicitation process (*i.e.*, through a series of prompted questions whose answers assign values to relevant attributes). One advantage of this method is that it can help avoid some standard problems with information retrieval systems (*e.g.*, how to interpret text expressions that have multiple potential meanings) by clarifying the lesson author's inputs during elicitation.

Query-driven simulation (Wildberger et al., 2000) could also assist with interactive elicitation efforts. In this method, the user has access to a knowledge base that can answer questions about data that, although not stored explicitly, can be derived by running a simulation. Query-driven simulation could likewise support automated extraction efforts, such as those performed in active collection strategies.

Active (scan) collection: Most lessons are initially recorded in (primarily unstructured) text format. However, LL systems should internally represent lessons with non-text attributes. Therefore, information extraction techniques are particularly important for locating potential lessons from an organization's documents. For example, the European Space Agency's LL system collects lessons from project summary documents and other sources of knowledge (*e.g.*, alerts and audit reports). Information extraction techniques could also help to filter lessons for active casting dissemination sub-processes, so that lessons are disseminated only to users who have expressed an interest in their topic.

Ashley (2000) has argued that other textual CBR techniques, which include standard information extraction techniques, may also be useful for extracting lessons expressed in pure text format. Examples of these techniques include representing lessons as (distributed, object-oriented) case retrieval nets, and using latent semantic analysis to automatically generate indices for lessons.

Verify: Although not always considered typical or appropriate, lessons are also used to reveal potential improvements in managing decision processes. Vandeville & Shaikh (1999) propose an approach for analyzing lessons, using approximate reasoning techniques, to improve project management.

Store: The choice of how to index and represent lessons is based in part on their retrieval method. These representations may include various data structures that are popular in AI (*e.g.*, weighted features, objects, frames, discrimination nets, concept maps, conceptual graphs, hierarchical task networks).

Passive and active dissemination: Retrieval methods that can support these sub-processes include, for example, various case retrieval and text indexing approaches such as latent semantic analysis (Strait et al., 2000). Domain-specific ontologies (Eilerts & Ourston, 2000) can assist here by providing details on domain-specific similarity measures.

Broadcasting dissemination: Mani et al. (2000) demonstrated how text processing techniques can be used to automatically summarize lessons to produce LL reports, which can then be shared with members of an organization.

Executable reuse: Automatically applying the suggestion of a lesson requires modeling it using the same representation used in the decision-making process targeted by the lesson. For example, we ensured this property in our proposal for an active lessons delivery system for planning tasks (Weber et al., 2000).

These suggestions only begin to touch on opportunities for using AI techniques in LL systems; many other potential synergies exist. For example, machine learning approaches could be used to process user feedback on how to apply a lesson's suggestion, which could then be used to control its application when the user executes it in future decision-making contexts.

6.2 Embedded intelligent lessons learned systems

Although there appear to be benefits of using active, proactive, or reactive dissemination sub-processes, the only LL systems that use them are research prototypes. Their common theme is that they are embedded in the decision support systems that their lessons target. Although embedded LL systems are only a recent topic of AI research, it is gaining popularity. This subsection summarizes AI research on embedded dissemination strategies for LL systems.

6.2.1 Task-specific lessons learned systems

Task-specific LL systems are not organizational; they collect and disseminate lessons for a specific task. For example, CALVIN (Leake et al., 2000) captures lessons concerning searches for relevant (online) information resources on specific research topics. The subject and research results are used to index lessons. The underlying methodology for storing, disseminating, and reusing lessons is CBR. In addition to its other innovations, CALVIN uses *concept maps* to visualize knowledge artifacts related to its lessons, which is a promising approach for representing an organization's systemic view (Senge, 1990).

6.2.2 Architectures for embedded lessons learned systems

Active lessons delivery architectures are an approach for lesson dissemination, and perhaps other LL processes, in which a LL system is implemented as a module of a decision support system (e.g., Weber et al., 2000). By monitoring targeted decision-making processes, and possibly user interactions, these systems can automatically notify users of potentially relevant lessons.

An interesting example of an embedded architecture for knowledge artifacts is the Air Campaign Planning Advisor (ACPA) (Johnson et al., 2000), which has been applied to the task of air campaign planning and has been integrated with a deployed military planning tool (*i.e.*, the Joint Planning Tool). ACPA is composed of a web-based ASK system linked to a performance support tool through a task model-based task tracking system. The goal of this integration is to prompt the user with relevant planning knowledge on an as-needed basis.

ACPA is triggered by monitoring the progress and the problems encountered by the user. It supports two modes of dissemination: proactive and reactive. ACPA responds when a user asks for help (reactive). It also responds when the system identifies potential problems in a user's evolving plan (proactive) that can be addressed by a relevant story. These stories are stored as related sets of video clips (and associated text) that have been recorded by domain experts.

Developed independently, the Active Lessons Delivery System (ALDS) (Weber et al., 2000) is closely related to ACPA. ALDS employs a different lessons delivery approach that was demonstrated for authoring deliberative plans for noncombatant evacuation operations. ALDS monitors an evolving plan scenario as it interacts with its user, and brings lessons to their attention when they are deemed relevant to their current focal task. It also highlights other tasks for which lessons become applicable during the planning process. Thus, ALDS implements an active dissemination sub-process.

Although they have similar philosophies (*i.e.*, embedded, active knowledge artifact delivery, which is also shared by CALVIN), ACPA and ALDS have several differences. ACPA manages a corporate memory that stores best practices, links between stories and their examples, and other information. In contrast, ALDS manages only lessons, leaving other modules in a multi-modal plan authoring system (HICAP) to manage other information (*e.g.*, plan refinement cases). The main advantage of specializing in lessons is the benefit from specific solutions tailored to the particular idiosyncrasies of such artifacts. Next, ACPA tracks the user's interface actions to build a model that infers their intent, which can greatly help to determine when knowledge artifacts are relevant to a user's actions. ACPA does not reason on the stories nor highlight reuse components or conditions. Stories are presented to users that have to interpret the stories and then must decide whether to apply the lesson captured in the story according to its interpretation and without any specific guidance. In contrast, ALDS is implemented as a module in the plan authoring tool HICAP (Muñoz-Avila et al., 1999). HICAP authors plans decomposing complex tasks into primitive ones depending on the user interactions and the state of the world. ALDS supervises changes in the plan and in the state, triggering lessons when the tasks in the plan and the state match a lesson's applicable task and conditions. ALDS prompts users with lessons whose *suggestion* can be implemented with the press of a button, which reuses embedded knowledge to change the targeted process. Therefore, while ACPA provides a rich environment for retrieving artifacts and related information, it supports only browsable reuse. ALDS supports executable reuse, allowing users to optionally incorporate a lesson's suggestion directly into the plan they are constructing.

6.2.3 Accessing lessons

Two systems demonstrate alternative ways to embed LL systems in decision support processes so as to enhance lesson access. First, the Department of Energy's deployed Automated Job Hazard Analysis (AJHA) (Bickford, 2000a) system, which manages lessons that focus on safety issues, employs a passive LL dissemination module that allows users to access lessons via a hyperlink within AJHA.

Another example is Cool Air (Watson, 2000), a CBR system whose cases are installations of heating, ventilation, and air conditioning (HVAC) equipment. Cool Air has been deployed to assist Western Air technicians. Implemented on the Internet, technicians input problem specifications to Cool Air and retrieve cases describing potential HVAC designs. Along with these designs, Watson described how a research demonstration variant of Cool Air also returns a set of associated *trouble tickets* (*i.e*., technical lessons learned by previous users concerning the stored design) with its cases. Thus, it supports a process that is similar to active dissemination in that it associates lessons, on a fixed basis, with existing <design,configuration> pairs. This demonstrates the utility of simpler approaches for active lesson dissemination. In combination with case retrieval, it is a general yet powerful approach for disseminating technical lessons that can be deployed in a large variety of organizations.

6.3 Technological barriers to developing intelligent LL systems

Technological barriers for developing intelligent lessons learned systems relate primarily to the availability and use of hardware, programming languages, platforms, and AI technologies. Standalone, passive LL systems are limited by several assumptions about end-users (see Section 4.1) even when enhanced with intelligent techniques (e.g., case retrieval tools). One major requirement is that users utilize software tools to perform daily tasks. Consequently, LL systems could be incorporated into these software tools, thus avoiding standalone architectures. For example, while Enterprise Resource Planning (ERP) systems have been criticized for their complexity and high costs, ERP is *"now considered the price of entry for running a business and, at least at present, for being connected to other enterprises in a network economy"* (Kumar & van Hillegersberg, 2000). Furthermore, ERP systems have become the platform for implementing intelligent applications such as data mining and decision support systems. Thus, it is easy to envision that the LL systems could also be implemented in ERP platforms.

Integrating LL systems in military organizations could involve using portable devices such as laptops, palmtops, hand-held, and wearable computers. These portable devices could make it possible the access LL systems in remote sites, or even on the battlefield. For example, at the DOE the primary function of the manager is to plan job processes. Thus, even if the employees who might benefit from lessons do not use software tools in their job/task, it may be possible for the job manager to use a LL system and communicate the appropriate lessons to them. For embedded architectures, the requirement will be that the LL module can be integrated within the targeted software tool.

The selection of the programming language to implement an LL system also defines a potential technological barrier for integration. This has been observed through the development of intelligent LL systems (e.g., ALDS, CALVIN, and HVAC) and the subsequent decision to use Java , primarily due to its support for Internet applications. Another important issue in the design and implementation of LL systems concerns security. For example, military systems have to be compatible with SIPRNET. As defined in Section 4.2, LL repositories can be classified, unclassified, or restricted. Security, political, legal, and access issues have been discussed by Mackey & Bagg (1999) and in Secchi (1999). Finally, AI techniques also pose technological limitations. For example, understanding and correctly interpreting natural language still remains a challenging problem.

7 Conclusion

We conducted a survey on lessons learned (LL) systems after obtaining relevant information online, from relevant publications, and through interviews. The limited impact of LL systems in the surveyed organizations motivated us to investigate these systems and the potential benefits that AI can bring to improve their ability to promote knowledge sharing.

In this article, we introduced a categorization of LL systems that can be used to investigate the suitability of AI techniques. Our survey reinforced that the two most evident problems contributing to the ineffectiveness of LL systems concern text representations for lessons and their standalone design.

The KM literature (Reimer, 1998; Aha et al. 1999; Leake et al., 2000) is unanimous in stressing how KM applications should be incorporated into the processes they intend to support. In

Section 6 we summarized proposed approaches for implementing embedded architectures that account for this design goal.

Text formats are troublesome for computational treatment, and attempts to create structure in records have rarely addressed core issues, such as highlighting the reuse component of a lesson. In Section 5.1, we proposed a representation for planning lessons where we define six attributes corresponding to the minimum components of a planning lesson: originating action, conditions, contribution, result, suggestion, and applicable task.

We also surveyed LL systems according to two sets of criteria. First, because these systems are designed to support LL processes, they vary in accordance to the methods they use for implementing the LL sub-processes that we summarized in Section 4.1 (*i.e.*, collect, verify, store, disseminate, and reuse). Second, we also discussed several system characteristics not directly related to LL sub-processes (i.e., content, role, orientation, duration, organization type, architecture, representation, confidentiality, and size), which we described in Section 4.2.

Finally, we surveyed AI techniques that have been proposed for improving performance LL system performance, focusing on how they address specific problematic issues we highlighted in our survey. We are confident that, after further exploration, some of these techniques will benefit deployed lessons learned systems in the future.

Acknowledgements

This research was supported by grants from the Office of Naval Research and the Naval Research Laboratory. Thanks to Bob Lucas, Karl Branting, Héctor Muñoz-Avila, John Bickford, and William Mackey for relevant comments and contributions, and members of the organizations we interviewed for discussions and contributions related to this survey. Thanks also to André Testa for his assistance with creating Figure 1.

References

Aha, D.W., Becerra-Fernandez, I., Maurer, F., and Muñoz-Avila, H. (Eds.). (1999). *Exploring Synergies of Knowledge Management and Case-Based Reasoning: Papers from the AAAI 1999 Workshop* (Tech. Rep. WS-99-10). Menlo Park, CA: AAAI Press.

Aha, D.W., & Breslow, L.A. (1997). Refining conversational case libraries. *Proceedings of the Second International Conference on Case-Based Reasoning* (pp. 267-278). Providence, RI: Springer.

Aha, D.W., & Weber, R. (1999). Lessons learned links. Available at [http://www.aic.nrl.navy.mil/~aha/lessons].

Aha, D.W., & Weber, R. (2000). *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech. Rep. WS-00-08). Menlo Park, CA: AAAI Press.

Ashley, K. D. (2000). Applying Textual Case-Based Reasoning and Information Extraction in Lessons Learned Systems. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 1-4). Menlo Park, CA: AAAI Press.

Bartlett, J. (1999). Unpublished slides.

Beebe, J. (2000). *After action review*. Presentation given at the Spring 2000 DOE Society for Effective Lessons Learned Sharing (SELLS) Workshop. [http://tis.eh.doe.gov/ll/proceedings/proceedings400.htm]. Oakland, CA: Unpublished.

Bickford, J. (2000a). Compiled notes for SELLS workshop. April 5-6, 2000.

Bickford, J. (2000b). Sharing lessons learned in the Department of Energy. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 5-8). Menlo Park, CA: AAAI Press.

Davenport, T.H. & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Boston, MA: Harvard Business School Press.

DOE-STD-7501-99 (1999). *The DOE Corporate lessons learned program*. United States Department of Energy, Washington, DC. [http://www.tis.eh.doe.gov/ll/docs.html]

Everett, J.O., & Bobrow, D.G. (2000). Resolving redundancy: A recurring problem in a lessons learned system. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 12-16). Menlo Park, CA: AAAI Press.

Eilerts, E. & Ourston, D. (2000). The value of ontologies. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 9-11). Menlo Park, CA: AAAI Press.

Fisher, D., Deshpande, S., & Livingston, J. (1998). *Modeling the lessons learned process* (Research Report 123-11). Albuquerque, NM: The University of New Mexico, Department of Civil Engineering.

Gery, G. (1995). Attributes and behaviors of performance-centered systems. *Performance Improvement Quarterly, 8*, 44-93.

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

Knight, C., & Aha, D. W. (2000). A common knowledge framework and lessons learned module. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 25-28). Menlo Park, CA: AAAI Press.

Kolodner, J. (1993). *Case-Based Reasoning*. Los Altos, CA: Morgan Kaufmann,

Kühn & Abecker, (1997). Corporate memories for knowledge management in industrial practice: Prospects and challenges. *Journal of Universal Computer Science*, 3(8), 929-954.

Kumar, K., & van Hillegersberg J. (2000). ERP experiences and evolution. *Communications of the ACM*, 43(4), 23-26.

Leake, D. (1996). *Case-based reasoning: Experiences, lessons, and future directions*. Menlo Park, CA: AAAI Press.

Leake, D.B., Bauer, T., Maguitman, A., & Wilson, D.C. (2000). Capture, storage, and reuse of lessons about information resources: Supporting task-based information search. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 33-37). Menlo Park, CA: AAAI Press.

Mackey, W. & Bagg, T. (1999). System Issues Related to Implementation the Internet. In *Proceedings of the Ninth Annual International Symposium of the International Council on Systems Engineering* (pp. 679-686). Brighton, England: INCOSE.

Mahe, S., Rieu, C., Beauchene, D. (1996). An original model to organize know-how in a benchmarking context. *Proceedings of the Tenth Banff Workshop on Knowledge Acquisition for Knowledge Based Systems*. [http://ksi.cpsc.ucalgary.ca/KAW/KAW96/mahe/maherieu.html]

Mani, I., Concepcion, K., & van Guilder, L. (2000). Automating briefing production for lessons learned systems. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 43-45). Menlo Park, CA: AAAI Press.

Muñoz-Avila, H., McFarlane, D., Aha, D.W., Ballas, J., Breslow, L.A., & Nau, D. (1999). Using guidelines to constrain interactive case-based HTN planning. In *Proceedings of the Third International Conference on Case-Based Reasoning*, (pp. 288-302). Munich: Springer.

Nemoto, M., Fukuchi, Y., Suzuki, K., Matsuda, T., & Miyachi, Y. (1999). Data collection and report preparation of the NASDA KYO-KUN system. In P. Secchi (Eds.), *Proceedings of Alerts and LL: An Effective way to prevent failures and problems*. (Technical Report WPP-167) (pp. 75-78). Noordwijk, The Netherlands: ESTEC.

O'Leary, D. E. (1998). Enterprise Knowledge Management. *Computer Magazine*, 31, 3, 54-61. [http://www.intraware.com/ms/mktg/indaa/evaltch/EntKnowMngt.html]

O'Leary, D. E. (1999). Knowledge Management for best practices. *Intelligence: New Visions of AI in Practice*, 4(10), 12-24.

OPNAVINST (1996). *Universal Naval Task List*. 3500.38/MCO 3500.26/USCG COMDTINST M3500.1. 30 SEPTEMBER 1996. [http://navmsmo.hq.navy.mil/nmsiscat/index.cfm?RID=DTS_N_1000779]

Reimer, U. (Ed.) (1996). *Papers from the 1996 Workshop on The Lessons Learned Cycle: Implementing a Knowledge Pump in your Organisation*. *First International Conference on Practical Aspects of Knowledge Management*. [http://www.xlog.ch/pakm/procdocu.html].

Reimer, U. (1998). Knowledge integration for building organisational memories. In *Proceedings of the Eleventh Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*. Banff, Canada: Unpublished.

Sampson, M. (1999). NASA parts advisories – nine years of experience, and counting. In P. Secchi (Eds.), *Proceedings of Alerts and LL: An Effective way to prevent failures and problems*. (Technical Report WPP-167) (pp. 31-35). Noordwijk, The Netherlands: ESTEC.

Sary, C., & Mackey, W. (1995). A case-based reasoning approach for the access and reuse of lessons learned. *Proceedings of the Fifth Annual International Symposium of the National Council on Systems Engineering* (pp. 249-256). St. Louis, Missouri: INCOSE.

Secchi, P. (Ed.) (1999). *Proceedings of Alerts and LL: An Effective way to prevent failures and problems* (Technical Report WPP-167). Noordwijk, The Netherlands: ESTEC.

Secchi, P., Ciaschi, R., & Spence, D. (1999). A Concept for an ESA lessons learned system. In P. Secchi (Eds.), *Proceedings of Alerts and LL: An Effective way to prevent failures and problems* (Tech. Rep. WPP-167) (pp. 57-61). Noordwijk, The Netherlands: ESTEC.

SELLS (1999). *Proceedings of the Society for Effective LL Sharing's Spring Meeting.* Las Vegas, NV: Unpublished. [www.tis.eh.doe.gov/ll/sells/proceedings399.htm]

SELLS (2000). *Proceedings of the SELLS Spring Meeting April 5-6, 2000.* Oakland, CA: Unpublished. [tis.eh.doe.gov/ll/proceedings/proceedings400.htm]

Senge, P. (1990). *The fifth discipline: The art and practice of a learning organization*. New York: Doubleday.

Siegel, A. (2000). Private communication.

Stein, E.W. (1995). Organizational memory: Review of concepts and recommendations for management. *International Journal of Information Management*, 15(2), 17-32.

Stewart, T. A. (1997). *Intellectual capital: The new wealth of organizations*. New York: Currency Doubleday.

Strait, M. Haynes, J., & Foltz, P.W. (2000). Applications of Latent Semantic Analysis to Lessons Learned Systems. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 51-53). Menlo Park, CA: AAAI Press.

Tautz, C., Althoff, K. D., & Nick, M. (2000). A case-based reasoning approach for managing qualitative experience. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 54-58). Menlo Park, CA: AAAI Press.

Tulak, Maj.A. N. (1999). Methodologies for Collecting Lessons Learned. *Joint Center for Lessons Learned Bulletin*, 1 (1). USACOM JWFC-15.

van Heijst, G., van der Spek, R., & Kruizinga, E. (1996). Organizing corporate memories. *Proceedings of the Tenth Banff Workshop on Knowledge Acquisition for Knowledge Based Systems*. [http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96Proc.html]

Vandeville, J.V. & Shaikh, M. A. (1999) A structured approximate reasoning-based approach for gathering "lessons learned" information from system development projects. *Systems Engineering,* 2(4), 242-247.

Watson, I. (2000). A Distributed CBR application for engineering sales support. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 59-62). Menlo Park, CA: AAAI Press.

Weber, R., Aha, D.W., Branting, L.K., Lucas, J.R., & Becerra-Fernandez, I. (2000). Active case-based reasoning for lessons delivery systems. In *Proceedings of the Thirteenth Annual Conference of the International Florida Artificial Intelligence Research Society* (pp. 170-174). Orlando, FL: AAAI Press.

Wildberger, A. M., Gross, M. M., & Ayres, T.J. (2000). Learning lessons intelligently in the electric power industry. To appear in D.W. Aha & R. Weber (Eds.), *Intelligent Lessons Learned Systems: Papers from the AAAI Workshop* (Tech.Rep. WS-00-08) (pp. 68-72). Menlo Park, CA: AAAI Press.

Appendix: Glossary of Acronyms

ACPA	Air Campaign Planning Advisor (Johnson et al., 2000)
AFCKS	Air Force Center for Knowledge Sharing
AI	Artificial intelligence
ALDS	Active Lessons Delivery System (Weber et al., 2000)
ALLCARS	Automated LL Collection and Retrieval System
AMEDD	US Army Medical Lessons Learned
CALL	Center for Army Lessons Learned
CBR	Case-based reasoning
CNES	Centre National d'Etudes Spatiales
COIN	Corporate Information Network
DOE	Department of Energy
EM	Environmental Management
ESA	European Space Agency
ESH	Environment, Safety and Health
HVAC	Heating, ventilation, and air conditioning
INS	Immigration and Naturalization Service
JCLL	National Space Development Agency of Japan
JTF	Joint Task Force
NAWCAD	Navy Air Warfare Center, Aircraft Division
NAWCAD/CALL	NAWCAD's Combined Automated Lessons Learned
NLLS	Navy Lessons Learned System
NSA	National Security Agency
RECALL	Reusable Experience with CBR for Automating LL
SELLS	The DOE's Society for Effective Lessons Learned Sharing
SIPRNET	Secure Internet Protocol Network
WWW	World Wide Web