

Categorizing Intelligent Lessons Learned Systems¹

Rosina Weber¹, David W. Aha², Irma Becerra-Fernandez³

¹Department of Computer Science, University of Wyoming, Laramie, WY 82071

²Navy Center for Applied Research in AI, Naval Research Laboratory, Washington, DC 20375

³Florida International University, Decision Sciences and Information Systems, Miami, FL 33199

{weber,aha}@aic.nrl.navy.mil, becferi@fiu.edu

Abstract

Lessons learned systems are knowledge management solutions that serve the purpose of capturing, storing, disseminating and sharing an organization's verified lessons. In this paper we propose a two-step categorization method to support the design of intelligent lessons learned systems. The first step refers to the categories of the lessons learned processes the system is designed to support. The second step refers to the categories of the system itself. These categories are based on systems available online and described in the literature. We conclude by summarizing representation and other important issues that need to be addressed when designing intelligent lessons learned systems.

Motivation and definition

Lessons learned (LL) systems have been deployed by many military, commercial, and governmental organizations¹ to disseminate validated experiential lessons. They support organizational lessons learned processes, which use a knowledge management (KM) approach to collect, store, disseminate, and reuse experiential working knowledge that, when applied, can significantly benefit targeted organizational processes (Davenport & Prusak, 1998). Unfortunately, based on our interviews and discussions with members of several LL centers (e.g., at the Joint Warfighting Center, the Department of Energy (DOE), the Naval Facilities Engineering Command, Goddard Space Flight Center (NASA), the Construction Industry Institute), we learned that LL systems, although well-intentioned, are rarely used.

Our goal is to design, develop, evaluate, and deploy LL systems that support knowledge sharing. In this paper, we categorize LL systems and identify some pertinent research directions that may benefit from applying artificial intelligence (AI) techniques.

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 Lessons Learned Center and the Secretary of the Army -

Research, Development, and Acquisition (SARDA) among others, still use this notion. Today, this concept has evolved because organizations working towards improving the results obtained from LL systems have adopted **binding criteria** (e.g., lessons have to be **validated** for correctness and should **impact** organizational behavior). This definition is now used by the American, European, and Japanese Space agencies:

A lesson learned is 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...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, 1999).

Lessons learned, as well as other KM artifacts, are usually described with respect to their origin, application, and results. Table 1 contrasts artifacts of frequent interest in KM strategies.

Table 1. Contrasting knowledge management artifacts.

| Artifacts | Lessons Learned | Incident Reports | Alerts | Best Practices |
|------------------------------|-----------------|------------------|----------|----------------|
| Originates from Experiences? | Yes | Yes | Yes | Possibly |
| Whole processes? | No | No | No | Yes |
| Describes failures? | Yes | Yes | Yes | No |
| Describes successes? | Yes | No | No | Yes |
| Orientation | Organization | Organization | Industry | Industry |

Categorizing Lessons Learned Processes

Our thesis is that LL systems exist to support organizational lessons learned processes. Figure 1 depicts the essential components of a generic LL process. We developed this model by examining how several organizations have deployed and utilized LL systems.

Most organizations produce LL processes to communicate how lessons are to be acquired, verified, and disseminated (e.g., Sells, 1999; Fisher et al., 1998; van Heijst 1996; Secchi, 1999; Verdin, 1999). Given that it is an

1. In D.W.Aha & R. Weber (Eds.) (2000). Intelligent Lessons Learned Systems: Papers from the AAAI 2000 Workshop , 63-67. (Technical Report WS-00-008). Menlo Park, CA:AAAI Press. ¹

Our WWW page, www.aic.nrl.navy.mil/~aha/lessons, contains additional information on the organizations mentioned in this paper.

organizational process, it involves both human and technological issues. In this paper, we limit our scope to technological issues.

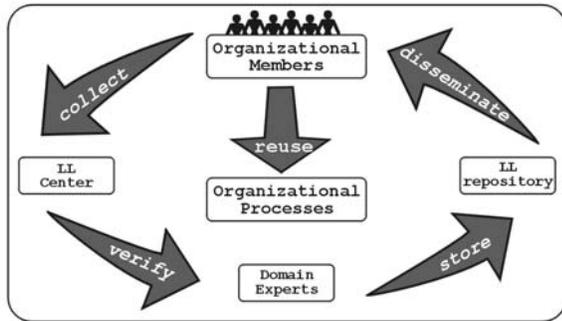


Figure 1. Lessons learned process.

According to our model, LL processes implement a strategy for reusing experiential knowledge necessary to support an organization's goals. LL systems can be categorized in accordance with the subset of the five sub-processes that they support, namely *collect*, *verify*, *store*, *disseminate*, and *reuse*.

Collect: This sub-process can be performed by at least, five categories of methods:

1. **Passive Collect:** Organization members submit their own lessons using a form (e.g., online) in 2/3 of the organizations surveyed. The Center for Army Lessons Learned (CALL) has an excellent passive collection form with online help and examples.
2. **Reactive Collect:** Lessons are obtained through interviews. Used by the National Space Development Agency of Japan (NASDA), and also described in Vandeville & Shaikh (1999) and Tautz et al. (2000).
3. **After Action Collect.** Typically used in military organizations such as the Air Force and Joint Center for Lessons Learned. This highlights the distinction of when lessons are collected. Because of this, different organizations can benefit from a collection during or near the completion of a project (Vandeville & Shaikh, 1999).
4. **Active Collect:** Two different collect methods are called *active*. *Active Scan* takes place when communication processes are scanned to search for lessons (Knight & Aha, 2000). In contrast, the *Military Active* collect starts by identifying a set of problems that demand solution and a collection procedure composed of four phases: mission analysis and planning, deployment and unit link-up, collection operations, redeployment and writing the report (implemented by CALL).
5. **Interactive Collect:** 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 Unified Naval Task List). In LL systems designed for training purposes, verification can be used to combine and/or 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). Task-relevant representations, such as the DOE's categorization by safety priority, are also often used.

Disseminate: We have identified five categories of the dissemination sub-process:

1. **Passive dissemination:** Users search for lessons in a (usually) standalone retrieval tool.
2. **Active casting:** In this method, adopted by the DOE, lessons are broadcasted to potential users via a dedicated list server.
3. **Active dissemination:** Users are pro-actively notified of relevant lessons, as exemplified by systems described by Weber et al. (2000) and Leake et al. (2000).
4. **Proactive dissemination:** The system builds a model of the user's interface events to predict when to prompt with relevant lessons. This approach is used to disseminate videotaped stories in the Air Campaign Planning Advisor (ACPA) as described by Johnson et al. (2000) and by Microsoft (Gery, 1995).
5. **Reactive dissemination:** When users realize they need additional knowledge, they can invoke a help system to obtain relevant lessons. This is used in the Microsoft Office Suite and in ACPA.

Reuse: We have identified four categories of reuse sub-processes:

1. **Browsable recommendation:** The system displays a retrieved lesson's recommendation that the user is able to read and copy.
2. **Executable recommendation:** User can optionally execute a retrieved lesson's recommendation. Proposed in the ALDS architecture (Weber et al., 2000), this capability requires embedding in a decision support software tool.
3. **Learning recommendation:** New users can input alternative applications for the lesson.
4. **Outcome reuse:** This involves recording the outcome of using a lesson, which can help to identify a lesson's utility. In the Lockheed Martin's Oak Ridge LL system, LL coordinators are expected to identify actions taken or planned relative to given lessons.

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.

Categorizing Lessons Learned Systems

This section describes several ways for categorizing lessons learned systems.

Content: Because lessons are not the only KM artifact designed for reuse, some organizations will use similar collection, verification, storage, dissemination, and reuse processes for objects such as incident reports or alerts. **Pure** LL systems only manipulate lessons; they represent 40% of LL systems surveyed (e.g., the Air Combat Command Center for LL, Air Force Center for Knowledge Sharing Lessons Learned (AFCKSLL), U.S. Army Medical LL (AMEDD), Joint Center for LL (JCLL), Automated LL Collection And Retrieval System (ALLCARS), and Reusable Experience with Case-Based Reasoning for Automating LL (RECALL)). **Hybrid** systems (60%) also include other knowledge artifacts (e.g., the DOE Corporate Lessons Learned Collections also store alerts, best practices and facts).

Role: LL systems differ according to the nature of the processes (*roles*) they support. Those supporting a technical role may include solutions to novel problems that do not correspond to past experience (e.g., Xerox's EUREKA system (Everett, 2000)). Military repositories are often used in planning roles.

Purpose and scope: An LL system's organizational objective is to share knowledge that is relevant to the organization's goals. The organization's purpose defines the scope of the LL system. For example, when the French Space Agency (CNES) states that the purpose of their LL system is to improve competitiveness, it means that a lesson that does not generate an impact in the agency's competitiveness is outside the scope of their system. Some LL systems support a group of organizations. For example, the European Space Agency maintains a system for its community. On the other hand, while other systems such as CALVIN (Leake et al., 2000) have a task-specific purpose, in this case to share lessons on how to perform research on a generic task.

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: Some organizations are adaptable and can quickly incorporate lessons learned in their processes, while others employ rigid doctrine that is only slowly updated. Rigid organizations can employ LL systems to gather grounds for modifying and generating doctrine.

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, 2000).

Attributes and Format: Most lessons (90%) combine textual and non-textual attributes. They are originally collected in text format and supplemental fields to provide structure and clusters are added.

Confidentiality: Lessons can be classified, unclassified, or restricted. For example, the United States Air Force (USAF)'s Center for Knowledge Sharing Lessons Learned provides Internet access to unclassified lessons and Secret Internet Protocol Router Network (SIPRNET) (<http://knowledge.langlely.af.smil.mil>) access to classified lessons. The Internet site also provides access to classified lesson titles, along with appropriate links to the SIPRNET site.

Categorizing LL systems can have many benefits. For example, knowing the system's purpose and scope will help to judge whether a search is appropriate. In addition, LL system categories can guide the design of improved systems. Mackey and Bagg (1999) describe other criteria and guidelines for LL systems.

Lessons learned representation

We have analyzed the lessons stored in surveyed repositories, searching for clues on how to maximize reuse based upon the lesson's representation. We first focus on lessons in the planning role, searching for components in these lessons to identify what is the minimum relevant information they embed that is satisfactory to enable reuse.

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. 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. The components of a planning lesson are:

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

Result: The result of the originating action: positive or negative. Lessons can be originated from a failure or from a successful result.

Lesson Contribution: is the element that is combined with the originating action that is considered responsible by the result, the particular element combined with or characterizing the action that is responsible for the success of this originating action. Lesson contribution can be a method, a resource, the inclusion of an element to a checklist, or the review of a document where you can identify important aspects to consider.

Applicable task: The lesson author, as a domain expert, is able to identify to which task (decision or process) the lesson is applicable. The applicable task may not be limited to one. New users should suggest new tasks when identifying applicability.

Conditions: Defines 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. One or two conditions may be absent and the lesson will still be valid. If none of the conditions hold, then it is very likely that the lesson will not be applicable.

Suggestion: It is an interpretation of the lesson and its use to the applicable task: the recommended action is to reuse the lesson.

This representation reduces the lesson content to a minimum. Additional comments can be kept in textual fields to be shown to the user. This minimum representation makes lessons learned suitable to computational use and thus facilitating the reuse of lessons. Experiences can be easily converted into lessons. For example, consider lessons originated from successful experiences, and follow the sequence: *“identify the lesson contribution (what particular element combined to or characterizing the action that is responsible for the success of this originating action) and repeat the originating action making sure you repeat the lesson contribution.”*

We illustrate this representation with a lesson from the Joint Unified Lessons Learned System (<https://www-secure.jwfc.acom.mil/protected/jell>) concerning noncombatant evacuation operations. The selected lesson's summary is: *"The triple registration process was very time consuming and contributed significantly to delays in throughput and to evacuee discomfort under tropical conditions in CEBU."* Our representation for this lesson is:

Originating action: implement the triple registration process (register evacuees with the triple registration process)

Action result: time consuming and evacuee discomfort: negative.

Contribution: triple registration process is problematic.

Applicable task: evacuee registration.

Conditions: under tropical conditions.

Suggestion: Avoid the triple registration process when registering evacuees. Locate an INS (Immigration and Naturalization Service) screening station at the initial evacuation-processing site. Evacuees are required to clear INS procedures prior to reporting to the evacuation-processing center.

The expression ‘under tropical conditions’ is a condition for reuse. In this lesson, the *applicable task* repeats the *originating action*, although this is not necessarily true for

all lessons. This lesson refers to a negative *result* (e.g., evacuee discomfort). When the *result* is negative, the lesson *contribution* should be avoided. Thus, the first statement of *suggestion* is inferred. The alternative method is provided in the lesson text. The use of ‘suggestion’ instead of ‘recommendation’ is due to customer’s feedback (Sampson, 1999).

The technical role is defined by a technical context; accordingly, technical lessons are typically delivered through jobs or projects. We have learned by repositories such as EUREKA (Everett, 2000) and HVAC LL system (Watson, 2000) that many technical lessons are expressed in terms of a triplet: problem-cause-solution. These examples indicate at least one representative type of technical lessons. (a little awkward) Nonetheless, further research on technical lessons might indicate other types that can demand for a different representation.

Open issues in designing intelligent lessons learned systems

This paper addresses categorizations to help identifying an adequate design methodology for intelligent LL systems. Besides the issues related to categorization, our research has also pointed to some other relevant issues that deserve attention.

Level of abstraction of the lessons: This issue refers to the generality vs. specificity required to guarantee the reuse of lessons. Generic lessons have higher recall but less precision (Weber, 1998 c.4). In CBR systems, generic lessons increase the need for adaptation.

Author’s contact information: Some LL centers (2/3) choose to disclose the contact information of the lesson author, while others (ALLCARS, RECALL) consider the anonymous contribution to be more reliable. This issue also has to be decided at the organizational level. The major drawback in disclosing author’s identity relates to concerns of the lesson to be used as an evaluation parameter.

Obsolescence: It becomes an issue when systems gain robustness that can increase retrieval time or decrease precision. At the organizational level, obsolescence of lessons should be discussed in the verification sub-process.

Textual lessons: The European Space Agency lessons learned system (ELLS) maintains an active (scan) collect of lessons from documents produced after projects and in other sources of knowledge such as alerts and audit reports. Their task is to extract relevant lessons learned from these documents. Document-based techniques are also useful for filtering in an active casting disseminate. For example, Ashley (2000) discusses use of textual CBR and information extraction to aid the use of textual lessons.

Information Retrieval: Besides case-based retrieval, two other methods that can improve retrieval are: the use of ontologies, discussed in (Eilerts & Ourston, 2000), and Latent Semantic Analysis, described in Strait (2000).

Maintenance: A military report on LL systems (GAO, 1995) suggests the use of subject areas to deal with specific problems claiming it helps to identify trends in performance weaknesses, facilitating maintenance.

References

- Ashley, K. 2000. Applying Textual Case-Based Reasoning and Information Extraction in Lessons Learned Systems. To appear in (Aha & Weber, 2000).
- Bickford, J.C. 2000. Sharing Lessons Learned in the Department of Energy. To appear in (Aha & Weber, 2000).
- Davenport, T.H., and Prusak, L. 1998. *Working knowledge: How organizations manage what they know*. Boston, MA: Harvard Business School Press.
- Eilerts, E. and Ourston, D. 2000. The Value of Ontologies. To appear in (Aha & Weber, 2000).
- Everett, J.O. and Bobrow, D. 2000. Resolving Redundancy: An Unsolved Problem in Lessons Learned Systems. To appear in (Aha & Weber, 2000).
- 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.
- GAO 1995. *Military Training—Potential to use Lessons learned to avoid past mistakes is largely untapped*. Report to the Chairman, Subcommittee on Military Personnel, Committee on National Security, House of Representatives. US General Accounting Office, DTD August, 95.
- Gery, G. (1995). Attributes and behaviors of performance-centered systems. *Performance Improvement Quarterly*, 8, pp. 44-93.
- Johnson, C.; Birnbaum, L.; Bareiss, R. & Hinrichs, T. War Stories: Harnessing Organizational Memories to Support Task Performance. *Intelligence*, 11, 1. ACM SIGART. Spring 2000
- Knight, C. & Aha, D. W. 2000. A Common Knowledge Framework and Lessons Learned Module. To appear in (Aha & Weber, 2000).
- Leake, D. B.; Bauer, T.; Maguitman, A. and Wilson, D. C. 2000. Capture, Storage and reuse of lessons about information resources: supporting task-based information search. To appear in (Aha & Weber, 2000).
- Mackey, W. & Bagg, T. 1999. System issues related to implementation the Internet. Proceedings of the Ninth Annual International Symposium of the International Council on Systems Engineering, June 1999, *Brighten England*, 1, pp. 679-686.
- Sampson, M. (1999). NASA parts advisories – nine years of experience, and counting, 31-35. In (Secchi, 1999).
- 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*, 249-256. St. Louis, Missouri: NCOSE.
- Secchi, P. (Ed.) (1999). *Proceedings of Alerts and Lessons Learned: 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, 57-61. In (Secchi, 1999).
- SELLS. 1999. *Proceedings of the Society for Effective Lessons Learned Sharing* (SELLS). Spring Meeting. Las Vegas, NV: Unpublished. URL: www.tis.eh.doe.gov/ll/sells/proceedings399.htm
- Stewart, T. A. 1997. *Intellectual Capital: the new wealth of organizations*. New York: Currency Doubleday.
- Strait, M.; Haynes, J. and Foltz, P. W. 2000. Applications of latent semantic analysis to lessons learned systems. To appear in (Aha & Weber, 2000).
- van Heijst, G., van der Spek, R., & Kruizinga, E. (1996). Organizing corporate memories, in *Proc. 10th Banff Workshop on Knowledge Acquisition for Knowledge-Based Systems*, 1996. URL: <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96Proc.html>
- Vandeville, J.V. and 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.
- Tautz, C.; Althoff, K.D. and Nick, M. 2000. A Case Based Reasoning Approach for Managing Qualitative Experience. To appear in (Aha & Weber, 2000).
- Verdin, R. 1999. Elaborating an efficient experience feedback process, 65-70. In (Secchi, 1999).
- Watson, I. 2000. Lessons learned during HVAC installation. To appear in (Aha & Weber, 2000).
- Weber, R. 1998. Intelligent Jurisprudence Research. Doctoral diss., Dept. of Production Engineering, Federal University of Santa Catarina, Brazil. URL: www.eps.ufsc.br/~rosina/html/activities.html
- Weber, R., Aha, D.W., Branting, L.K., Lucas, J.R., & Becerra-Fernandez, I. 2000. Active case-based reasoning for lessons delivery systems. To appear in *Proceedings of the Thirteenth Annual Conference of the International Florida Artificial Intelligence Research Society*. Orlando, FL: AAAI Press.