Lessons Learning System Attributes: An Analysis

by Ludwig Benner Jr and William L. Carey

A presentation about changes needed

to satisfy expectations of

lessons learned from accidents

European Safety and Reliability Data Association

36rd Annual Seminar

Coimbra Portugal 2009

Good afternoon
This is a presentation about changes needed
to learn lessons from accidents more effectively

What happened to lessons already learned?

Guam - February 23 2008

Narita - March 23 2009

New York - March 19 2008

They didn't learn them.

"This technique was **never formalized** in a technical order change or captured in **'lessons learned' reports**. Hence, only some pilots and some maintenance technicians knew of the suggestion," according to Carpenter's executive summary of the accident.

The report said, "The human factor of **communicating critical information** was a contributing factor to this mishap."

From Maj. Gen. Floyd L. Carpenter, who headed an accident investigation board. (AP News)

Another lesson not learned.

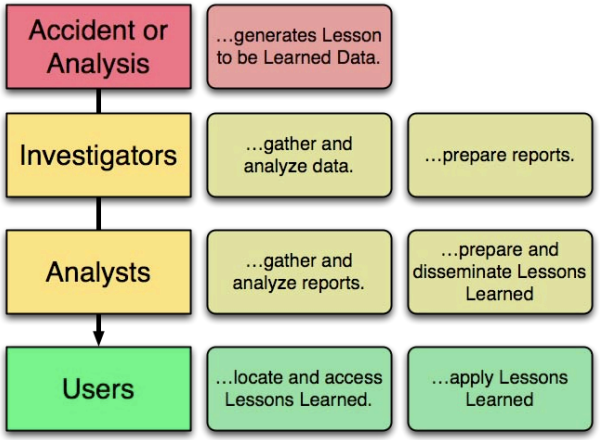
$US 1.4 Billion + lesson

We need to redesign the system.

- Lessons learned processes work poorly due to inherent design flaws.
- An alternative lessons learning system with new attributes is needed to meet users' needs successfully.
- Redesign of lessons-to-be-learned source data and lessons documentation is an essential first step to lessons learning system optimization.

Here is a summary of results of our continuing study of lessons learned from accident investigations.
Contemporary LL systems work, but poorly due primarily to inherent system design flaws
an alternative system is needed - and easier to achieve than trying to repair present processes
redesign of investigation inputs and lessons documentation is essential first step

Present Generic
Lessons Learned System Components



Here is a summary of the system description for the generic model we synthesized. The role of analysts in the system operation is especially noteworthy, and reflects a strategic system design decision, as we eventually learned.

We also found it useful to think of the data generated by an accident or incident as raw “lessons-to-be-learned” data from which lessons must be developed.

Observed Lessons Learned
System Attributes

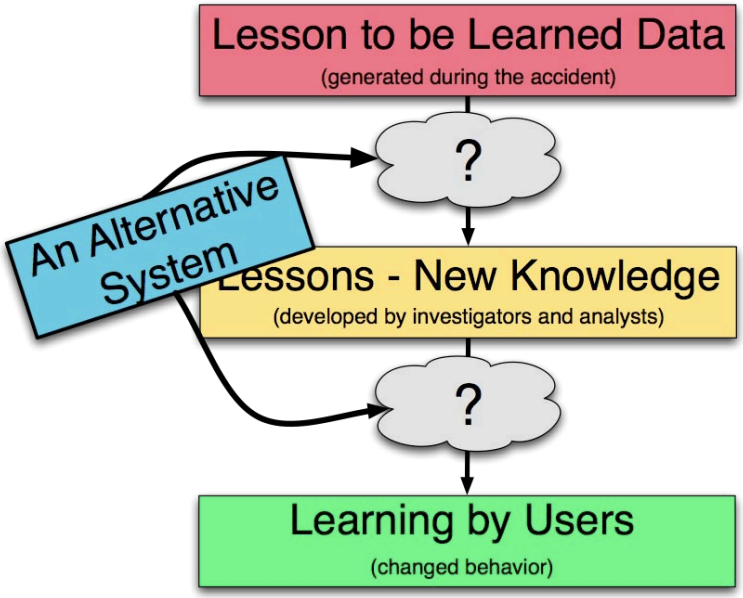
1. Divergent views of LL
2. No listing of LL by that name in reports
3. “Undisciplined” natural language inputs
4. Recommendations are proposed responses to LL
5. Causes, factors, issues etc affect taxonomies
6. Analysts select recommendations to promote
7. Recommendations assume favorable change
8. Key words may be assigned to help retrieval
9. Context buried in verbiage
10. Recommendations “pushed” to addressees
11. Assimilation by others “pulled” haphazardly
12. If used, results metrics are unstructured

Here is a list of the attributes we observed. Each is linked to previously reported examples of lessons learning impediments posed by contemporary processes. The linkages should be readily recognizable.

Lessons Learned ≠ Changed Behavior

Time for a new system?

Lessons learned processes do not produce changed behaviors very well. Widely acknowledge underperformance of present processes reflect deeply ingrained design decisions and system attributes. Therefore, we elected to try to identify a more successful system would look like.



We analyzed the functions and tasks needed to covert data generated by an accident into changed behaviors AND safer performance.
We also found we had to separate the functions of users who were the “learners” in the system from the “developers” who produced the lessons to be learned by the users.
That led us to new system boundaries.
Lessons are new knowledge - a new understanding of what happened.
Learning is the application of that knowledge or understanding to change behaviors.
Thus a lessons learning system.

Whom should the system serve?

Users.

As we analyzed the necessary system operation, to attract and engage users, the main "driver" for the lessons learning system design must be the users' perspective and resultant needs, rather that the investigators' or analysts perspectives of their own needs and outputs.

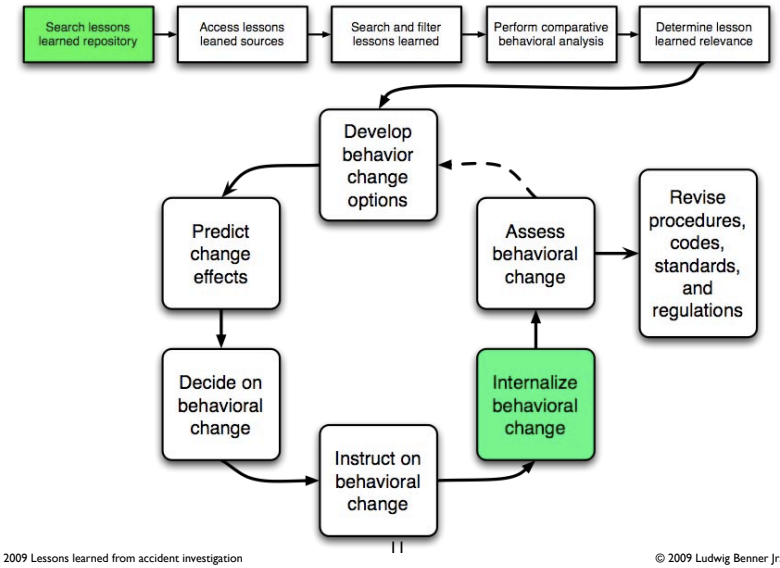
The System must be designed and optimized to serve users who can bring about changed behaviors in people, objects or energies
Design should NOT be driven by investigators' or analysts' perspectives.

What functions do users have?

- ✦ Access relevant lessons.
- ✦ Interpret the lessons.
- ✦ Change behaviors.

Users must access lessons, interpret them for relevance and applicability, and then produce the changed behavior needed

The User's Part of the System



ESReDA 36 2009 Lessons learned from accident investigation

11

Users have to do a lot. The user part of system model starts with accessing "lessons learned" in repositories, then finding relevant lessons, and then producing changed behaviors, followed by updating of repositories after success is confirmed

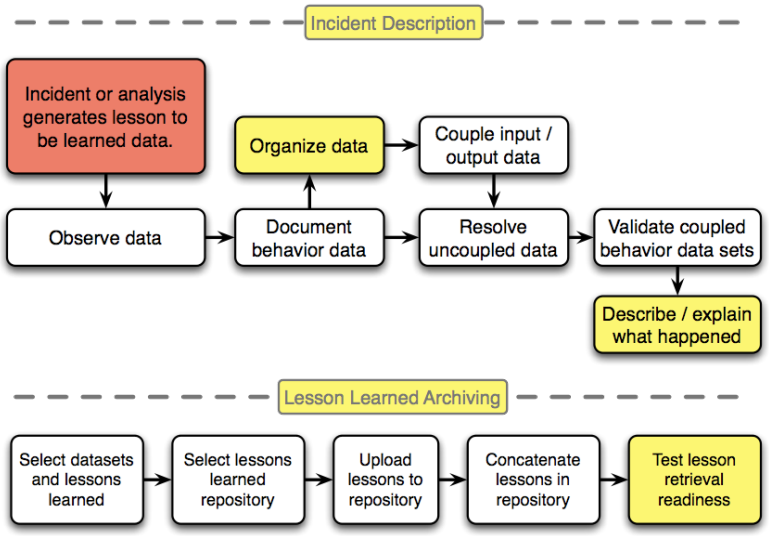
As we have since discovered, during operational decision making, some of the sequences may differ, depending upon the repository strategy adopted, but the all the components seem necessary.

What functions do developers have?

- ✦ Find lessons from Lessons to be Learned data.
- ✦ Document those lessons.
- ✦ Archive those lessons to provide access to users.

We found it helpful to distinguish between the finding and documentation of the lessons and the subsequent “archiving” functions involved in making the documented lessons accessible and assimilable for users.
Investigation functions are needed to develop LTBL data and document all lessons
Archiving functions are needed to make LL easily accessible and assimilable for users

The Developer's Part of the System



Lessons developers also have a lot to do. This developer part of the learning system model reflects several strategic choices by us, based on previously reported work. For this model, we chose to

define the "lesson learned" as a description of what happened during the accident process,

document the lessons as coupled behavior sets in order to do that

Identification of lessons learning system components was derived in part from personal experiences with functions and actions required to bring about successful behavioral changes in people, objects and energies through accident investigations.
This accident killed a firefighter training officer. Our finding out what happened, and subsequent tasks, eventually led to major behavioral changes in the US fire services' responses.

Oh, there were some observations during unsuccessful efforts, too

What criteria should we use

to design a system that

efficiently, economically, quickly, and reliably

produces changed behaviors?

We identified 13 system attributes that present processes either lack or do not satisfy adequately.

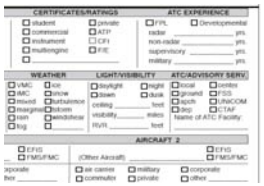
Most are new or newly defined.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Dynamic process compatibility

ASRS Reporting Form defines static attributes



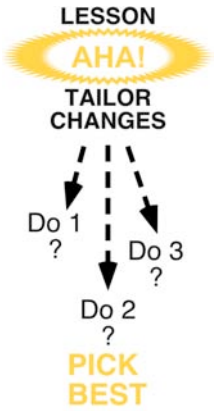
ASRS Format for description of occurrence dynamics

Relatively good data dictionaries and definitions of static data, such as that required by the US voluntary aviation safety reporting system now exist, but when describing the dynamics of an accident, we present blanks for writing unstructured narratives. No wonder it is so difficult to develop lessons learned from such data. Formal reports are not much better: the Commercial Aviation and Helicopter Safety Teams had to glean and recast data from formal reports to get the information they needed to propose safety improvement changes.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Dynamic process compatibility
- Multiple change options

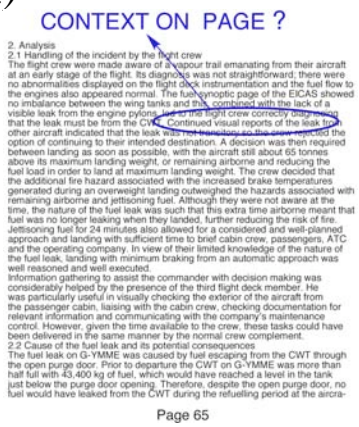


Recommendations by analysts rarely offer options for fixing "lessons" learned by investigations so they can be tailored to the specific activities of users. There are exceptions, particularly in some engineering lessons learned processes. Ambiguously worded recommendations requiring interpretations, it might be argued, offer tailoring opportunities, but that poses other problems.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Dynamic process compatibility
- Multiple change options
- Context identification



Documented lessons need to provide some form of context information for each lesson, to help users understand what happened, with the context in which it happened.
Formal accident reports often contain the context, somewhere in the narrative if one has the time, skill and tools to find it.
Context should not be a treasure hunt.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Dynamic process compatibility
- Multiple change options
- Context identification
- Expeditious accessibility

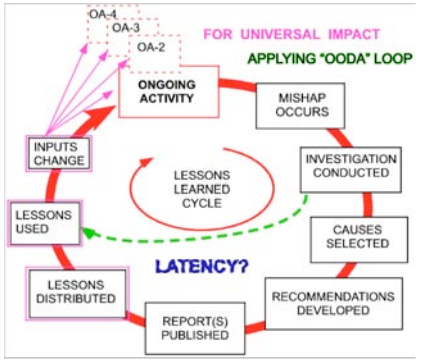
A major obstacle to use of lessons in present processes is potential users' difficulty in accessing the lessons.
Locating and accessing lessons is a challenge due to lesson strategy, data architecture, media, taxonomies, and other choices.
New options are available. A good metric would be how long it takes users to access a lesson.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Dynamic process compatibility
- Multiple change options
- Context identification
- Expeditious accessibility
- Minimal LL latency

680 → 41 → ?



Lessons learned "latency" is the delay between the time an accident generates the raw lessons-to-be-learned data and the time the lesson becomes available to potential users. One sample of 20 recent reports from a major investigation organization had a 680 day average latency period. A recent descriptive preliminary incident report by another organization had a 41 day latency period.

Learning systems could benefit from application of Boyd's OODA loop concepts. Bypassing analysts' functions by changing investigation and reporting of lessons could dramatically reduce latency periods.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Maximized signal-to-noise ratio

Low S/N

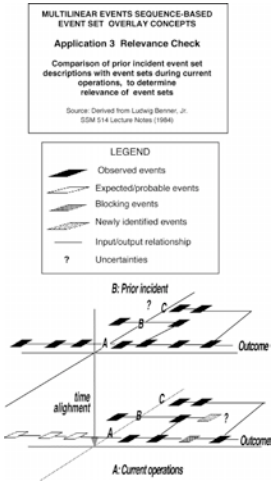
High S/N

A frequent user complaint is the quantity of data that must be searched to find the morsel - or "signal" - of interest to a potential user.
It is difficult to "get the picture" from “noisy” data. This is mostly due to use of unstructured narrative form and vocabularies of lessons.
Learning system design must address this signal-to-nose maximization challenge.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Maximized signal-to-noise ratio
- Expeditious relevance determination



Determining relevance of an accessed lesson is a subjective decision by a user. Users need to be able to "overlay" the lesson data onto their activities.
The longer this decision takes, the greater the disincentive for the user to use the system.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Maximized signal-to-noise ratio
- Expeditious relevance determination
- Maximized assimilability

Assimilation is the absorption or integration and use of lessons for one's benefit - the ultimate goal of the system. The spotty record of assimilation and achievement of new behaviors, for contemporary processes, raises the question: what is the best way to document lessons to ensure maximized assimilability? New choices are needed.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

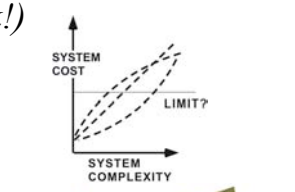
- Maximized signal-to-noise ratio
- Expeditious relevance determination
- Maximized assimilability
- System scalability

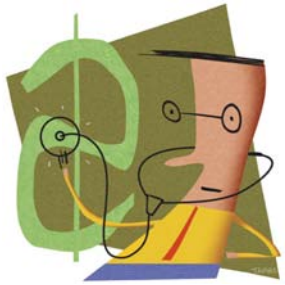
As system content grows, that growth should not sacrifice quality. Scalability needs to be designed into a lessons learning system so its growth does not discourage users from using it.
Retrieval problems with taxonomies, key words and categories suggest an alternative approach is needed.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Maximized signal-to-noise ratio
- Expeditious relevance determination
- Maximized assimilability
- System scalability
- Price sensitivity





Lessons learning systems cost money. Resources devoted to lessons learning systems are not without limits. The price sensitivity of such systems must be a consideration in system design, which means maximum efficiency of the lesson development, dissemination and use functions is an attribute to achieve in their design.

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Controlled socialization

What will Lessons convey?

Socialization is a subtle system attribute to consider.
Socialization of lessons, or how lessons fare in the social milieu after they are "published," poses at least two kinds of challenges - creating a climate to encourage the behavior changes, and avoiding a climate for obstructing changes.
Restricted sharing of the lessons, for example, can obstruct changes. The vocabulary used to document lessons can inflame or encourage reactions to their documentation and dissemination

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Controlled socialization
- System Performance metrics

System Attribute Metrics

Change Effects Metrics

Metrics for needed for how well a specific lessons learning system satisfies the attributes just described, to determine the success of system changes.
Another set of metrics is needed to determine if a changed behavior produced the expected improvement in performance, such as reduced risk, or cost, or improved efficiency, outputs, or other metric? In other words, was the lesson learned successfully?

Lessons Learning System attributes from Users' perspective:

(Or what present processes lack!)

- Controlled socialization
- LL Performance metrics
- Timely repository updating

Repositories must be kept trustworthy, by purging lessons learned that didn't work or were misdefined or otherwise unsuccessful, so users can sleep well after they use the repositories.

Lessons Learning System attributes from Developers' perspective:

(Or what present processes lack!)

Investigation components

- Purpose includes LL
- Input-output framework
- Focus on behavior data
- Specifications for building blocks
- Machine support for data sets
- Objective quality assurance

We found these attributes of the investigation components of a lessons learning system that are needed to optimize lessons learning system performance. Again, each can be linked to a specific impediment previously reported. The intent of each is described in the paper.

Lessons Learning System attributes from Developers' perspective:

(Or what present processes lack!)

Lesson documentation components

- Tools for behavior sets
- Behavioral output specs
- Machine processing support
- Internet repository capabilities
- Rapid repository access
- Objective quality assurance
- Repository updating capability

The lessons documentation components constitute a web of integrated system functions. Here are the needed attributes of the lesson documentation components we found. The intent of each is shown in the paper. Strategy choices affect these attributes. To isolate these attributes, we chose to show lessons as behavior data sets in this system.

Other observations

Specially appointed investigation entities address LL explicitly; established investigation entities do not.

Lessons learned system designs reflect past strategy choices that had inadvertent adverse effects on present processes and practices

Specially appointed entities like Buncefield and Challenger investigators speak directly to LL; Most established investigation agencies do not. LL strategy choices adversely affect LL process underperformance.
We thought it worthwhile to highlight both for you.

I should mention that some broad LL Programs like the US DoE SELLS program do list LL, but make no mention of listing lessons learned in investigation reports.

We need to redesign the system.

- Lessons learned processes work poorly due to inherent design flaws.
- An alternative lessons learning system with new attributes is needed to meet users' needs successfully.
- Redesign of lessons-to-be-learned source data and lessons documentation is an essential first step to lessons learning system optimization.

Again, a summary of results of our continuing study

Contemporary LL systems work, but poorly due primarily to inherent system design flaws

a better system is needed to meet users' needs successfully.

redesign of investigation inputs and lessons documentation is essential first step

To help get initiatives started:

Open Source (LGPL) library:

http://code.google.com/p/meslib/

Contribute your ideas too!

We believe very strongly in our findings, and the potential for evolutionary development of improved systems. So strongly that we are making publicly available an Open Source Library of software we developed, to launch the first steps toward needed changes. The Software Library includes a royalty-free license for use by anyone who wants to redesign their investigation data inputs and lessons documentation to support lessons learning system improvements.

Complete OS X sample app in Objective C, early development library in platform independent C++. Some sample PHP for online stuff too.

(Library GPL is the license)
(work in progress)
(C++, Objective C)

Thank you for Listening!

email: luben@starlinesw.com
email: billcarey@mac.com

Code at: http://code.google.com/p/meslib/

Slides at: www.starlinesw.com/ESReDA36.ppt

Thank you for listening.

Any questions?