Risk-based Continuous Systems of Systems Data Integrity Auditing of IoT

Dr. Byron Mattingly
Thursday: 1:45p-2:30p
12 October 2017
The views expressed in this presentation are my own and do not reflect the views of my employer or other organizations with which I am or have been affiliated.

All cited Trademarks are the property of their respective owners.
Abstract

As the cost of micro-electromechanical sensors (including gyroscopes, accelerometers and pressure sensors) has dropped by more than 80% in the past years, the number of connectable things equipped with these sensors continues to undergo explosive growth, thus boosting the expansion of information due to mobile, cloud, social and other forces. In 2020, it is estimated that the "internet of things" (things that will input data themselves without being entirely dependent on people for information) will reach 30b and that total amount of data in the digital universe has increased from 130 exabytes in 2005, to 8.6 zettabytes (=8.6 x 1000 exabytes) in 2015 to more than 40 zettabytes in 2020.

Increasingly, then, "things" considered as complex adaptive systems--systems of "agents" in a network acting in parallel--are entering a rich but ever-expanding ecology at the "edge of chaos", regions in a phase transition zone far from equilibrium between stability and chaos. Such systems offer new challenges for the quality professional seeking to control and mitigate risk over and above multi-agent “systems of systems” because of properties like hierarchical self-organization, emergent but non-linear behavior, and self-similarity.

This session examines how refactoring (complex adaptive) systems of (complex adaptive) systems of the Internet of Things (IoT) partitioned by risk can be used to continuously audit data integrity flows similar to block-chaining techniques used to establish non-reputable financial audit trails. In particular, specific suggestions for risk controls applied to the chain of custody of data audit trails can be used to develop pipe-like message passing guidelines for an I/O semantics that is protocol-independent thereby minimizing unintended interactions and side-effects that may lead to, for instance, patient harm in the case of mobile medical devices.
Expected Takeaways

✓ Reimagining continuously auditing systems of systems in terms of the Boyd OODA Paradigm.

✓ Better understanding of integrating risk controls to follow critical chain of custody data audit trails in complex software ecologies such as mobility medical apps in the clinical and non-clinical environments that deploy IoT architectures.

✓ Better understanding of how continuously auditing and monitoring residual risks can significantly improve the data integrity of a complex adaptive system.
Harvey unloaded 33 trillion gallons of water in the U.S.

Saturday evening update, Sept. 2, 2017:
The overwhelming majority of Harvey’s rains are over, and we have a new calculation for the total volume of water it dispensed on U.S. soil: 33 trillion gallons. This number incorporates the rainfall not only in Texas and Louisiana, but also in Tennessee and Kentucky, which also experienced torrents.

Harvey unloaded 33 trillion gallons of water in the U.S.

33 x 10^{12} gals = 29.97 cubic miles

U.S. = 3.797 million square miles

Equivalent rainfall of \( \frac{1}{2} \) inch across U.S.


If one teaspoon = one byte of data . . .

757,500 gallons of water per second
768 “bytes” (teaspoons) to a gallon

582 MB/s of data flowing @ 1 teaspoon / byte

How big is 44 ZB ($= 44 \times 10^{21}$) bytes of data?

How big is $44 \text{ ZB} = 44 \times 10^{21}$ bytes of data?

1 Zettabyte is $10^{21}$ bytes = 1 Trillion GB
44 ZB = $5.73 \times 10^{19}$ gallons of water

“According to a recent report by market research firm IDC, the IoT currently comprises some 20 billion connected “things” — all of them collecting, sharing, and/or using data — and that number is expected to approach 30 billion by 2020.”

“There’s Plenty of Room at the Bottom”
—Richard Feynman (APS Meeting, Dec 29, 1959)

Gyroscope ~ $90\$ < 10x Price Drop in 5 years!

MEMS Market Grows as Prices Decline

Sources (8/21/2015):
Old Software Never Dies

//COBUCLG JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//HELOWRLD EXEC COBUCLG,PARM.COB='MAP,LIST,LET'
//COB.SYSIN DD *
001 IDENTIFICATION DIVISION.
002 PROGRAM-ID. 'HELLO'.
003 ENVIRONMENT DIVISION.
004 CONFIGURATION SECTION.
005 SOURCE-COMPUTER. IBM-360.
006 OBJECT-COMPUTER. IBM-360.
0065 SPECIAL-NAMES.
0066 CONSOLE IS CNSL.
007 DATA DIVISION.
008 WORKING-STORAGE SECTION.
009 77 HELLO-CONST PIC X(12) VALUE 'HELLO, WORLD'.
075 PROCEDURE DIVISION.
090 000-DISPLAY.
100 DISPLAY HELLO-CONST UPON CNSL.
110 STOP RUN.
//LKED.SYSLIB DD DSNAME=SYS1.COBLIB,DISP=SHR
// DD DSNAME=SYS1.LINKLIB,DISP=SHR
//GO.SYSPRINT DD SYSOUT=A
//


October 12 - 13, 2017 26th Annual ASQ Audit Division Conference: The Intercontinental Hotel, Addison, Texas
Code listing for the Apollo Guidance Computer (AGC) program

> 3M parts / 700k components / 20k contractors

Source: [http://authors.library.caltech.edu/5456/1/hrst.mit.edu/hrs/apollo/public/archive/1701.pdf](http://authors.library.caltech.edu/5456/1/hrst.mit.edu/hrs/apollo/public/archive/1701.pdf)
Boeing 777 = 3M parts from 500 suppliers

Source (8/25/2015):
Model Based Systems Engineering with Interface Control Documents (ICDs)

Risk = Severity x Probability(hazard)

4 Pillars of MBSE:
1. Structure
2. Behavior
3. Requirements
4. Parametrics

Start Here: http://www.omgsysml.org/

Example risk tools:
PHA, FTA, FMEA (use, process, design)

October 12 - 13, 2017
26th Annual ASQ Audit Division Conference: The Intercontinental Hotel, Addison, Texas
“As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it.”

—Meir Manny Lehman, 1980

Boyd OODA for Validation

From “The Essence of Winning and Losing,”
J. R. Boyd, January 1996
http://www.d-n-i.net
Feedback Loops Reduce Uncertainty and Manage Change

- OODA Loops
- Shewhart / Deming Cycles (PDCA)
- Six Sigma (DMAIC)
- Lean Manufacturing
- Lean Startup
- Lean UX
Agile Approach

Project Setup

Develop, Test & Feedback

Develop, Test & Feedback

Develop, Test & Feedback

... Release

26th Annual ASQ Audit Division Conference:
The Intercontinental Hotel, Addison, Texas

October 12 - 13, 2017
Verification Driven Agile Development

From this:

- Bug injection
- Bug detected
- Bug found
- Bug fixed

... to this:

- $T_{detect}$
- $T_{find}$
- $T_{fix}$

$T_d$, $T_{find}$, $T_{fix}$

Bug$_{injection}$
Bug$_{found}$
Bug$_{fixed}$
TC = Test Case
Goal: 90-95% TCs Automated
"The greatest complexities arise exactly at boundaries. . . . The lesson of boundaries is hard even for systems thinkers to get. There is no single, legitimate boundary to draw around a system."
—Donella H. Meadows, *Thinking in Systems*, pp. 95, 97

**Limits to Growth (1972)**

- Non-linear interactions between systems
- Exponential depletion:
  \[ y = \frac{1}{r} \ln ((r \times s) + 1) \]
  where:
  - \( y \) = years left;
  - \( r \) = continuous growth rate
  - \( s \) = static reserve
  - \( R \) = reserve;
  - \( C \) = (annual) consumption

What is re-factoring?

Code refactoring is a “disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior,” undertaken in order to improve some of the nonfunctional attributes of the software.

“Decisions made to defer necessary risk management and control throughout a software development lifecycle may result in regulatory debt.”
What is Regulatory Debt?

How to Pay Down the Debt:

• Technical Debt $\rightarrow$ Refactor

• Regulatory Debt $\rightarrow$ Risk Control (esp. Refactor into “System of Systems”)

“The computing scientist’s main challenge is not to get confused by the complexities of his own making.”

—E. W. Dijkstra
7.5.3 Control of documented information

7.5.3.1 Documented information required by the quality management system and by this International Standard shall be controlled to ensure:

a) it is available and suitable for use, where and when it is needed;

b) it is adequately protected (e.g. from loss of confidentiality, improper use, or loss of integrity).

7.5.3.2 For the control of documented information, the organization shall address the following activities, as applicable:

a) distribution, access, retrieval and use;

b) storage and preservation, including preservation of legibility;

c) control of changes (e.g. version control);

d) retention and disposition.
Electronic Records
Any digital information that a computer system can create, modify, maintain, archive, retrieved or distribute

Key Elements
• Who created the electronic record?
• When was the record created?
• Do audit trails track changes (and deletions) throughout the file retention time?
• Can only authorized personnel and entities use the system?
What is data integrity?

ALCOA

- **A**ttributable -- the originator of the data authorized / authenticated
- **L**egible -- non-reputable traceable audit trail can be reproduced / viewed (by data consumer)
- **C**ontemporaneous -- data time stamped when recorded
- **O**riginal record -- "true copy" preserves integrity (and meaning!) of record (including subsequent activity--cf. WHO)
- **A**ccurate -- data are correct, truthful, valid and reliable
Blockchains

“Blockchains are secure by design and are an example of a distributed computing system with high Byzantine fault tolerance.”

Source: https://en.wikipedia.org/wiki/Blockchain

Image Source: https://bitcoin.org/en/
Questions?

Thank You!