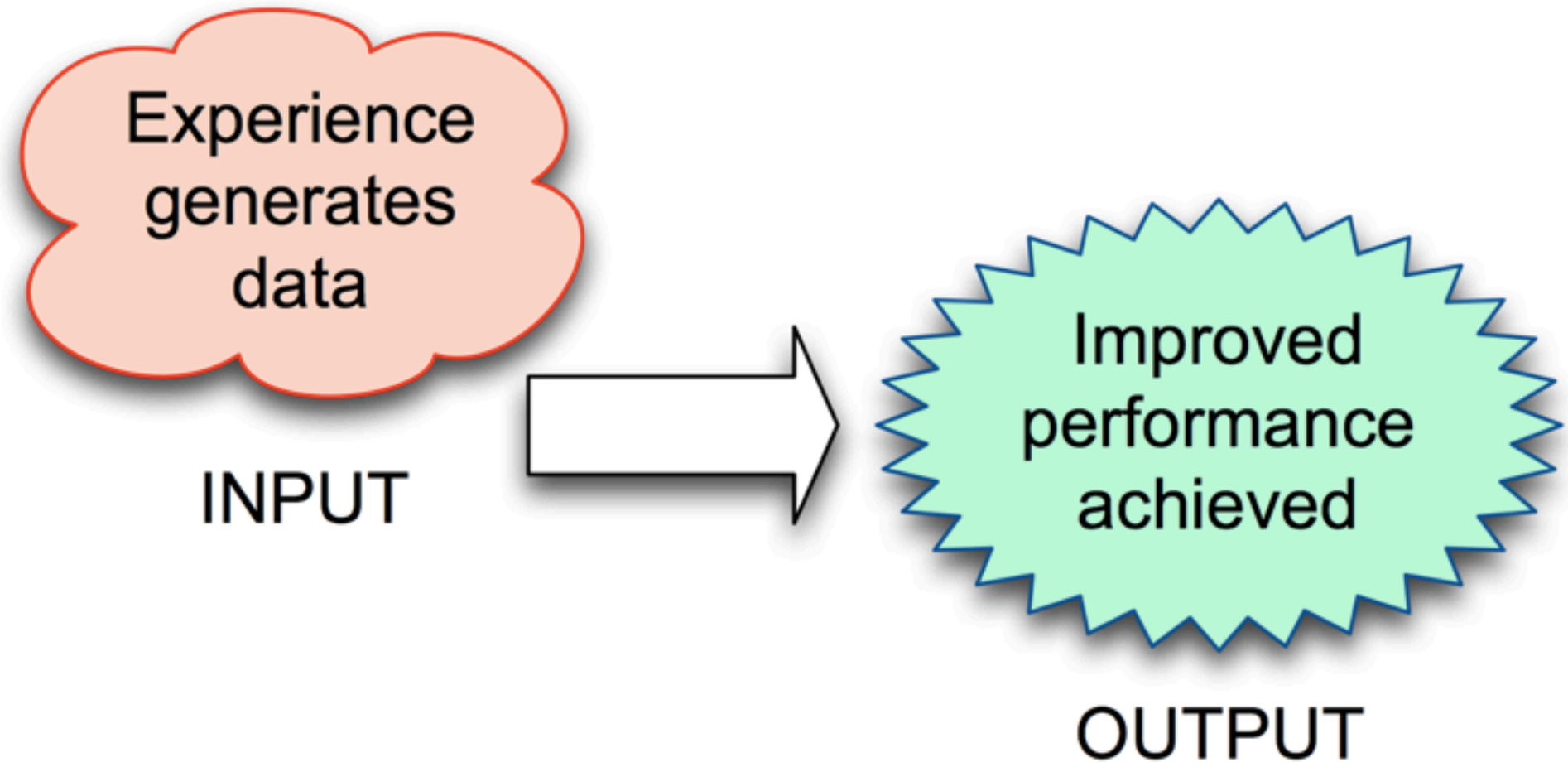
16th Annual Conference - Baltimore, MD USA

TRANSFORMING EXPERIENCE DATA INTO PERFORMANCE IMPROVEMENT

BY LUDWIG BENNER JR
VP Starline Software Ltd
Oakton VA 22124

Our Topic:



Produce Improvements Most Efficiently

2

This is our topic – transforming data generated by experiences into inputs that produce improved performance most efficiently.

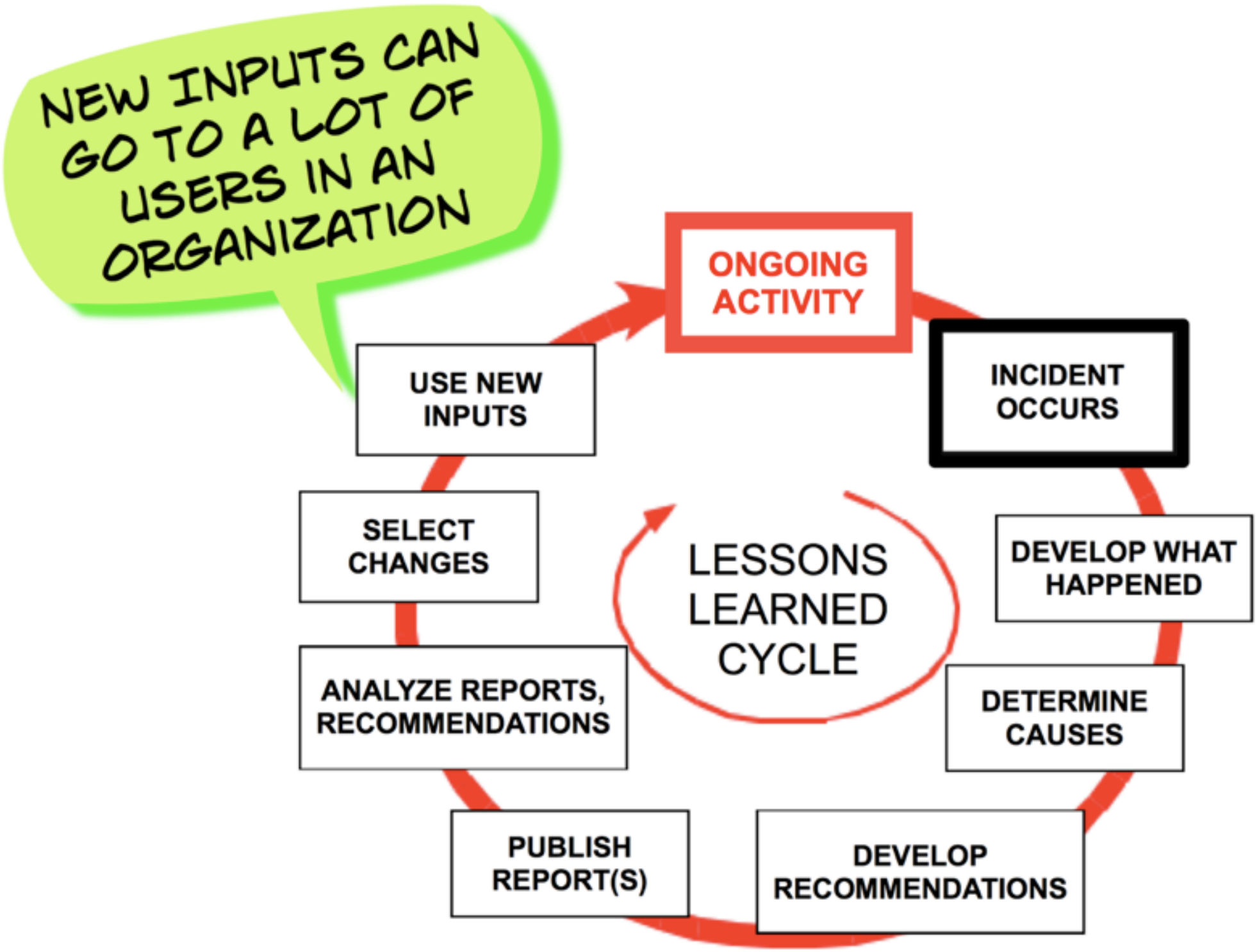
Current practices after incidents...

- Cause-finding investigation models
- Determine what happened
- Select Causes and Factors
- Make Recommendations
- Close Recommendations

Recommendations = fixes for lessons learned.
Lessons to be learned = ?

Let's start with how things are done now. Generally we start with cause-based models, determine what happened, pick out the causes or cause factors, develop recommendations and eventually do something to close out the recommendations. Often analysis steps are inserted between the 3rd and 4th steps, and sometimes users have to dig out the information, but this outlines the main steps.
Experience can produce lessons to be learned by others.
My question to you: if I am looking for what I can improve, where can I find a list of the lessons to be learned from the experience? A nice, concise, easy-to-acces, easy to use list that I can put to immediate use in my operations?

Incident Lessons Learned Cycle



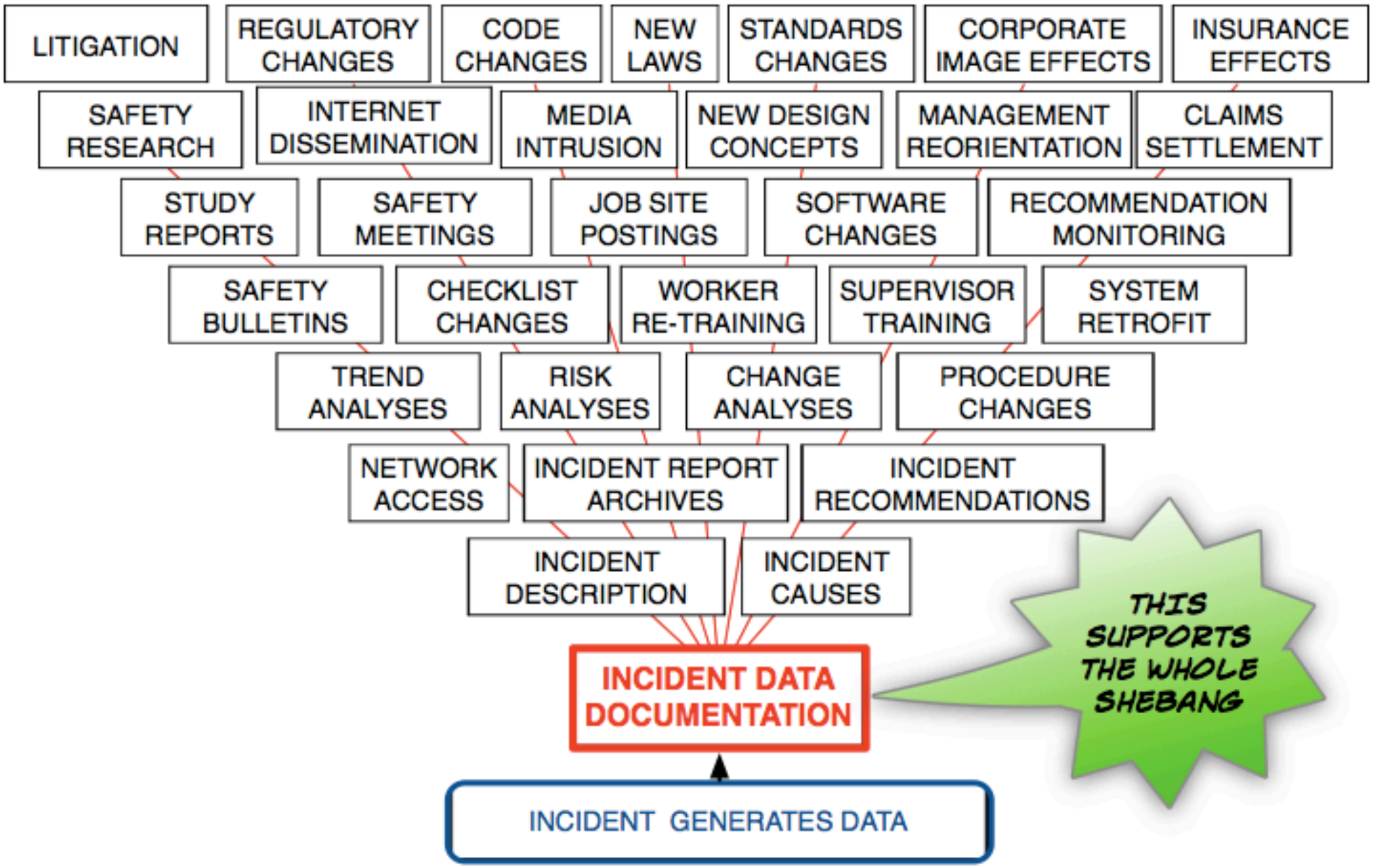
Now, if we look at an ongoing activity, the lessons learned system is continuous when the organization is a “learning organization” -one that joins adaptive learning with “generative” learning to create its future.
In a learning organization, an incident lessons learned cycle is a continuous “loop” where experience changes inputs to the ongoing activity-- after it becomes available.
We adopted the term **latency** to measure the time between the incident and the improved performance that resulted – and it’s usually quite long. **REMEMBER THIS METRIC**. We want to minimize latency.
-This graphic illustrates the general nature of present incident lessons learned cycles.

Who uses data from experiences?

- System operators
- Supervisors
- Managers
- Labor Reps
- Engineer & Designers
- Trainers
- Maintenance staff
- Procedures writers
- Safety staff
- Regulators
- PR staff
- Media
- Litigators
- Claimants
- Insurers
- Investigators
- Analysts
- Programmers
- ERP

Data to support improvement can be used in many places in an organization, or external to the organization. Here are some of the typical users who can put such data to work.

Experience Data Dependency Pyramid



6

BUT, it's crucial to recognize that all users depend on data generated by an incident, **documented so they can put it to use.** This illustration tries to illustrate the dependency of subsequent uses on the incident data documentation.
For maximum efficiency, that documentation should serve all users directly with minimal latency, without further categorization, abstraction, interpretation, reconfiguration or analogy.
Is that possible? I think so.

Experience Source Data Documentation

- Required to
  1. develop understanding of what happened – the process
  2. define lessons to be learned to improve performance
- Documentation affects all that follows
- Better get documentation right to avoid GIGO

First, we must acknowledge that this documentation of the experience-generated source data is pivotal to developing an understanding of what happened, and defining the lessons to be learned from that understanding.
—We must also recognize that this documentation affects all other improvement efforts that are expected to flow from the experience.
—So we'd better get this documentation right to head off GIGO in our improvement efforts

General impediments to developing lessons to be learned..

- Prevailing causation models
- Source data transformation
- Natural language barriers
  - Dynamic representation
  - Vocabulary
- Existing habits

8

It's not easy. We also have to recognize that we face some significant impediments to getting the source data documentation we would like to have…
The major impediments are...

- – practices based on models that encourage or -at a minimum– tolerate subjective, inconsistent, abstract, ambiguous and other detrimental documentation.
- – practices that accept widely variable capture, interpretation and documentation of experience–generated data.
- – practices that use natural language that is linear, not conducive to describing dynamic processes, and a vocabulary loaded with judgmental, pejorative, ambiguous or abstract words that impede verifiable descriptions and analyses of what happened.
- – comfort levels with present practices. (don't rock the boat)

Natural Language Barriers for Static Data:

PLEASE FILL IN APPROPRIATE SPACES AND CHECK ALL ITEMS WHICH APPLY TO THIS EVENT OR SITUATION.

REPORTER

- ☐ Captain
- ☐ First Officer
- ☐ pilot flying
- ☐ pilot not flying
- ☐ relief pilot
- ☐ check airman
- ☐ Single Pilot
- ☐ Instructor
- ☐ Trainee
- ☐ Dispatcher: _____ yrs
- ☐ Other: _____

FLYING TIME (in hours)

- Total Time _____ hrs
- Last 90 Days _____ hrs
- Time in Type _____ hrs

CERTIFICATES & RATINGS

- ☐ Student
- ☐ Sport/Rec
- ☐ Private
- ☐ Commercial
- ☐ ATP
- ☐ Flight Instructor
- ☐ Multiengine
- ☐ Instrument
- ☐ Flight Engineer
- ☐ Other: _____

ATC EXPERIENCE

- ☐ FPL ☐ Developmental
- radar _____ yrs
- non-radar _____ yrs
- supervisory _____ yrs
- military _____ yrs

AIRSPACE

- ☐ Class A ☐ Class E
- ☐ Class B ☐ Class G
- ☐ Class C ☐ Special Use
- ☐ Class D ☐ TFR

CONDITIONS/WEATHER ELEMENTS

- ☐ VMC
- ☐ IMC
- ☐ Mixed
- ☐ Marginal
- ☐ fog ☐ snow
- ☐ hail ☐ thunderstorm
- ☐ haze/smoke ☐ turbulence
- ☐ icing ☐ windshear
- ☐ rain ☐ other: _____

LIGHT/VISIBILITY

- ☐ dawn ☐ night
- ☐ daylight ☐ dusk
- Ceiling _____ feet
- Visibility _____ miles
- RVR _____ feet

ATC/ADVISORY SVC.

- ☐ Ramp ☐ Center
- ☐ Ground ☐ FSS
- ☐ Tower ☐ UNICOM
- ☐ TRACON ☐ CTAF
- ATC Facility Name: _____

	AIRCRAFT 1			AIRCRAFT 2		
Your Aircraft Type (Make/Model) (e.g. B737) NOT "N #", Flt #, etc.: _____			Operating FAR Part: _____	Other Aircraft: _____		Operating FAR Part: _____
Operator	☐ air carrier ☐ air taxi ☐ corporate	☐ fractional ☐ FBO ☐ government	☐ military ☐ personal ☐ other: _____	☐ air carrier ☐ air taxi ☐ corporate	☐ fractional ☐ FBO ☐ government	☐ military ☐ personal ☐ other: _____
Mission	☐ passenger ☐ personal	☐ cargo/freight ☐ training	☐ ferry ☐ other: _____	☐ passenger ☐ personal	☐ cargo/freight ☐ training	☐ ferry ☐ other: _____
Flight Plan	☐ VFR ☐ IFR	☐ SVFR ☐ DVFR	☐ none	☐ VFR ☐ IFR	☐ SVFR ☐ DVFR	☐ none
Flight Phase	☐ taxi ☐ parked ☐ takeoff ☐ initial climb	☐ climb ☐ cruise ☐ descent ☐ initial approach	☐ final approach ☐ missed/GAR ☐ landing ☐ other: _____	☐ taxi ☐ parked ☐ takeoff ☐ initial climb	☐ climb ☐ cruise ☐ descent ☐ initial approach	☐ final approach ☐ missed/GAR ☐ landing ☐ other: _____
Route in Use	☐ airway (ID): _____ ☐ direct ☐ SID (ID): _____	☐ STAR (ID): _____ ☐ oceanic ☐ vectors	☐ visual approach ☐ none ☐ other: _____	☐ airway (ID): _____ ☐ direct ☐ SID (ID): _____	☐ STAR (ID): _____ ☐ oceanic ☐ vectors	☐ visual approach ☐ none ☐ other: _____

If more than two aircraft were involved, please describe the additional aircraft in the "Describe Event/Situation" section.

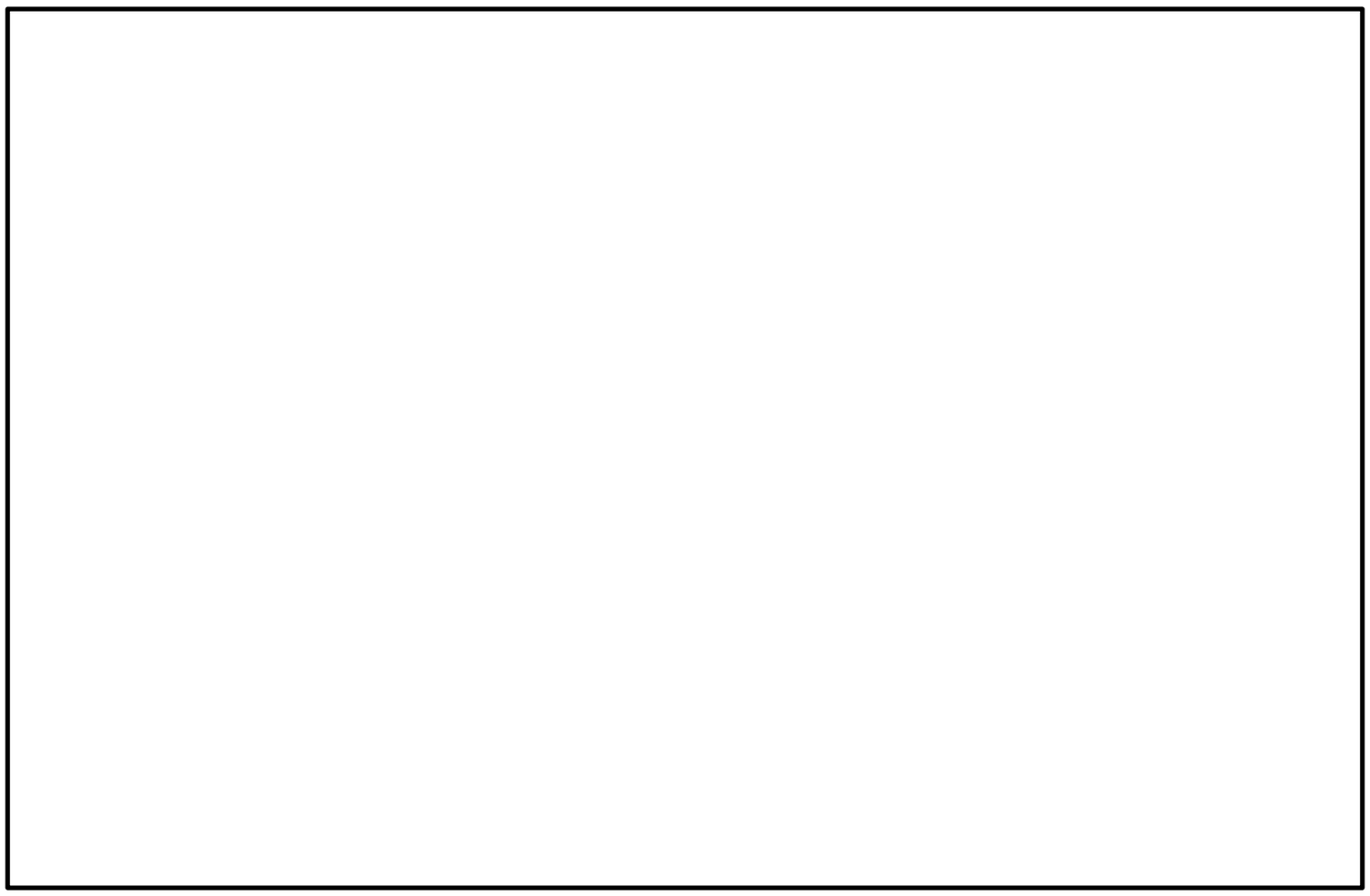
Source: ASRS form

9

Natural Language Barriers for static data have long been recognized — attempts to overcome them include
– forms with labeled blanks, - dictionaries with definitions of terms, – glossaries with what terms are intended to mean, – software traps, – check lists,
But what have we done to cope with dynamic data demands – descriptions of what happened?

Let's look at the Aviation Safety Reporting System as an example. You can see static data's pretty well defined, although not totally without some ambiguity.

Natural Language Barriers for Dynamic Process Data:



Source: ASRS form

but not so for dynamic data – ASRS Data Specifications =“Keeping in mind the topics shown below, discuss those which you feel are relevant and anything else you think is important. Include what you believe really caused the problem, and what can be done to prevent a recurrence, or correct the situation. (USE ADDITIONAL PAPER IF NEEDED)”

- How the problem arose - How it was discovered - Perceptions, judgments, decisions - Actions or inactions - Contributing factors - Corrective actions
- Factors affecting the quality of human performance

so we rely on habits and intuition…

Source data transformation challenges:

- Find/observe all "**TRACKS**" left by dynamic actions or people/object/energy behaviors during an incident
- Find "**PROGRAMMER**" inputs that shaped what people/ objects/energies did during an incident
- Transform tracks and programmer "**observations**" into structured data inputs or "**building blocks**" (BBs) to reconstruct what happened.
- Structure data to enable definition of the **lessons to apply** from the incident.

Source data transformation into documented inputs for analyses poses significant challenges –
-actions leave tracks during incident process – on people and things – you need to find all you can
-programmers influence what people, objects and energies do during normal ops and incidents –
-investigators have to transform what their senses can pick up, to document the data into building blocks for subsequent uses.
- Those BBs must enable definition of lessons to be learned, so they are critical task.

To overcome impediments,

1. Standardize **input data** structure

2. Standardize **data analysis** structure

3. Standardize **output** structure

All must be integrated!

Here are the steps I have found are needed to accomplish what we've been trying to get done – providing users with all available, readily assimilable experience data, quickly and efficiently
– First. standardize the structure of the input data documentation, to enable integration into a description of what happened and subsequent uses. It has been done in other fields - standardized distance, temp measurement, etc, but not this field
– Next, standardize the analysis structure – lots of precedents – but not conducive to producing readily assimilable outputs for end users
– Then, standardize the outputs – we now have risk matrixes, sort of, ICAO reports, various logic trees and symbols, blue print symbols, dictionaries or glossaries, etc but not readily assimilable ALL MUST BE INTEGRATED After looking at how others have resolved different parts of this problem, we've adapted ideas from work flow studies, cybernetics, learning organization and economics studies and – heavily – from music

Musical Score Model

A COMPARISON BETWEEN A MUSICAL SCORE AND MULTILINEAR EVENTS SEQUENCES CHARTING METHODS FOR ACCIDENT INVESTIGATORS

Actors:

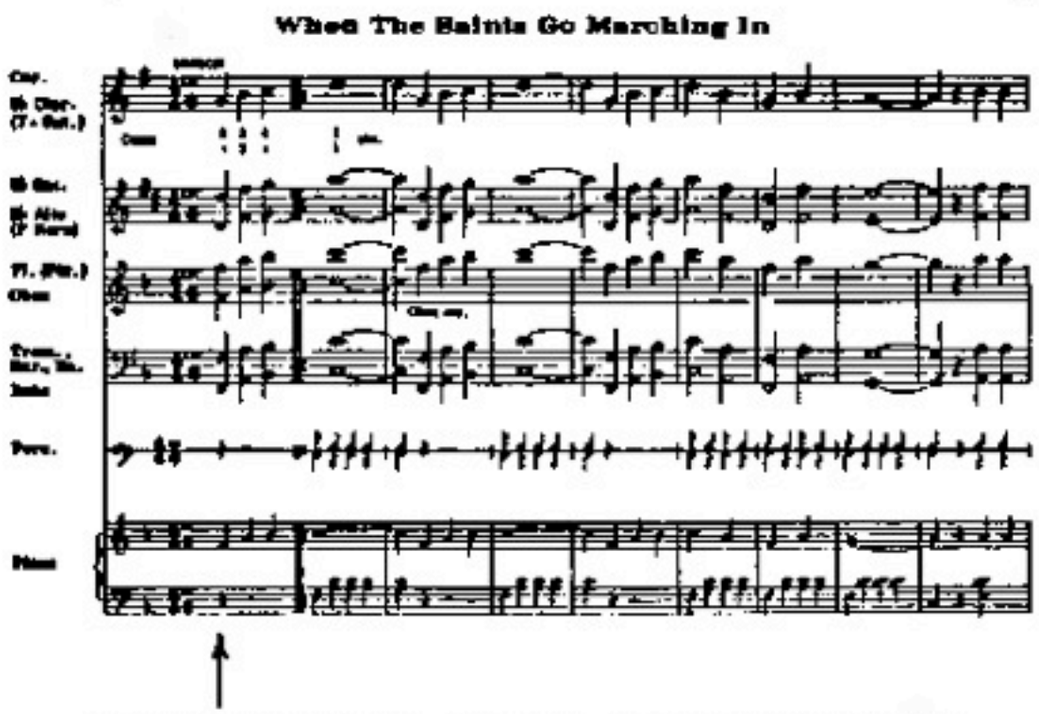
Implicit /Explicit

players

1,2,3	cornet b clarinet t sax
4,5,6	e sax e alto Fr. horn
7,8	flute oboe
9,10,11,	tromb bar.,Bn. bass
12,13	drums
14	piano

Time reference scale:

t_o t_{end}



Vertical alignments describe dynamic relationships among changes of state and steady states (at successive time units) produced by each events set

Horizontal lines and entries describe the time/locationally-ordered sequences of acts by each actor (events) defining their temporal and spatial sequential logic

By tracking the actions of individual actors from appropriate "records" one can reconstruct the entire sequence of events constituting the "song" and reproduce the "score." The score can then be reexamined and analyzed to serve the needs of the analyst.

By convention, events flow from the left of the page toward the right as time passes.

Notes:*

1. *Clefs indicate the line space and pitch, ie, the arena where the events take place.*

2. *Unison describes a policy for the activity*

4. *"Oboe,etc" (line 3-4) gives a warning sign to reduce the probability of confusion.*

*No pun intended!

4. *"count + 2 + 1 etc" (lines 1,2) leads to coordination of timing of actions by each actor.*

5. *define "location" for each next change of state(by scale + timing) and the duration of each steady state by the symbol used. "rest" symbols do the same.*

6. *defines the rhythm, or the basis for the coordination of the events during the conduct of the activity.*

7. *The "count" and "oboe" can be viewed as safeguards (or countermeasures) instituted by the composer or arranger to reduce the likelihood of mishaps.*

For example, this is the model used in the music world to structure and document dynamic actions by players or singers into a song
a notation system consisting of staffs, clefs, notes, bars, accents and symbols to describe a a complete song.
- standardized building blocks with time signatures – called notes
- standardized organization of the building blocks on staff lines, coordinated in their temporal sequence, – called melodies and harmonies
- a standardized output documenting the complete process required to reproduce the desired musical results, or compositions called songs

It accomplishes a lot – the documentation of the composer's envisioned process in a form that enables faithful and repeated reproduction of the composer's original intent – or if you want to modify it, prediction of the results

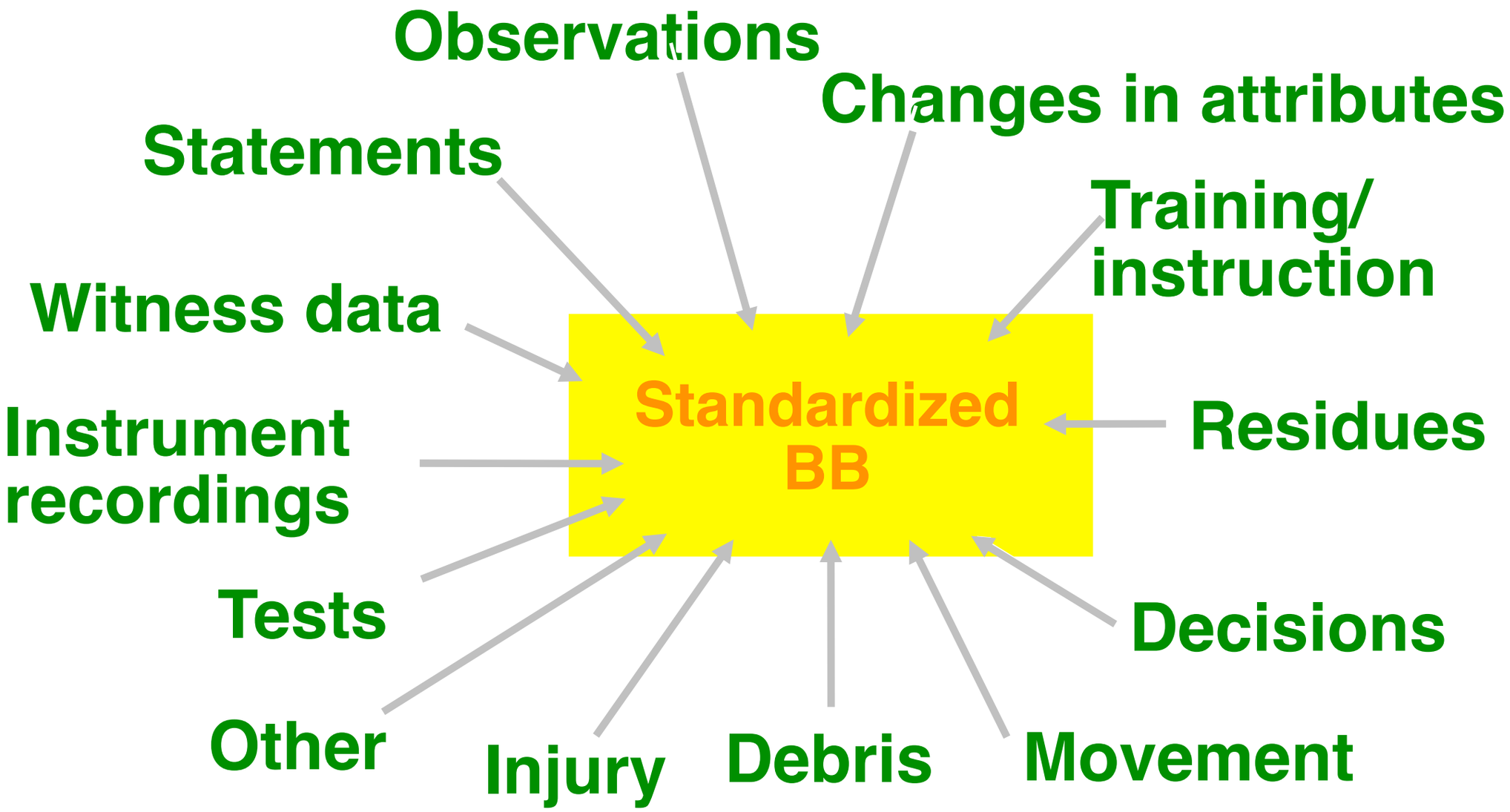
1. Standardize input data structure

Experience-generated source data must be **transformed** into standardized BBs that...

1. Are true **LOGIC STATEMENTS** with subject and predicate
2. Use **UNAMBIGUOUS VOCABULARY**
3. Facilitate **INPUT DATA ANALYSES**
4. Enable **INTERACTION LINKAGES**
5. Permit **VALIDATION** of descriptions
6. Support **DOWNSTREAM USES** of data

Let's look at the building blocks that we need to create from the experience-generated data. For our purposes, that data must be transformed in BBs nto BBs with these attributes…. (This too could use much more time, but here's the essence:)
Logic statements can be verified as true or false from the experience data observed. To describe what happened, grammar and syntax for each BB must be consistent and unambiguous – should be able to make a mental movie of what happened
Should be easy to use, enable showing of interactions, and permit validation of the descriptions to ensure quality
Ideally, end users should be able to use BBs in their documented form, without declassification or taxonomic guesswork to determine relevance and utility

TRANSFORM EXPERIENCE SOURCE DATA INTO STANDARDIZED BUILDING BLOCKS:



15

Transforming original source data into BBs unambiguously like musical notes is challenging.

This shows you the variety of experience data that needs to be standardized into building blocks for reconstructing what happened. Each poses its own challenges. For example specifying inputs from associated tests – and getting results that can be transformed in the needed structure is an interesting exercise.

BB Vocabulary

AVOID Poison Words in BBs

AND	HE	IT	WAS	...LY
OR	SHE	WE	WERE	FAILED TO
THERE	THEY	THEM	DID NOT	INADEQUATE

- plural actor names (firefighters)
- passive voice (was struck)
- opinion verbs (violated)
- compound actor names (crowd)
- conditionals (if, may)

AMBIGUOUS OR ABSTRACT WORDS
PREVENT INPUT DATA ORGANIZATION AND
LOGIC TESTING BEFORE SUBSEQUENT USES

16

Vocabulary is another challenge. Also, vocabulary used for documenting BBs must be as unambiguous as possible, or else the words will “poison” the analysis and other subsequent tasks. Each actor must have a unique name or identifier. Actions must also be as unambiguous as possible, which means they are at the lowest level of abstraction possible. Tough to do, but doable with practice.

Standardized BB Structure that meets needed criteria

ACTOR		1. Person or thing that did something
ACTION		2. What Actor did
OBJECT/DESCRIPTOR		3. Additional data defining action
LOCATION		4. Location where action occurred
REMARKS		5. For remarks about BB or investigatoin
SOURCE		6. Source of data for BB creation
BEGIN DATE/TIME		7. Date and time action began/status
DURATION		8. Duration of action
END DATE/TIME		9. Date and time action ended/status
LINKS FROM/TO	##	10. Link data with other BBs
N/S STATUS		11. N/S validation test status

This isn't pie in the sky. Here is a structure that has been used successfully to meet the BB criteria.
Each element has its essential role for later organizing, linking, testing and output preparation tasks.
10 and 11 are needed to capture the coupling of input/output related actions, and the status of the logic testing of the analysis for gaps in the description of what happened.

XML Building Block Example

```
<?xml version="1.0"?>
<mesblock unique_id=""> (9)
                    <actor></actor> (1)
                    <action></action> (2)
                    <object></object> (3)
                    <location></location> (4)
                    <start_time type=""> (6)**
                            <year></year>
                            <month></month>
                            <day></day>
                            <hour></hour>
                            <minute></minute>
                            <second></second>
                            <millisecond></millisecond>
                    </start_time>
                    <end_time type=""> (7)
                            <year></year>
                            <month></month>
                            <day></day>
                            <hour></hour>
                            <minute></minute>
                            <second></second>
                            <millisecond></millisecond>
                    </end_time>
                    <source></source> (5)
                    <remarks></remarks> (8)
                    <nstest> </nstest>***
                     <link></link>****
</mesblock>
```


This is an example of that BB structure using the XML format, to support machine interoperability.
This structure makes it easy to set up data entry, parsing, concatenation and analyses of the experience source data. Which leads me to the next structure needed.

2. Standardize Analysis Structure

To determine what happened, structure should...

- Enable development of a **process description** of the incident
- Accommodate and organize all BBs as acquired
- Enable BB temporal and spatial sequencing and manipulation
- Provide for linking of interacting BBs to show input-output relationships and context
- Expose gaps or missing BBs in incident understanding
- Allow necessary and sufficient logic testing
- Facilitate risk raising/risk reducing lessons to be learned definitions

19

What do you do with your new BBs?
Here's my list of what you should expect your analysis structure to do for you, as you try to gain understanding of what happened during incidents you experience

kind of keep the musical score ideas in mind with this...

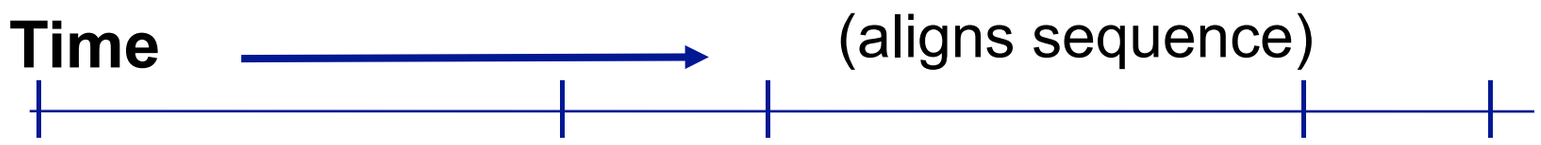
Data Analysis Structures have been proliferating…

Some examples:

- CAS
- CAUSE TREE
- Change Analysis
- ECFC
- Fishbone
- Five Whys
- FMEA
- FRAM
- HFACS
- MES/STEP
- PROACT®
- Reality Chart™
- SnapCharT®
- STAMP
- TOP-SET
- Tripod Beta
- Why-Because

Lots of choices – but remember criteria...

BB Analysis Structure that works well...



lists all involved actor names

Actor A

Actor B

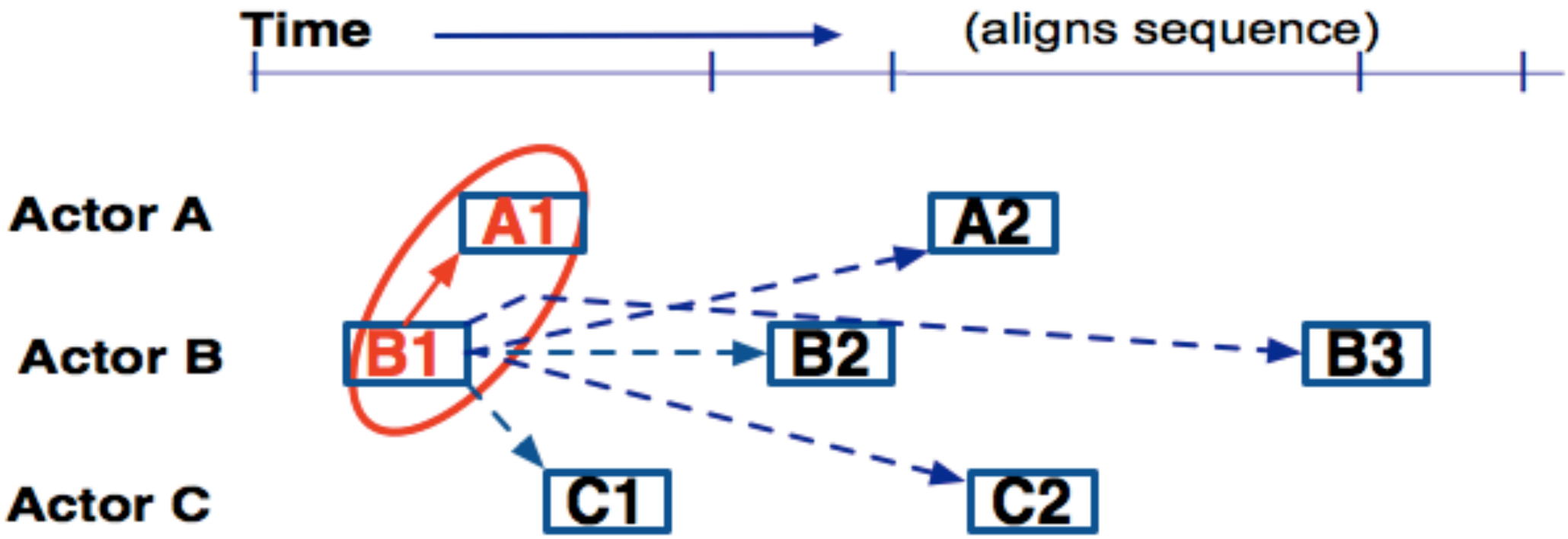
Actor C

Actor n...

BBs can be positioned on worksheet as acquired, aligned by row and start time

Organizing the BBs is the first analysis task. Based on musical score model, this is the structure I prefer because it works well with the BBs just described – fast, disciplined organization of input data, expandability, expedited linking and testing as understanding grows…
you end up with a rigorous process flow chart description
and… it Provides useful results from episodic incidents!!

Linking interactions…



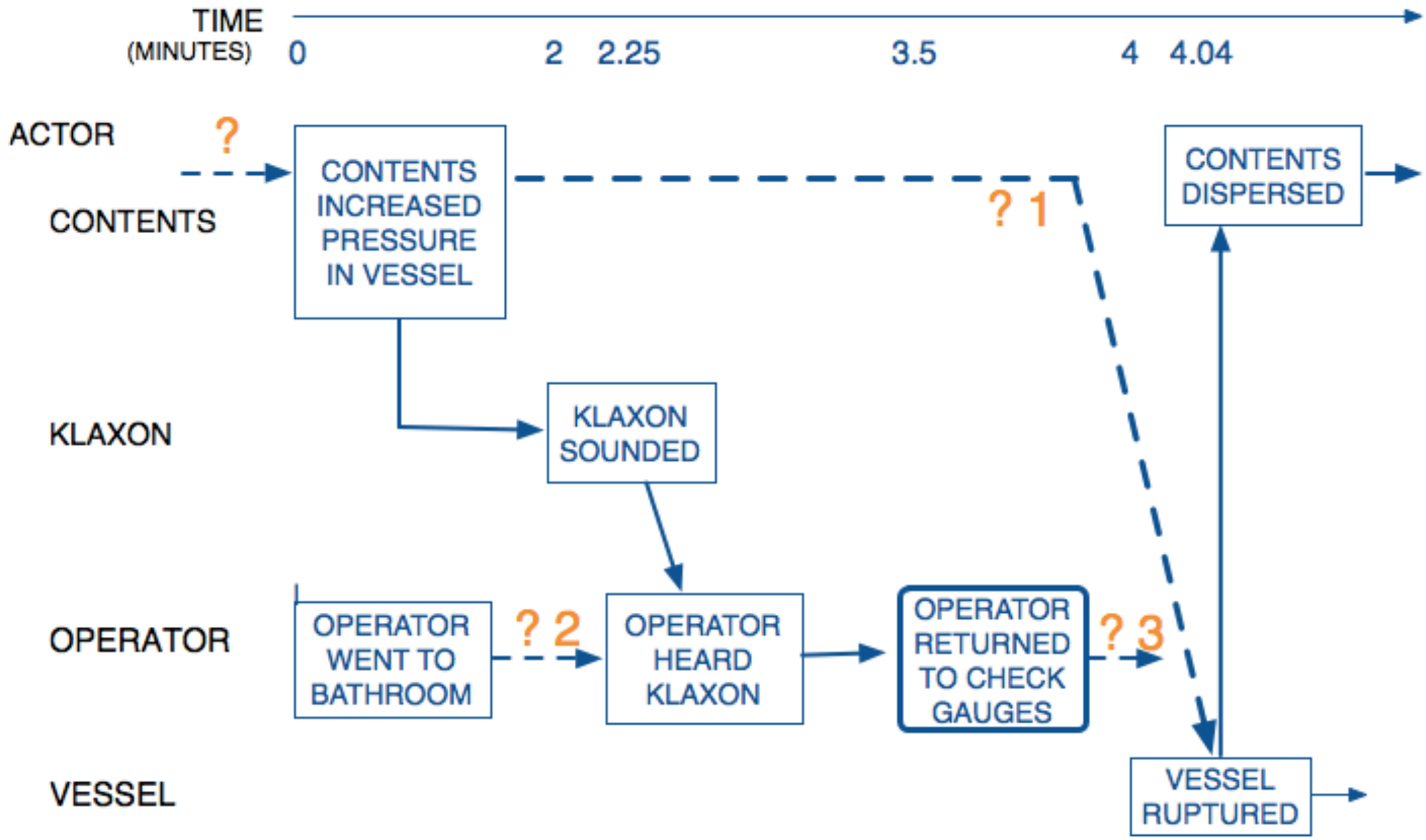
1. Select two BBs on worksheet.
2. Was B1 an input needed to make A1 happen?
3. If so, link B1 to A1 to show B1 as input to A1.
4. Repeat step 2 for A2, B1, B2, B3, C1, and other later BBs.

22

Linking of BBs is second analysis task to show interactions during dynamic processes, and define what happened. This is a manual task, requiring logic skills and understanding of how a system worked, or is supposed to work or could work. To my knowledge no algorithm exists for computers to do this linking, although computer software does exist to support and document this manual task.

Let me point out that this task provides an opportunity to replace causal models with an input-output model during the analysis. That's important because it makes cause determination irrelevant, giving organizations the opportunity to avoid all the problems involved with cause determinations and uses of causes in the dependency pyramid. If you get nothing else out of this presentation, remember input-output offers a way you can describe what happened without using cause or factors. Try it, to see how long you can get by without using the term cause or caused.

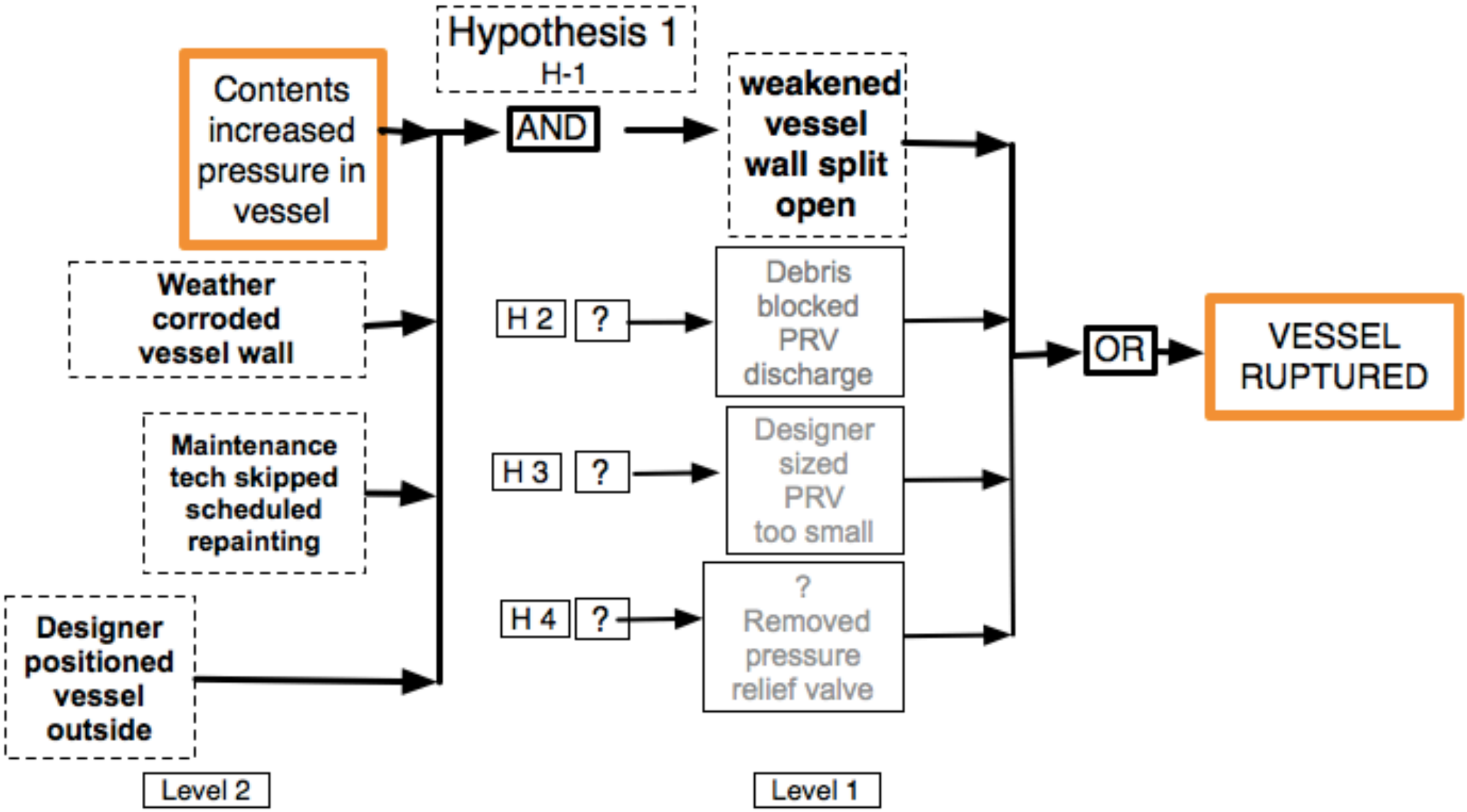
Standardized Worksheet Structure Example
(abbreviated BBs displayed)



This is a section of a worksheet being developed after a pressure vessel rupture. It already tells you a lot. Each ? indicates a gap in the flow of the actions, and a need for more information. For example, the tentative link to the contents tells you that you would need to pursue why the contents increased the internal pressure. The tentative link or dotted line from the contents to the vessel rupture ?1 indicates another gap in your understanding of what happened: why would the vessel rupture? The tentative link at ?2 indicates a need to find out where operator was when he or she heard the klaxon. The ?3 indicates a need to find out what operator did before the vessel ruptured. The solid links indicate sufficient information to tentatively conclude the system performed as expected.

What to do about the gaps?

Closing gaps...



24

That's where "bounded logic trees" are used to develop hypotheses about what happened in the gap. For each hypothesis, supporting data is then sought, to determine which hypothesis most likely occurred. The surviving hypothesis is then entered onto the worksheet where further interactions, if any, can be identified and linked.
This is when I think hypotheses should be introduced while trying to understand what happened

3. Standardize lessons to be learned structure

To define lessons to be learned…

- Look at each pair of linked BBs
- Determine if those behavior pairs constitute part of a risk raiser or risk stopper for the incident
- Each risk raiser or stopper becomes part of a lesson to be learned for subsequent performance improvements
- Create “behavior sets” including coupled inputs, behavior and outputs as lessons to be learned structure

25

Since I don’t think recommendations - or preordained fixes – satisfy end user needs well, what should be the structure for lessons to be learned by end users? Here’s my answer.

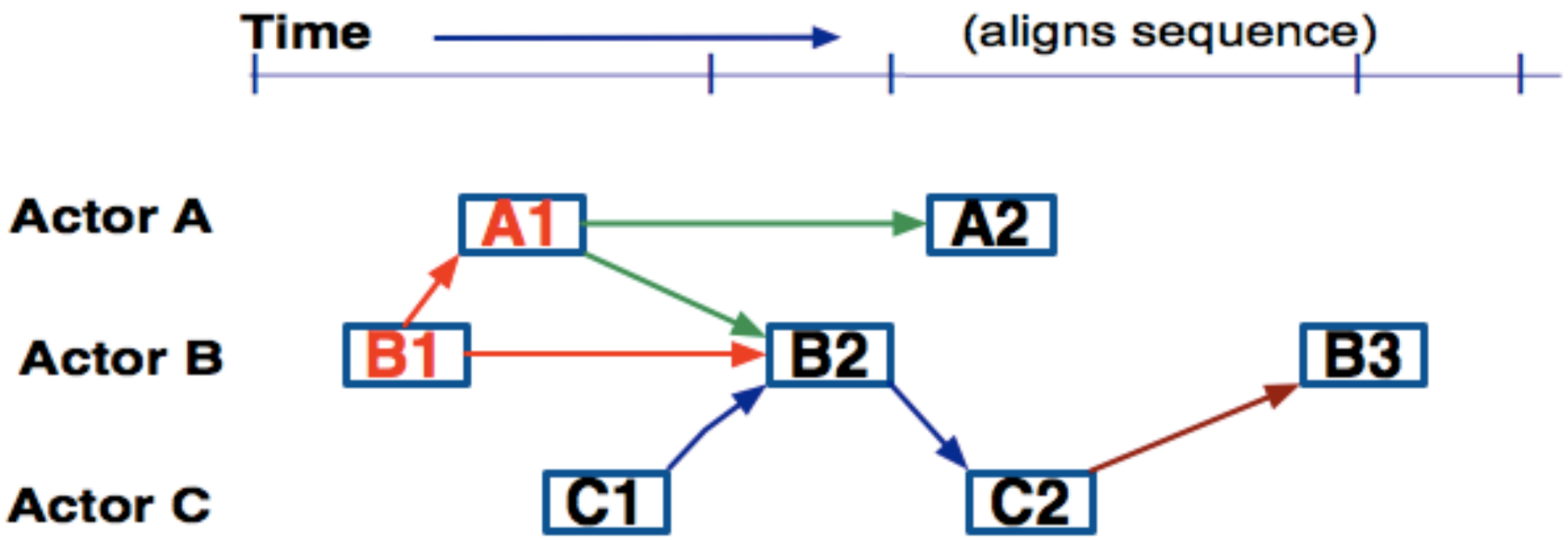
– With all the links in place, we can develop “behavior sets” to describe interactions that were needed to produce the incident outcome. Each behavior set can be a risk raiser when unwanted outcomes occurred, or a risk-stopper when an unwanted outcome was aborted. Either kind of set provides a lesson to be learned.

Interaction inputs and outputs form behavior sets

26

A behavior set consists of an input to a behavior, and subsequent behaviors influenced by that behavior. The necessary and sufficient inputs identify why the behavior occurred and the outputs define the effects of that behavior on subsequent actions, all of which were required to produce the final outcome.

Forming behavior sets...



Acotor n...

The three BEHAVIOR SETS shown by links on this worksheet are:

1. **B1 was a necessary input to A1 which in turn was a necessary input into B2 and A2**
   or: B1 > A1 > A2,B2
2. **B1,A1,C1 > B2 > C2**
3. **B2 > C2 > B3**

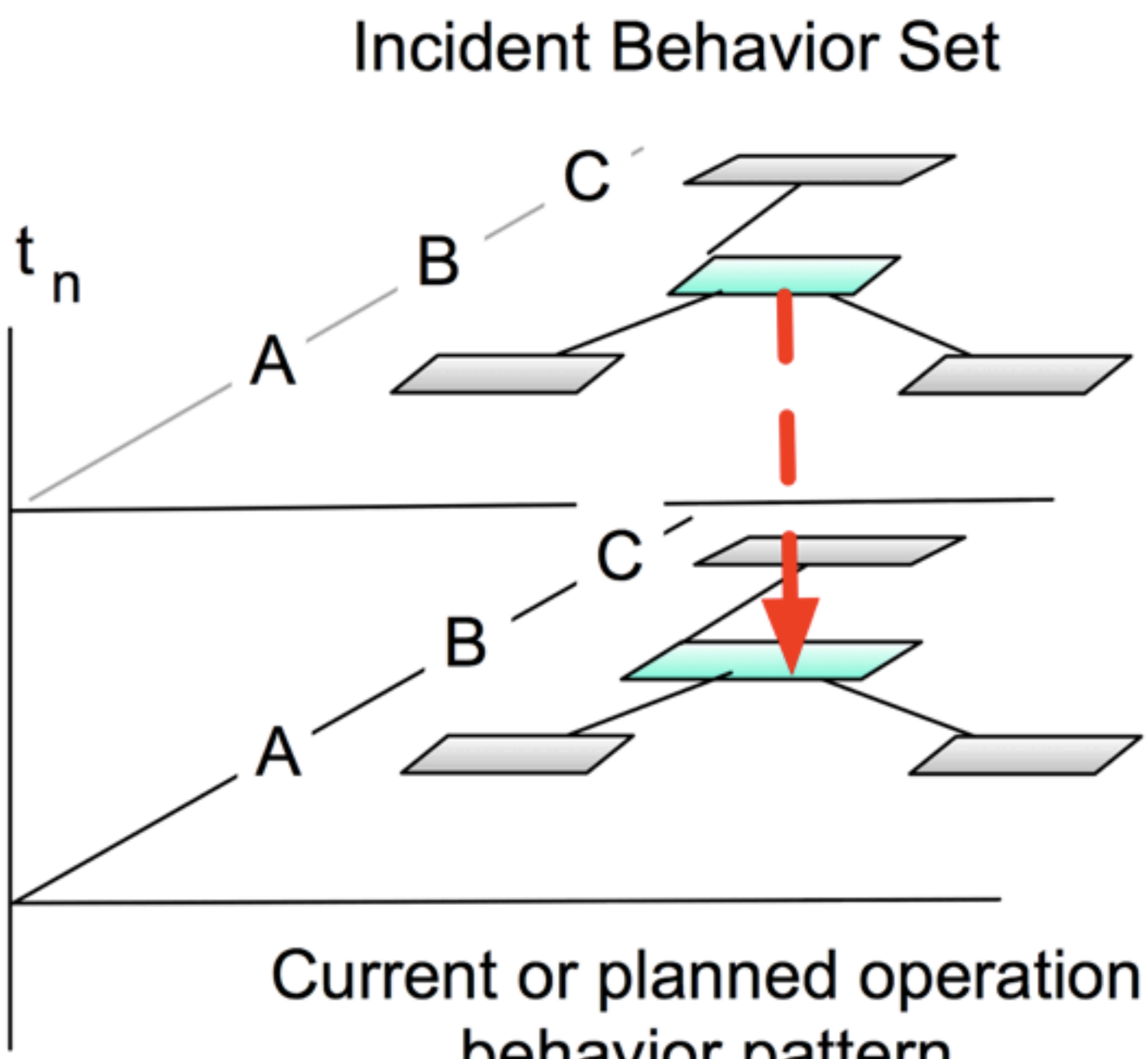
B3 is last BB, which is an outcome. C2>B3 is behavior pair,

27

Behavior sets can be created directly from the BB and links on the structured worksheets, in the manner shown here.

- If the BB and analysis structures presented here are used for software operations, behavior sets can be created by machine and presented in tabular, graphic or narrative format, depending on users' needs.
- These behavior sets describe the behavior patterns to be avoided or emulated in future operations to reduce future risks, or simply to improve future performance.
- In this form, they provide users with lessons to be learned that can be OVERLAID onTO their current or planned operations, thus improving their relevance determination, assimilability and effectiveness monitoring.

Behavior Set Overlay

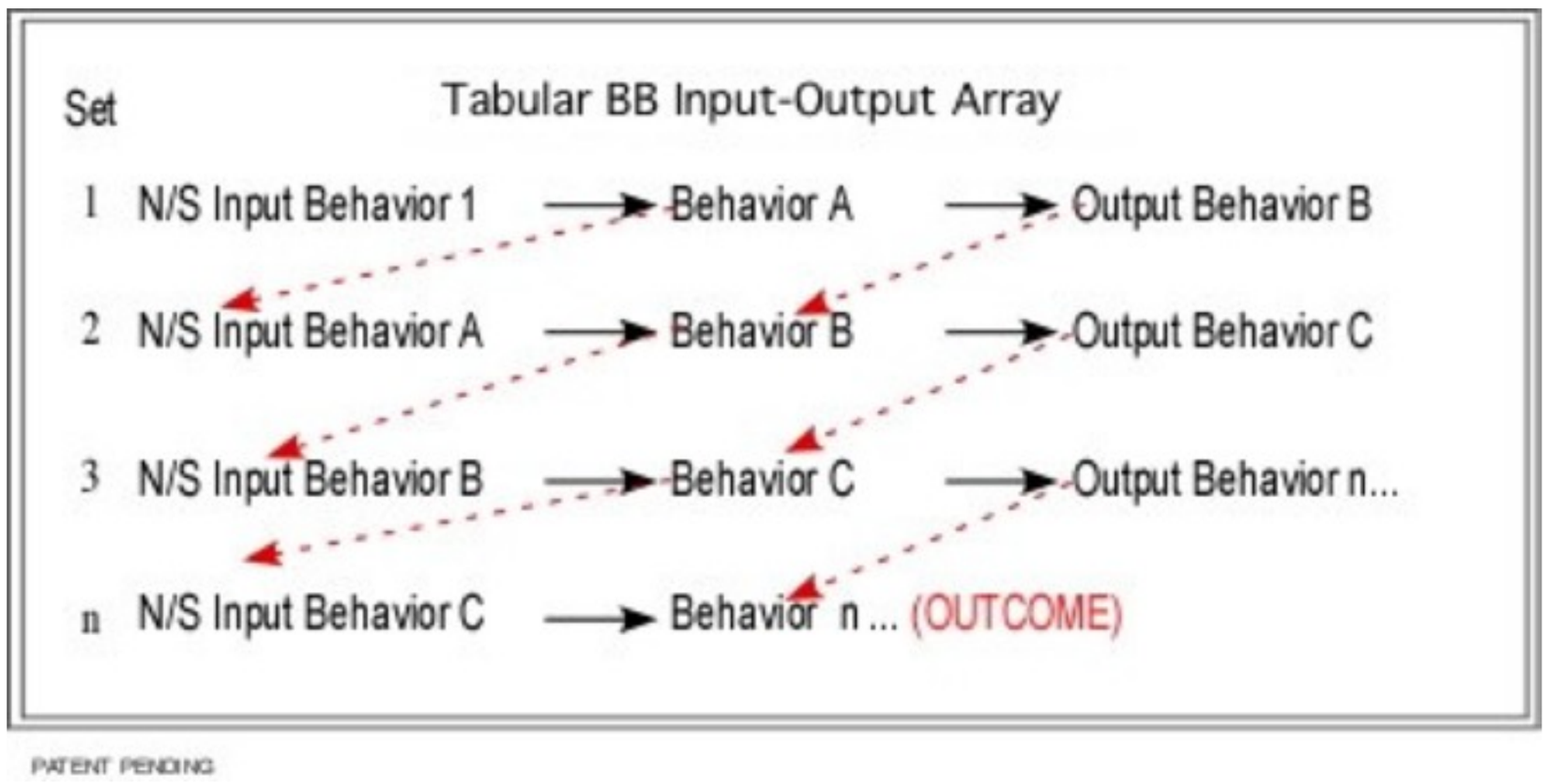


28

The is an example of how a user would overlay a behavior set onto a current or planned operation. If the incident pattern exists in the overlaid operation work flow, a change in the pattern, timing, magnitude or other attribute of overlaid behavior set is indicated.

BBs recast into Input/Output Array to show Behavior Sets in Lessons To Be Learned Array

Lessons learned display, showing behaviors *in context*
(actions, coupled inputs and outputs)



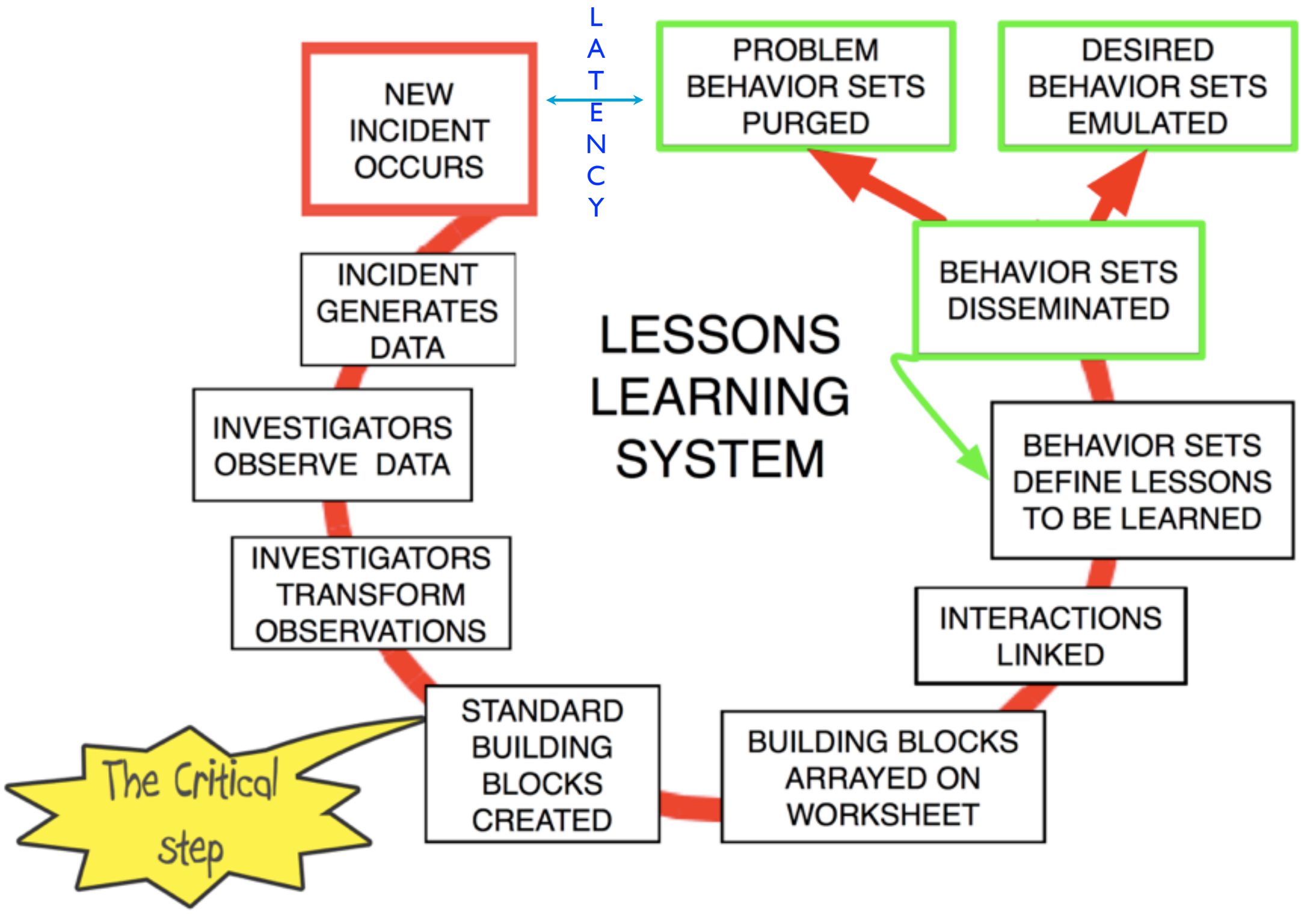
Machine-generated from behavioral building blocks
created on a personal PC or on the Web

29

•The XML data permitted us to develop machine-generated **BEHAVIOR SETS** and arrays from behavior building blocks. Each set shows an action or behavior ***in context***. The input actions and output actions necessary to produce the process outcome, as determined by the investigators, are shown. The sets can concatenated by machine to build data bases for other analyses.

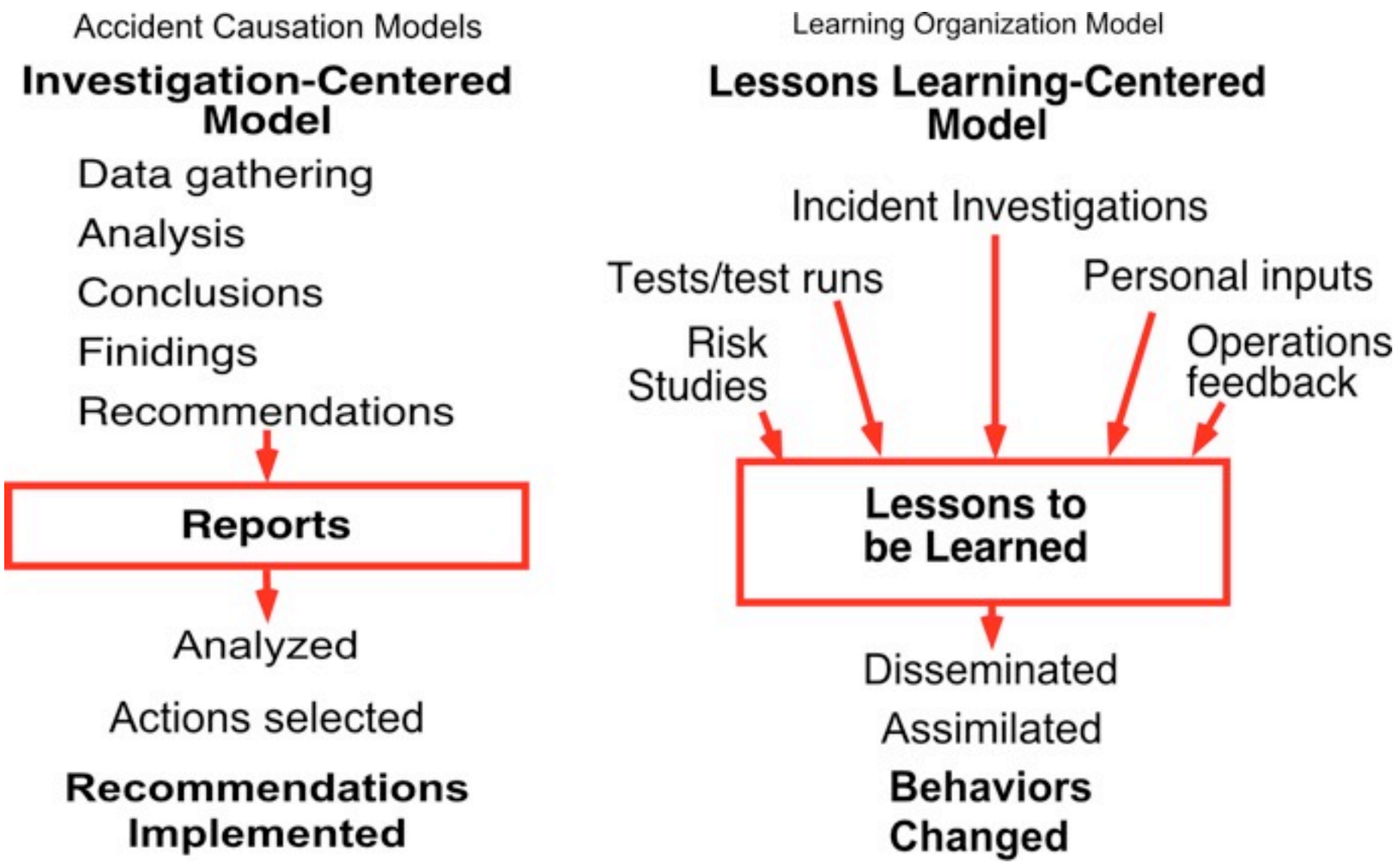
•This opened the door to providing users concrete action or behavior sets which are patterns they can **OVERLAY** on their own work processes, showing them where and how their processes can be corrupted – and where they can be improved. By posting aggregated machine-generated set data on a web site, authorized users can access and return for refresher reviews or new queries as often as they like to reinforce **habit formation**

Learning Lessons



Here's how such a system of structurally standardized building blocks, structurally standardized analyses and structurally standardized outputs works. The critical step, is the source data documentation into the structurally standardized BBs.

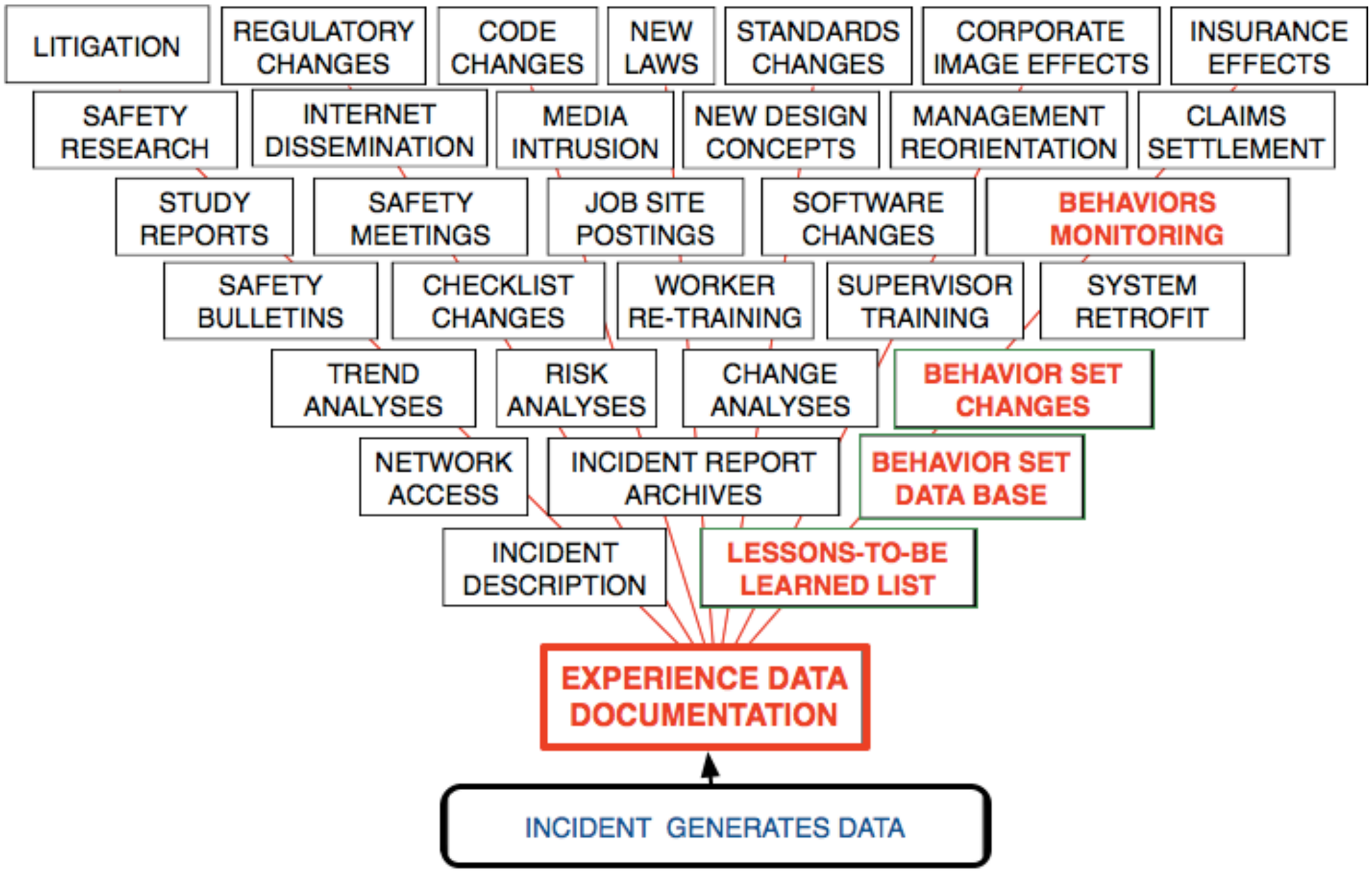
Comparison of learning models



31

the paper shows the differences in the present causation model and the I/O model this way

LTBL Experience Data Dependency Pyramid



32

as illustrated by the modified Lessons to be learned Experience Data Dependency Pyramid, these documented incident-generated source data can flow more directly to the end users who modify behaviors in their activities.

Behavior Sets: why bother?

Lessons to be learned development...

- Enable dynamic behavior descriptions
- Provide rapid trustworthy LTBL documentation
- Produce readily assimilable lessons to learn
- Minimize subjectivity with
  - ✓ transparent logic
  - ✓ unambiguous descriptions
  - ✓ elimination of “red herrings”
  - ✓ averted “cause” differences

33

To wrap this up,
Why bother to explore this approach, using behavior sets?
Here are some of the reasons you might want to bother looking at your lessons to be learned development practices and how they might be improved in your organization with behavior sets...

Behavior Sets: why bother?

Lessons to be learned dissemination...

- timely standardized machine interoperable data storage inputs
- semantic web-friendly dissemination and access capability
- minimal verbiage to encourage retrieval
- behavioral context for lessons to be learned
- readily structured LTBL outputs for end users' needs

34

Here are some of the reasons lessons to be learned dissemination would be faster, better cheaper with behavior sets...

Can provide unambiguous, uncontroversial LTBL and ready accessibility for trainers, designers, supervisors, operators and any other authorized user to develop changed behaviors that improve performance in their areas of responsibility.

Behavior Sets: why bother?

Lessons to be learned implementation...

- Minimal condensed change sets to work with
- Readily overlaid onto end users' operations
  - ✓ to assess relevance
  - ✓ to find remedial options
  - ✓ to assess feasibility and consequences of options
  - ✓ to prepare implementation plan
  - ✓ to monitor change over time
- Offer metric for confirming improvement actions
- Enhanced safety communication among managers, supervisors, operators, trainers, designers and safety advocates

35

Here are some of the reasons LTBL implementation could be better with behavior sets...

They provide unambiguous inputs as overlays for trainers, designers, supervisors, operators to assess and adopt or adapt

Last one assumes you use I/O vocabulary

To implement structures...

Incrementally...

- Add standard data input module to your present investigation software
- Adopt matrix-and-links analysis tools to develop behavior sets
- Publish lessons-to-be-learned (LTBL) as separate report entity
- Modify software to produce tailored LTBL search and retrieval results
- Use behavior sets for risk analyses and improvements

36

Transition need not be that painful if implemented **INCREMENTALLY** over time
Before doing anything, flow chart your present process from the generation of the source data until a successfully changed behavior has been habituated to achieve lasting improved performance. Try using the structures described here.
If you want to actually start making changes based on what you've learned here, you can do it incrementally…
- First step is to address the experience data transformation task by adding a standard structured experience data input module meeting the criteria described to your present software.
- When better BBs become available, introduce the matrix and links analysis module.
- When LTBL behavior sets become available, publish them the behavior sets
- When they are published modify search and retrieval tools
- After they become accessible, start overlaying them on current or planned operations., and set up feedback channels.

Summary of Highlights

- ☑ Goal: shortest path from Experience to Improvements
- ☑ Experience generates your input source data
- ☑ Acknowledge source data documentation dependency
- ☑ Transform source data into standardized BB structure
- ☑ Structure analysis to produce behavior sets
- ☑ Overlay behavior sets on operations to define changes
- ☑ Use latency as system performance metric
- ☑ Implement incrementally

Here are the main points that I think you could use to help you produce better performance improvement processes.

I’m finished!

Give you any ideas?

Raise any questions?

luben@starlinesw.com

For additional information visit Starline Software’s web site at

ww.starlinesw.com or see Technical Notes at

www.investigationcatalyst.com/TN/technotes.html

38

There’s so much more I could say about extracting source data, transforming it, vocabulary, setting up data bases, access and retrieval strategies, relevance testing, system performance evaluation, role in risk analyses and decision making, impact on litigation, safety communication and more - but now you have the basics, so I’ll stop here.
Any questions or comments?