Inside Analysis

Story
Slides

Meetup - Eric
Welcome
Mission
Definition 1
Graphic 1
Definition 2
Quote 1
Exploration
Graphic 2
Quote 2
Exploration
Reality Check
Definition 3
Quote 3

Meetup - Geoff
Data Science for Normalized Data with a View into HealthCare.gov
About the Speaker
PSIKORS Institute
HealthCare.Gov: What We Know
Contractors See Weeks of Work on Health Site
The HealthCare.Gov Data Veracity Challenge
HealthCare.gov Can't Handle Appeals of Enrollment Errors
Simple Picture for Congress
Simple Picture for Congress Annotated
The Issue is Data Normalization
Data Normalization Matrix
Data Normalization
Using Data Science to Solve Normalization and Enable Higher Level Functions (Semantics)
Data Science Foundations Can Help

The Ψ–KORS™ System Model

Corporate NoSQL™

Aligning Data Values With Business Context

Data Value Semantic Errors = Inconsistent, Difficult to Merge, Report, Analyze

DataStar Normalization/Virtualization Platform

An Example Data Normalization Project

Clarifying Data Use and Meaning

Initial Results: First 30 Days

Outcome of First 30 Days

Deep Dive

Title Slide

Guests

Computers Don’t Lie

Logs Remember EVERYTHING

We Don’t Fully Understand Computers

Systems Management and Operational Intelligence

The Operational Intelligence Marketplace at InsideAnalysis.com

HealthCare.Gov Issues

What We Know

Lessons Learned From Others

Simple Picture Created for Congress

Govt Never Learns

A Few of Many Gotchas

How to Solve 1

PSI-KORS: Sense Making from People to Execution Technical Architecture

Questions

Thank You

Spotfire Dashboard

Research Notes

Transcript

Point 1: Context

Point 2: Look at the Computer Logs

Point 3: What We Know

Point 4: Serious Problems with Data Aggregation and Accuracy

Point 5: Lessons Learned from Others

Point 6: Simple Picture Created for Congress

Point 7: The Web Environment Was Not Originally Designed for High-performance Applications
Story

Redux: What Went Wrong with the Obamacare Web Site, and How Can It Be Fixed?

Inside Analysis debuted last October with an excellent discussion of "What Went Wrong with the Obamacare Web Site, and How Can It Be Fixed?". We want to re-visit that in our Federal Big Data Working Group Meetup by having Inside Analysis update us and to show the pilot work that Semantic Community and Be Informed did to show how it can be fixed.

The transcript has been parsed in to key points to summarize the discussion and set the stage for the Semantic Data Science and Metamodel work as follows:

- **Point 1: Context** - Still Good Point
- **Point 2: Look at the Computer Logs** - Still Good Point
- **Point 3: What We Know** - Still Good Point
- **Point 4: Serious Problems with Data Aggregation and Accuracy** - Is This True Now?


"Another sore point was the Medicare agency’s decision to use database software, from a company called MarkLogic, that managed the data differently from systems by companies like IBM, Microsoft and Oracle. CGI officials argued that it would slow work because it was too unfamiliar. Government officials disagreed, and its configuration remains a serious problem."

MarkLogic President told me that the ML Data Hub worked almost perfectly: [http://semanticommunity.info/MarkLogic_Government_Summit_2013#Story](http://semanticommunity.info/MarkLogic_Government_Summit_2013#Story)

So there were capacity (solved), data aggregation and accuracy (still true now?), metamodel (still non-existent or too simple), and STP problems (because of metamodel problems - see Point 13 below)
Response: Point 4 is the most true of all the points. You can see that from the new images of news articles I put into the slides. Indeed, this is the single biggest problem that backend technical people know will overwhelm HC.gov (see quotes in news articles) compared to the relatively simple capacity problem.

The web capacity issues were simply a reflection of the dreadful acquisition process and incompetent HHS program mgmt. Note I did not say contractor program mgmt. The much larger problem space is data merging.

The kick-them-in-the-head approach to ML is disingenuous, back-stabbing, and technically untrue. I am not a big fan of ML's technology as the best fit HC.gov, but it is certainly no worse than the complete mismatch of Oracle, IBM, SyBase, or other run-of-the-mill legacy DB techs for this purpose. They were simply made a scapegoat, and I guarantee with Govt knowledge and acquiescence.

I think it will be a mistake to concentrate on issues from a few months ago and ignore the latest ones. Note the report this morning about security concerns because they hired Russian programmers. Nice adherence to the FAR!!

The metamodel issue is not as important as the data normalization one.

That is, you, me, frank, sally, jose, mighty mouse, can all have different opinions (none wrong) about the need, value, LOE of defining a metamodel for web site as distinct from the information model defining the integrated record as distinct from the data model defining the data storage and system-system exchange, but in the end I can build a very high performance sophisticated web site without a metamodel and be quite happy. You can build one with a metamodel and be equally happy. Mighty mouse might do something different.

ML's contention that their hub worked well is OK with me. My point is that the architecture and very design of the hub itself, regardless of the DB tech, was inherently flawed.

- **Point 5: Lessons Learned from Others** - Still Good Point
- **Point 6: Simple Picture Created for Congress** - Still Good Point
- **Point 7: The Web Environment Was Not Originally Designed for High-performance Applications** - Still Good Point
- **Point 8: What Are the Alternatives to This?** - Still Good Point
- **Slide 9: Government Never Learns** - Still Good Point
- **Point 10: A Few of the Many Gotchas** - Still Good Point
- **Point 11: How to Solve Number One** - Still Good Point
- **Point 12: PSI-KORS Metamodel** - Still Good Point
- **Point 13: Interlocking Timeouts** - Still Good Point

My Comment: Be Informed calls this the Straight Through Processing (STP) problem which it handles in their Metamodel and Platform

Response: Point 13- Be Informed is entirely incorrect. That grossly simplifies how real servers do handshakes and the very large number of independent time-outs in the lineage of a call from a browser click (timeout1 started) through all of the proxy servers, load balancers, SSL devices, app servers, DB servers, DB triggers, ... and so one. Claiming there is...
such a thing as STP using a metamodel is laughable when you get to the hardware level. It can work in a greatly simplified arrangement, the kind of thing Microsoft tries to convince folks they provide from within Visual Studio, but this does not work in practice for even a trivial load balancing setup using Apache httpd and any app server, DB server. It is not even relevant to the architecture discussion when building something at the scale of HC.gov.

- **Point 14: Separating the Know From the Flow (Be Informed)** - See Point 13
- **Point 15: List of Government IT Failures** - Still Good Point, But Controversial
- **Point 16: Opportunity For an Open Source Project** - **Still Good Point** - Our Semantic Community - Be Informed Healthcare.gov demo is an "Open Source Project" in the sense that Be Free is free and the repository is available for further development
- **Point 17: Wrap Up** - Still Good Point

**My Comment**: So these points set the stage for the Data Science I did for Be Informed to do a Metamodel that works! And why I suggested using them because the message is simple and logical. I had hoped that the PSI-KORS Metamodel tool would be available for you to show it and/or try it myself.

**Response**: I am very happy to bring the PSIKORS model into the Semantic Community to test and align with other work. We can discuss this issue and plan a time to talk in detail amongst the players whenever you want to set it up.

**My Points Made During Discussions**: BeInformed works with legislators and policy-makers to see that their wording and intent can be implemented in the metamodel and application. This would have helped when Congress wrote the 1700 page Affordable care Act. Our pilot goals were two-fold: (1) to show how to build a better Web site and (2) to show how to build a better database with semantics and strong relationships from the start. The latter is well-known to be very difficult, if not impossible, with legacy databases, as Geoffrey Malafsky discussed. This pilot work is well-documented so it can be reused and reproduced by others.

In the broader context, there are two basic approaches to including semantics in big data applications:

- The first we showed was Semantic Medline/YarcData where the quality assured data used is the UMLS Ontology developed and maintained at the NLM by a team of subject matter experts for many years and the semantic NLP extraction of PubMed documents (22 Million so far) into RDF Triples (2.2 Billion so far). This data is now stored all in memory (512 TB available) in the new YarcData Graph Appliance and is searched and explored by scientists like Dr. Tom Rindflesh and others with some subject matter expertise to make discoveries. We are actively encouraging others to use this graph database.

- The second we showed was this Healthcare.gov Data Science pilot to help Be Informed knowledge modelers (Kees van Mansom in this pilot) develop metamodels that capture the processes and use quality assured input data needed for accurate registration and the calculation of benefits. This application provides an auditing capability to the user and builds a semantic strong relationship database that can be reused. It is very agile and can be modified and expanded readily because it uses what Gartner and Forester call Dynamic Case Management which is well beyond conventional Business Process Management (BPM).
Meetup - Eric

Welcome

**Presenter:**
Eric Kavanagh

Twitter: @Eric_Kavanagh  
InsideAnalysis.com

Mission

- Provide some perspective on the initial launch of Healthcare.gov
- Get back to First Principles as to how this launch failed, and what can be done going forward
- Solicit feedback from the group and perhaps write a follow-up article discussing relevant issues

Twitter: @Eric_Kavanagh  
InsideAnalysis.com
Definition

semantics

[si-man-tiks]
noun (used with a singular verb)
1. linguistics.
a. the study of meaning.

Graphic

Captain Obvious Strikes!
Computers Don't Lie!

What's the not-to-subtle implication here? Let's make it explicit:
Computers don't lie...
But people do!
Definition 2

**Definition**

**philosophy**

[fī-lōs′ō-fē′]
noun, plural philosophies.
1. the rational investigation of the truths and principles of being, knowledge, or conduct.

---

**Quote 1**

**Quote**

Immanuel Kant

Making plans is often the occupation of an opulent and boastful mind, which thus obtains the reputation of a creative genius, by demanding what it cannot itself supply; by censuring, what it cannot improve; and by proposing, what it knows not where to find.

---

**Exploration**

Exploration

Why the First Rollout of HealthCare.gov Crashed, an Architectural Assessment


Graphic 2

Graphic

From The Washington Post:

HealthCare.gov, built by 55 contractors, is one of the most complex pieces of software ever created for the federal government. It communicates in real time with at least 112 different computer systems across the country. In the first 10 days, it received 14.6 million unique visits, according to the Obama administration.

Quote 2

Quote

Immanuel Kant

To appeal to common sense, when insight and science fail, and no sooner -- this is one of the subtle discoveries of modern times, by means of which the most superficial rater can safely enter the lists with the most thorough thinker, and hold his own. But as long as a particle of insight remains, no one would think of having recourse to this subterfuge. For what is it but an appeal to the opinion of the multitude, of whose applause the philosopher is ashamed, while the popular charlatan glories and confides in it?
Obamacare Tax Credit Payment System Still Unbuilt, Official Testifies

By Devon Dwyer

Nov 19, 2013 5:49pm

Seven weeks after the launch of the new federal health insurance marketplace, "30 to 40 percent" of the online system still has not been built or tested, including the critical component that processes tax credit payments to insurers, a top Obama administration official said today.

"HealthCare.gov, the online application, verification, determination, plan compare, getting enrolled, generating an enrollment transaction, that's 100 percent there," said Henry Chao, the Centers for Medicare and Medicaid Services project manager in charge of Healthcare.gov, told a House panel.

"There's the back office systems, the accounting systems, the payment systems. They still need to be built, Chao said.

Reality Check

Right now, insurance companies are estimating the subsidies they'll receive. What does that mean? They really have little idea what to expect, which means another shock looms.

It also means there will be incremental payoffs to insurance companies. How transparent will those be? What other strings will be attached as the money gets handed out? Will we hear about all that?
Definition 3

**Definition**

**trans·par·ent**
[trans-pair-uhnt, -par-]
adjective
4. easily seen through, recognized, or detected: transparent excuses.

---

**Quote 3**

**Quote**

Kathleen Sebelius, Secretary of HHS

On dealing with the insurance companies:

“There are only 155 of them!”

Translation?

We can handle choking that many throats!

---

**Meetup - Geoff**

[Slides](http://semanticommunity.info/Healthcare.gov/Inside_Analysis)
Data Science for Normalized Data with a View into HealthCare.gov

Dr. Geoffrey Malafsky
CEO, Founder, PSIKORS Institute
gmalafsky@psikorsinstiute.net

About the Speaker

- Geoffrey Malafsky, Ph.D
  - Founder and CEO, TECHi2 consulting (past)
  - Founder and CEO Phasic Systems Inc. (current)
  - Founder and CEO, PSIKORS Institute (current)
  - Nanotechnology researcher, Naval Research Laboratory
  - Technology advisor and sleuth
    - DARPA, Office of Naval Research, Dept of Energy
PSIKORS Institute

- Established end of 2013 to promulgate key technologies to catalyze enterprise data normalization success
  - PSIKORS contextualization
  - Corporate NoSQL Data Modeling and Virtualization
  - Free non-commercial use
- Disseminate knowledge and lessons learned
- Facilitate community problem solving
- Guide awareness of true nature of challenge and knowledge of successful and unsuccessful approaches
- Provide community version of DataStar

HealthCare.Gov: What We Know

- Site experienced serious problems with handling capacity
- Site is experiencing serious problems with data aggregation and accuracy
- Program has had a lot of time to design, build, and implement
- Lessons Learned from Many IT Programs
  - Intricate interactions among many interface points and software components is nearly impossible to forward engineer so must be continuously tested and monitored
  - Human decision making is the most vulnerable and error prone part of the architecture
Contractors See Weeks of Work on Health Site

The HealthCare.Gov Data Veracity Challenge

- Web capacity problems to-date are relatively simple compared to expectation that multiple sources can be validated, merged, and adjudicated automatically, and within the timeline of a browser transaction
- This is rarely done anywhere and is not done successfully in large Federal Govt systems
  - Few if any examples of successful large data merging producing accurate data records
  - Many examples of large, costly, time-consuming failures
- DOD DIMHRS, FBI Trilogy, FAA, NGA GeoScout, Plus 10-50B more (yes B), Anything large using semantics
HealthCare.gov Can't Handle Appeals of Enrollment Errors

Simple Picture for Congress

This is the simple picture created for Congress so real architecture is many times more complicated. Note all the interaction points = points of vulnerability per transaction. Any one blockage will stymie transaction.

Questions:
1) Why this architecture?
   - Authoritative sources?
   - Partnerships?
   - Cost reduction?
   - Management?
2) Other strategies:
   a) simple web broker feed to other systems who handle sign up as part of their existing web based client interactions
   b) rent backend
   c) ……
The Issue is Data Normalization

- Most data environments are badly misaligned with semantically unknown relationships and value conflicts
- There will never be perfect data but you cannot even start doing analysis until you control your data and understand the good, bad, and untrusted
- Awareness and visibility are the crucial first steps
- Representation and semantics cannot be done with inaccurate combinations of data
  - Error propagation in probabilistic and heuristic approaches grows rapidly
Data Normalization

Data Normalization is the process of building and managing a continuously adaptive data environment that fuels current and future business needs for decision making and system operations.

- It ensures data (i.e. not just metadata) is as accurate, meaningful, and useful as possible while continuously adjusting to improve and add capability.
- It provides collaborative management of data assets, the designs governing who, why, and how of data, and the where, when, how of data use in operational systems.
- It solves the great challenge of mapping all source values to each target along the entire complex paths of enterprise data use.
Using Data Science to Solve Normalization and Enable Higher Level Functions (Semantics)

• Data Science is different than past IT hot spots
  ° "Science" binds it to a well structured culture, procedures, and ethics
  ° Science is fundamentally rigorous in maintaining auditable, open lineage of data collection, data rationalization, data analysis, theory comparison, adjudicating possible scenarios, and making conclusions
• Data Science is not analytics, Business Intelligence, warehouse design, Big Data, Cloud whatever, Hadoop, ....

Data Science Foundations Can Help

• Data analysis
  ° Multi-technique: statistics, models, graphical
  ° Understand the scope, limits, and biases or each (be skeptical)
• Making conclusions
  ° Iteratively second-guess, question, improve, openly receive fact-based rebuttals
• Need new techniques to capture human knowledge quickly, in context, from non-technical people, and seamlessly use in further technical efforts with very low cycle time
• PSIKORS knowledge and context model
• Corporate NoSQL data modeling and integration
The Ψ–KORS™ System Model

Corporate NoSQL™
Aligning Data Values With Business Context

- Suffix in source A, prefix in B, neither in C for same (part number, title, …)?
- Conflict syntactically (simplest case) and semantically (most difficult)
- Other tools & methods never solve this because they deal with the obstacles independently or not at all: Data values out-of-sync with metadata, data models

Data Value Semantic Errors = Inconsistent, Difficult to Merge, Report, Analyze
DataStar Normalization/Virtualization Platform

An Example Data Normalization Project

- Project by Data Governance Office and Enterprise Architecture
- Objectives:
  - Develop Enterprise Data Model that accurately represents corporate wide use (inc. to-be) of Company (partners, clients) and Insurance products (transactions, long-term), and has support of DGO and business units
  - Expand to other data areas and business units if successful
- Approach: Collect disparate artifacts as they exist and use DataStar to coordinate, correlate, and analyze to produce common data model
- Specific areas of concern
  - Limited documentation of major data systems
  - Variable versions of data models and definitions
  - Multiple groups owning and designing same assets
Clarifying Data Use and Meaning

Initial Results: First 30 Days

- On-site meetings identified multiple unknown conflicting activities
  - Risk Management Office developing CEO sponsored redesign of enterprise data integration, rules, definitions for risk analysis tool
  - Sharepoint Metadata repository difficult to maintain and limited content due to lack of shared input by business units
- DataStar ingests data models, glossaries, codes, corporate policies
  - Primary 9 data marts previously unseen (buried in ERWIN)
  - Company data warehouse
  - Metadata repository
- Automated analysis shows duplication, semantic variability, and items needing adjudication
  - Decisions made in-live meetings and implemented in DataStar in same day with DGO and business unit approval
Outcome of First 30 Days

- Risk Management Office agrees to participate and provide all of their information into DataStar
  - CEO sponsored definitions
  - Integration logic (Microsoft SSIS) and rules that will become corporate standard
- Company warehouse issues identified, adjudicated, and new common model developed in DataStar
- Metadata repository plan modified to have DataStar be central repository with DataStar ➔ Sharepoint conduit for corporate dissemination
- Project expanded to include Risk and other key data areas

Deep Dive

Slides

Title Slide

Guests
GUESTS

Host: Eric Kavanagh | CEO, The Bloor Group

Dr. Geoffrey Malafsky | CEO, Phasic Systems, Inc.

Dr. Robin Bloor | Chief Analyst, The Bloor Group

Computers Don't Lie

LOGS REMEMBER EVERYTHING
We Don't Fully Understand Computers

Systems Management and Operational Intelligence

Hence, SYSTEMS MANAGEMENT and OPERATIONAL INTELLIGENCE
HealthCare.Gov Issues

Community Awareness and Suggestions to Aid Recovery Effort
(Be Nice To Taxpayers)
What We Know

- Site is experiencing serious problems with handling capacity
- Site is experiencing serious problems with data aggregation and accuracy
- Program has had a lot of time to design, build, and implement
- Program has spent and is continuing to spend a very large amount of money

Lessons Learned From Others

- Very large capital investment required for backend and network
- Intricate interactions among many interface points and software components is nearly impossible to forward engineer so must be continuously tested and monitored
- Human decision making is the most vulnerable and error prone part of the architecture
Simple Picture Created for Congress

This is the simple picture created for Congress so real architecture is many times more complicated. Note all the interaction points = points of vulnerability per transaction. Any one blockage will stymie transaction.

Questions:
1) Why this architecture?
   - Authoritative sources?
   - Partnerships?
   - Cost reduction?
   - Management?
2) Other strategies:
   a) simple web broker feed to other systems who handle sign up as part of their existing web based client interactions
   b) rent backend
   c) ....

Govt Never Learns

• 10 secs of Govt debacles (10B in 1 breath)
  – FBI Trilogy
  – NGA GeoScout
  – DOD DIMHRS
  – FAA

• Devil in details
  – Business logic CANNOT be articulated with linear requirements process
  – Technical issues WILL continuously arise regardless of GANTT project plan
A Few of Many Gotchas

• Interacting time outs. Every software component in the chain has 1 or more timeouts
  – Any timeout expiration kills transaction to user
  – Some are easy: web server, app server
  – Some are opaque: DB concurrency, trigger
  – Some are unreliable: authentication

How to Solve 1

• Break the quagmire of misinformed decision making, naïve technical approaches, irrationally exuberant architectures
• Money does not grow on trees it comes out of your wallet (i.e. 1000 teenagers per adult instead of 1 or 2)
PSI-KORS: Sense Making from People to Execution Technical Architecture

Questions

http://semanticommunity.info/Healthcare.gov/Inside_Analysis
Updated: Sat, 19 Sep 2015 06:54:35 GMT
Powered by mindtouch
Thank You

THANK YOU!

All of our Webcasts are archived at InsideAnalysis.com

Spotfire Dashboard

Research Notes

Okay, the reason I thought this would all fit together was your excellent: What Went Wrong with the Obamacare Web Site, and How Can It Be Fixed?


http://insideanalysis.com/healthcare-gov-transcript/


Abstract: The high-profile federal Web site, Healthcare.gov, suffered an epic failure to launch on Oct. 1, but what, exactly, went wrong? In this inaugural episode of Deep Dive, we interviewed veteran federal contractor and Data Scientist, Dr. Geoffrey Malafsky of Phasic Systems, Inc., and Dr. Robin Bloor, Data Scientist and Co-Founder of InsideAnalysis.com, as they discussed the brass tacks of why Obamacare’s flagship sank in the harbor, and what can be done about it.

Malafsky provided granular details surrounding Healthcare.gov’s Web architecture, as well as the development atmosphere, critical strategic choices taken, the constraints of federal Information Technology acquisition practices, and the value of realistic data resolution practices and methodologies.

Malafsky discussed various ways to get the site up and running. In particular, Malafsky he outlined his PSI-KORS Methodology, which was used to save a $300 million data resolution project with the United States Navy.
So I thought you would just summarize this:

Eric: the brass tacks of why Obamacare’s flagship sank in the harbor, and what can be done about it.

Geoff: granular details and various ways to get the site up and running

Transcript

Source: http://insideanalysis.com/healthcare-gov-transcript/

Eric: All right, ladies and gentlemen, hello and welcome to the first ever Deep Dive. My name is Eric Kavanagh. I will be your host for this inaugural episode of a show that is designed to really dig deep into the details of something significant that’s going on out in the world. We can focus on technology. We can focus on methodology. We can focus on any number of different things. Obviously, the topic today is going to be healthcare.gov. We have several guests. Of course, there’s yours truly, Eric Kavanagh. I’ll be your host. We’ve got Dr. Geoffrey Malafsky of Phasic Systems and Dr. Robin Bloor who, of course, is the Chief Analyst here at The Bloor Group.

I wanted to say a couple quick things about this whole issue. Before we go into that, let me go ahead and share this graphic. This is a graphic that was presented to Congress just recently that shows the main contractors and it goes through what happened here, what the architecture of the site is, the healthcare.gov website.

Point 1: Context

In total disclosure, no one on the call here, at least in the presenter side, is actively involved in this project. I would tend to think that if we were actively involved in the project, we would not be allowed to speak about it. That seems to be the policy in play here, but we’re going to talk about this graphic, what it means, and we’re going to talk about some other stuff that we know for sure is the case.

Now, Dr. Malafsky is a veteran federal contractor. In fact, he’s doing a lot of work with the federal government right now today. He may even talk about some of that at length, but he’s going to talk about really some of the protocols that go into government contracting and how a lot of that stuff conflicts with what we’re trying to accomplish with healthcare.gov.

Obviously, there are some very significant constraints when you start dealing with contracting with the federal government, which I have to say, I was thinking about over the weekend, is ironic when you consider the fact that the last couple of years, we’ve had trillion-dollar deficits and this year, I think it’s about 700 billion is where it’s running. Obviously, the constraints are not that tremendous because we’re spending a lot more than we make.

Point 2: Look at the Computer Logs

Anyway, let me go ahead, and just work through what we’re talking about here today. Basically, the comments that I wanted to make were simply that computers don’t lie. Right now, my computer is telling me that my Mac is not here, which is why things aren’t the way they normally are, but… Computers always tell the truth. And they also remember

http://semanticommunity.info/Healthcare.gov/Inside_Analysis
Updated: Sat, 19 Sep 2015 06:54:35 GMT
Powered by marktouch
things, so that’s one of the important things to keep in mind, and we’re going to talk about that today, is that computer logs remember just about everything. That’s why you have computer logs. It’s for troubleshooting purposes. When I was at the Teradata Partners Conference just this past week, Mark Smith from Ventana Research made an interesting comment; he said, “Geez, I hope those guys have Splunk. Just throw Splunk at it and you can figure out exactly what went wrong.”

Somewhere in all those log files is the truth about why the site crashed. Now, we know a number of things and Geoffrey’s going to talk about them right now with respect to why it crashed. Of course, we don’t know everything. One of the interesting things about computers is that at a fundamental level, we still don’t know exactly, precisely how they work. That’s where this whole issue or this whole discipline of operational intelligence came from, and of systems management: you’re getting a window into what’s happening there in the world of your computer systems, in the world of your servers and so forth — and any of you who have ever looked at systems management software know it’s like watching a stock-ticker go by. It’s some fairly interesting stock, but you really need to know what you’re doing, obviously, to dig down into that, but the fact is that all those logs are there. You can find the truth if you work hard enough, and, of course, that’s what we’re all trying to figure out.

With that, I’m going to go ahead and hand it over to Geoffrey Malafsky. Geoff, the floor is yours.

Point 3: What We Know

Geoffrey: All right, welcome, everybody, and we greatly appreciate your time. Eric has done a great job laying this out and I’m just one of the spokespeople. This is really as the title slide says ‘A Community Awareness and Suggestion Endeavor.’ We don’t want this just to be a fun, ‘let’s discuss things.’ We actually know that the community has a lot of expertise in this area. At one point, Eric will give you a few email addresses and we actively encourage serious insights and serious suggestions. We have an absolute filter that the non-serious go away, but that’s what we’re trying to do here.

The subtitle here is ‘To Aid the Recovery Effort’ and the starting point is we’re going to accept that everybody on the contractor side and the government side is a serious person, they want this to work, that they have a lot of knowledge, they have a lot of expertise, they have a lot of capability, but that does not necessarily add up to success. We also want to avoid recriminations because that’s already being taken care of. That’s being well covered.

We also want to keep attention to what’s in parentheses at the bottom, which is … this is also a Be Nice to Taxpayers Day, because every time you’re reading the headline that they’re solving the problem, they’re solving it by opening your wallet. Just keep that in mind. If we want to go to the next slide …

There’s a handful in here and they really are talking points. The next slide is titled What We Know, and there’s four simple bullets here because these are things that we know. As Eric said, we’ve got to be very clear here that we’re not on the inside. We’re not part of the project. We don’t have the project documentation. We’re not funded as part of the project. We don’t have the detail of architecture or any of that knowledge, but what we are saying is as a community, there are an awful lot of lessons learned and technical knowledge about this type of system, and things that you should do and should not do.
There’s also a lot of knowledge about how these things get procured. One of those things, which I want to bring everyone’s attention to, is they don’t get caught up in what you’re reading in the papers or seeing on TV. The real problem here is called government acquisition. It’s not one or two contractors. It’s not Health and Human Services. It is a bigger problem, which is government acquisition, which has now almost pre-ordained the situation.

One of the things that we know that has been going on is that the website is experiencing serious problems with handling capacity. We know that it’s experiencing serious problems with data aggregation and accuracy, and that is a distinct problem from the number one problem that everyone’s talking about (capacity), and it is a far more serious problem.

**Point 4: Serious Problems with Data Aggregation and Accuracy**

This capacity problem is actually one of the simplest ones. It’s frustrating for many, but even if you solve that problem, what is far more serious is the underlying data aggregation and accuracy. This entire government program will fall flat on its face is if there isn’t extremely high-level accuracy, trustworthiness and meaningful aggregation.

We also know that the program has had a lot of time to build, design, and implement this solution. One of the other things that happens with government acquisition is the notion, “We just need more time and money,” and you really need to act like a parent in this situation and just say, “Actually, you don’t. Maybe you’ve just gone about approaching it the wrong way, and that we need to review that.”

Then the other part of that is the program has spent and is continuing to spend a very large amount of money. This isn’t $10, $20. This isn’t like a few thousand dollars for a cruise. You’re talking enormous quantities of money, and with this latest crisis interdiction, I can guarantee you that the people who are most happy today are the suppliers of the servers, so that would be Dell and Teradata and IBM or whoever has the contract, database software, Microsoft, and Oracle, and Sybase and the others, a bunch of experts in consulting firms, and they’re all going, “Life is good,” because what’s happened is the government only knows how to spend large quantities of money more, instead of doing any serious server redesign.

**Point 5: Lessons Learned from Others**

We go to the next slide which is titled Lessons Learned from Others. What do we know about high-capacity sites? Obviously, there are very successful high-performance sites out there from all the search companies, the social media companies, etc., the e-commerce companies. And we know from them — which has been described many, many, many times in both the popular press but also technical journals like Engineering Society, like IEEE have been writing about this for years now or ACM — that it requires an extraordinarily large capital investment for the back-end and the network. Again, those are two distinct entities.

The back-end is all of the servers and the switches and the routers and all that stuff, and then the network is something distinct from that. Both of those have to be managed. We know that there are a lot of intricate interactions among the many interface points in software components, and it is almost impossible to forward-engineer that. What do we mean by forward-engineering? A good engineer with the best case scenario says, “I’m going to lay down all the things I need to do. I’m going to lay down all the pieces and parts I have. I’m going to lay down how I build that in. I’m going to take in
best practices. I’m going to build it up and be risk-averse and I’m going to put this all together.” That’s great but in IT, we do not have that capacity today.

Lots of people sell as a consulting service or a hardware or a software application that we do (have that capacity), but then you have to read the fine print and it’s “only on Tuesday when there’s a full moon and Sally is back from vacation.” In reality, IT is still a chaotic, learn-as-you-go environment, which means that you have to continuously test and monitor, not just two weeks ahead of deployment, not one month ahead of deployment, not two months ahead of deployment, but a lot of time and then continuously. You need to be prepared for all the inevitable problems that will continuously ensue.

The other major lesson learned that we have is that in this type of project, particularly from federal government acquisition, human decision-making is the most vulnerable and error-prone part of the architecture — by far. That decision-making starts even before anybody gets hired. It starts in what is called the requirements process, by which these people sit down and build the strategy of what they want the program to be. They even go so far as to build the strategy of what the technical solution will be.

Then it goes to the next phase of issuing request for proposals and getting stuff out, and then it goes to the next phase, which is actually the selection committee. When you marry that up, you’re going, “You know what? It’s pretty obvious if you read between the lines that that process exists more to propagate the acquisition process than it actually does as if it was your money in your house trying to build the most high-performance, cost-effective solution and you were supporting your own client.”

That decision-making process that you would personally go through is not part of the acquisition process. The people who will most vociferously disagree with that statement are called acquisition professionals, and I encourage them to show up at my house and we will sit around and debate this, because I have a lot of empirical evidence to back that up.

Point 6: Simple Picture Created for Congress

We’ll go to the next slide, which is a slide that Eric showed, which was showed at the Congressional Hearing by the contractors about what happens behind the scenes when the user is trying to go through and look through all these policies. You have to remember one thing. One, this is the simplified version of the architecture that has been produced for Congress. The real architecture is five to 10 times more complicated and intricate than what is actually being shown.

The first thing, we look at this and go, “They have some idea of where things are going,” but the first thing you should notice here, if you can remember what Eric showed at the outset, is all the different points of interaction. Anybody who has ever built a real system of any kind goes, “That is a problem. That is going to be a source of a problem.” Some of those problems I can predict. Some of those problems I can prepare for, and a whole bunch of those problems are just going to happen in a random, chaotic, ad hoc basis, and we’re going to be happy to prepare for that.

Point 7: The Web Environment Was Not Originally Designed for High-performance Applications

The bigger issue is, is that the web environment was not originally designed for high-performance applications. The web is based upon a very simple linking and page-oriented capability. Where is the big difference? In applications, there’s a
lot of real actions that occur that take a lot of time. It takes a lot of computer time. It takes a lot of time calling the storage media, takes a lot of time calling a different media to get some information.

The problem is that all of those things add up, so as you go from your client and you click on a browser, the first thing that happens is your browser’s an application. It has its own timeout. Many of you who use Internet Explorer will see that and sometimes the browser will pop up a warning that says, “The script on this page is taking too long. Do you want to end or go?” That’s okay if you’re just reading the Lands’ End site shopping for some gloves or something, but if you’re trying to do a heavy-duty application and you know it’s going to take a long time, then, “No, I don’t want to stop. I want to get my answer. I want to get my product out of the door.”

The first thing we should ask is, “Why was this architecture chosen? What were the rationale to choose this architecture.” Some of the possibilities are, which are typical, are we want to use authoritative data sources. We don’t want to carry the data ourselves, so we want to go to the authoritative data sources. One of the obvious ones here is getting financial data per person to determine if they deserve a subsidy or what their price point would be, so we go off to the IRS. Now, we have some link over there, but that link is live. That means as the user is sitting there waiting at their computer screen, that has to make a link over to the IRS systems. In a perfect world, that link happens, the computers talk, I get my result, and I come back.

What about in the real world? What about when the IRS systems are overloaded? What about when they are at capacity? What about when maybe they’re doing maintenance? What about it’s a network between the network operating center of the entry level webpage that the client sees to the IRS center? Maybe there’s some problems there. Maybe there’s a virus. Maybe there’s a Trojan horse running around and then Telecoms have shut down things to solve that problem. That will kill the transaction from the point of view of the user. That is just one of many such points in this architecture. The authoritative data source went, “That’s a positive, but that has a negative associated with it.”

What are some other possibilities you have chosen this architecture? Partnerships across agencies. HHS wants to partner with IRS, wants to partner with this, wants to partner with that. That’s a good thing. We like partnerships in government, so that’s a positive. Cost reduction, spread the cost out amongst many different government agencies. The IRS, you run all that. HHS, you run all this, you run this. That’s a great idea, I’m dubious about that because I have never seen that cost reduction show up in real dollars on a real contract. Rather it usually goes up as you’re seeing today.

What is very, very typical of IT programs is that the selection is made at a low cost and everybody who’s in the game knows that the cost will go up wildly because these situations occur. People go into crisis mode, and the only thing people know how to do in crisis mode is recriminate and dump huge quantities of money on it in no particularly sane manner.

The other one is management, and this is acquisition. It is a turf battle of management. Success is like old tribe of city states and tribal community. Each group whether it’s in … it’s not even in agency. It’s not even at the agency level. It could be a group within an agency. Their success is measured by what size program they own and manage. It’s not whether or not all those other things should be taken care of. If I can own a $5-billion program, I’m important. I’ve got big stuff. I have a job. I’m not going to be laid off, those type of things.
Point 8: What Are the Alternatives to This?

What are the alternatives to this? This should have happened way back when. The other alternatives are, we know that this is one possible architecture. It has some positives for it and has some significant negatives, but what else is a possibility? We know that another possibility is to have created a simple, very simple web-brokering system that all it does is by very simple app server code and even simpler client side Javascript creates a very pleasant interface that produces rolled up data to people. Here’s what you can do. Step through this. Step through this. Then any action that occur can be done at selection point and sent off to maybe somebody who’s actually going to own the actual program.

If the overall arching government program is set up such that the insurance companies actually own the client after they get a commitment, then why not just push the traffic off to the existing client interaction environment channel that the insurance companies already have? Yes, they might need to augment their own, but that would be a valid business reason because they're now going to get new clients. They can go inside and they can plan and they can grow and they can do their own thing, and that would be an extraordinarily simple relative to this architecture, cheap, easy to do, value added architecture. That’s one.

The other one, which is what’s going on now, is you have to ask the very serious question. If we know from all the search and social media and ecommerce companies that there’s this enormous capital investment in the backend, why would the government get into that? What possible reason would the government get into that? They don’t have those capital resources. They do. They can just come to my wallet, but the other companies, they go on the stock market. The stock market is a source of capital. They can invest that capital in their backend and then they can tell their shareholders, “See, this is how we make money. We provide a high quality very fast experience to our search or ecommerce or social media people. Our valuation goes up and everybody’s happy.” Government can’t do that. They are dumping money on this, and I would raise a serious question whether that wasn’t a critical failure factor from day one.

Slide 9: Government Never Learns

Now, we go to the next slide, which is titled Government Never Learns, and again I don’t mean to be a recrimination but the villain here is not people. The villain here is not contractors. It’s not the specific agency people. It is the overarching system and the one I’m showing here is a little empirical evidence to back up that claim.

I’m on Government Never Learns, and so on that, the first bullet is 10 seconds of discussion off the top of my head, and lots of other people listening in this area can come up with their own list, and if you have those list up, but here are some of the larger, more well-known debacle, and we just find the debacle is a very large amount of money spent, a lot of time, it went through the very rigorous formal government acquisition process and when it ended, it was openly acknowledged by everyone to have been a failure and there was no leave behind. We’re not talking, “It worked. We’ve got a lesson learned,” blah, blah, blah. We’re talking about nothing left behind.

One is FDI Trilogy, which was about $250 million and they went on to do what is going to be a stroke of brilliance from bureaucratic point of view. They did a full-court press on the actual mainstream press and said it was a success because they learned why it failed. Whatever you think. I leave outside of DC area, that was just a stroke of brilliant bureaucracy. Snatch success from massively costly defeat and they took that lesson, they went to Congress and said, “Now, give us more money for Son of Trilogy and maybe we’ll learn even more that way,” and Congress goes, “Yeah, that makes a lot of sense. Here’s $400 million.”

http://semanticommunity.info/Healthcare.gov/Inside_Analysis
Updated: Sat, 19 Sep 2015 06:54:35 GMT
Powered by mindtouch™
So NGA GeoScout, that was about a billion dollar of debacle. DOD DIMHRS, which is going to roll up HR systems across the DOD, bailed out at about a $800-million level, nothing left behind, nothing useful. FAA, I won’t mention contractors there. Let’s put some news now in your computers to this decades old system. That went down in flames. That’s just in the first breath. Those add up to be about $10 billion, nothing left behind, hence, the debacle title.

The reason for that is … again, the problem there was not people. Nobody needs to go to jail. Nobody needs to do anything. The devil is in the detail. One of the critical failure points that permeates particularly government IT projects, is this legacy, archaic, obsolete notion that you can articulate all the necessary business logic with some linear requirements process. That fundamentally does not work with large IT systems. But they’re addicted to it. They just can’t stop, so they build architecture frameworks, which by themselves, are great. You have the FEAF, the Federal Enterprise Architecture Framework. The DoD has DoDAAC (Department of Defense Activity Address Code), and so then they go, “Great. We’re going to pay a lot of contractors. We’ll spend a year or two years. We’ll do all this. We’ll give it to our acquisition professionals. They’ll do all this stuff.” Almost every single instance that has occurred has been a debacle failure.

The second part is that the technical issue is the real issue. The ones that really will be bedevil you that you have to deal with will continuously arise and they will arise regardless of whatever Gantt project plan you put together. You’re going to have the most decomposed Gantt chart you want. You’re going to have thousands of individual taskers and have a resource loaded, and you know what? The real world will ensue and the real world will say, “I don’t really care. The app server’s timing out and you don’t know why and I’m not going to tell you why,” and that’s what you have to do about that.

Point 10: A Few of the Many Gotchas

That allows us to go onto the next slide, which is titled A Few of the Many Gotchas. I’m just picking out here some that I actually learned the hard way in the government contract, so I’m just saying this. Again, by no means, this is comprehensive or even the ones killing the system, but it would be hard for me to believe that this isn’t a problem keeping some of those engineers up at 3 AM. I can almost guarantee that there is a beehive of activity at 3 AM right now going on in these things.

Those timeouts: If you remember that graphic, every single piece of software that touched in that graphic had its own timeouts, and it’s not even one timeout. It can be more. The expiration of any one of those will kill the entire transaction. Some of those are easy to set up and monitor like Eric talked about, log files. Those are like the timeouts on the web server and the app server. Some are more opaque. You have databases with concurrency and triggers, but they’re multi-interaction. If you really do a deep dive into how databases work, it is not a pretty sight.

The database servers love to say, “We keep everything orderly.” Not really. The one only that they can get the performance up and truly manage it is that there is a series of time stamped files that are created on the storage, persistent storage, and they are not rolled up into one comprehensive accurate set of data that’s available for anyone at any time because that takes too long. That would kill the transactional latency.

You have to look in those details and then that’s rolled up through a management interface and that goes by some very nice sophisticated names like triggers and concurrency, but it basically means it takes a bunch of time to go get the data, update the data, and if I can’t do it before another request comes in, I’m just going to tell you, “Forget it. I’m closed for business.”
Then there are some that are notoriously unreliable and that is going off to this third-party authentication services, which always take a long time. When you add that up, you would have to say — and lots of people have been down this path before, and lots of people have suffered these problems before — and you would have to say, “We’re either going to go down the same pathway or we’re going to learn from them, and maybe the best strategy is to avoid some of this.”

Point 11: How to Solve Number One

That gets down to the next of the last slide, and the slide is titled How to Solve Number One, because again, we’re saying there could be hundreds of these slides and you folks listening have a bunch of information. You have a bunch of expertise. You have a bunch of lessons learned. We want to encourage you as a community to send these in. We’re going to take on the job of consolidating them into a nice succinct, understandable format, and then Eric is going to actually go take them to the mainstream press, the Wall Street Journal, the Washington Post, because what we want to do is to try that person who’s working their butt off at 3 AM. If we can get some of these out there, it may spark an idea or two or it may aid them in solving this, because remember, they’re getting paid by our wallets.

Point 12: PSI-KORS Metamodel

The first thing that has to be done is breaking the quagmire of misinformed decision-making, naïve technical approaches, and irrationally exuberant architectures. That is what we’re showcasing now in the last slide, which is what we have, which is the PSI-KORS metamodel, for which there is now a PSI-KORS institute because we’re using this. This is really and depending at what we do or day job, which is data rationalization for large-scale system. The key part there was, as everyone solves the problem with data rationalization in their enterprise stores, you have the exact same problems. People don’t know what to do, what’s going on, who’s doing it, why it should happen. The actual data elements are somewhat irrelevant in that regard because it’s the meaningfulness of the data values, and that’s what we concentrate on.

We created the PSI-KORS meta-model to be purposely simple, intuitive and drawn from industry and peer-reviewed research standards, and we’re offering it out as part of the side course institute for non-commercial, let me repeat non-commercial, non-commercial widespread use because it really is a catalytic enabler for people, because what happens here and there’s lots of meta-models in the world, but what happens here is you can see by the color-coding is we’re gently guiding people. The people who live and breathe, the experts in organizational issues provide their snippets of key knowledge and it gets captured, and it gets related easily to this different, different people who were experts on the business activities, and then the different people who were experts in the technology.

Because what happens at a normal meeting is that the business people want to talk about technology. The technology people want to talk about business. Everybody wants to be an expert in what the organizational is, and you end up with a lot of chaos and poor decision-making.

Today’s Wall Street Journal has an article, I’m pulling it right here. Here’s the title of the article on the front page of today’s Wall Street Journal, ‘Health Sites Stymied by Lack of Direction’. You know what? That’s nonsense. This is a very expensive government program. There was a program manager. There was a lot of oversight. There were project managers. Every single contractor had a lead program manager. They had team managers. They had technical managers. This program was replete with managements, so don’t let them tell you that.
What it means, it was stymied by the lack of coherent direction, which is completely different. This PSI-KORS metamodel, quietly, without organizational friction, guides us to figure out, “If we’re going to have that interface, what do we know from somebody else’s experience? Maybe that we don’t want that interface. Maybe that’s not a wise thing.” It really engenders corporate knowledge snippets, consensual awareness and decision-making in real time, and we have been able to use this in steering committees of mixing executives with actual data engineers that in less than one hour has uncovered these issues, made these decisions, and made everybody happy. It’s a matter of getting the right information in front of the experts, getting them to ask the right questions, and make the right decisions.

Let me stop there and pass it back to Eric.

**Point 13: Interlocking Timeouts**

Eric: Okay, good. All right, so here are the slides we went through today. Let’s go back to this graphic here. I thought one of the more interesting things that you pointed out, and this is honestly, folks, is why this whole webcast came into being. I was talking to Geoffrey on Thursday or I think it was Thursday or Friday, and we started talking about specifically the interlocking timeouts, and the fact that if you have one of these timeouts across seven or eight different locations in the architecture, the whole thing crashes. I’m not sure if we talked about this, but I think it’s even in this graphic. Geoff, and maybe Robin, if you want to chime in here as well, you can comment on this, but a couple weeks before the launch, a decision was made to change the architecture or the web flow as people are logging into the site. What they did, because originally from what I hear, you were supposed to be able to log on as just some consumer out there, a potential, a prospect I should say for this particular solution, and browse through the different plans that were available. Once you found a plan that you like, you could say, “Now, I want to go ahead and see about purchasing this plan.”

That was all changed just several weeks before the launch and instead, the web flow had you creating a user account first, right away. The big issue there, of course, is if you’re creating all these different users simultaneously and there was a ton of traffic, that’s all going all the way down to the database. I guess, Geoffrey, I’ll ask you to comment and then, Robin, if you want to comment on that as well. That puts a huge strain on the database, right?

Geoffrey: Yeah. Again, there’s probably some purists out there that violently disagree with me and that’s a great reason to get together and talk about it. Part of everything we said in the PSI-KORS is predicated on the notion — and I know there’s some people from database vendors out there, please sit down before I say this thing — that the database is the absolute wrong, wrong, wrong place to have any business logic, any. Anything above zero, anything operate. That is an archaic approach. The databases are ungodly inefficient in doing any sort of computation, and so you don’t want to do that. It also violates the best practices of modern RT architecture, which is called the services architecture in which case there should be zero. That’s a zero business logic buried in the database. That even gets into the data modeling, which is what we preach. You should not be having business logic in the data model. You should not be having the database server execute business logic. It should all be up at a higher performance application layer, one that is visible to management, because one of the problems with putting in the database is it also doesn’t have any management oversight. One of the outcomes of that is ObamaCare. That is one of the outcomes, and that even shows up at the data integration layer, ETL. I know we’re getting a little technical, but almost no ETL ever sees the light of day, actually has any management oversight, but that produces the executable data.
That is a critical failure factor. That was known. There are a lot of purists in the world who say the opposite, that everything should be there, and then I would just say, “Prove it.” I would actually reach out to all the main vendors out there and say, “Take the last 10 years and prove that that has been a cost-effective approach relative to having zero at the data level and having all business logic up at application level.

**Point 14: Separating the Know From the Flow (Be Informed)**

Eric: We have an attendee commenting you’re overloading the term ‘database’ and says you have to store the business logic somewhere and that might as well be called the data store. That’s a semantic issue. I think when you’re talking about...

Geoffrey: No, that’s actually a software issue. The business logic does not “store”. The business logic shows up and this is what program and code is all about. The business logic shows up as functions and methods in high-level code whether it’s C-sharp, or Java, or C++ or whatever it is. That’s the embodiment of the business logic. You can capture the business logic in a documentation format and can be any architectural guide you want, but you don’t “need to store it” in a database. That’s not the logic. The business logic is actually what you want to happen, the logic, the rules, how things happen, and the problem is that evolves. That evolves over time. It has to be governed.

We’re not making it look opaque. What we’re doing is saying that the execution of the business logic, which must occur in code somewhere, even if it occurs at the database server level, it’s occurring in code. It’s where in the stack you’re putting that code. It has to be as close to the governance process, so it has visibility and it has to have a low-cycle time, so like you were just saying two weeks ahead of time. If governance changes its mind than what it wants to have happened, it has to be a very rapid cycle time between when that decision is made and to when it becomes executable. There has to be visibility there. It’s the lack of opaqueness that we’re after here, and so then you have to question. If you look at an actual technical stack of from the client browser all the way back to all the way servers, where can you gain the most visibility and the easiest ability to have change on a rapid-cycle basis.

Eric: We’ve got a couple other questions here. Hold on one second. Here we go. One of the attendees is writing, let’s see, “I respectfully disagree with Mr. Malafsky. Certain business logic needs to be within the database.” I think that it’s horses-for-courses thing there, but I think that the key that we’re trying to discuss right now is that if you have a site where you expect this amount of traffic to occur, you really need to be even more lean and even more strategic than you would normally be. I think the point you made at the start of the show here, Geoff, is that sites like Twitter and eBay and Google and other sites of that nature, a lot of those are public companies. If they’re not public companies, they get venture funding, so they have a tremendous amount of money that they invest in their backend.

The question that you raised, which I think is the number one question, is since this just a brokering site at the end of the day … in other words, you don’t buy insurance; you don’t buy insurance from Obama Care. You buy it through this website which links you to other insurance companies. You’re actually buying the insurance from the insurance companies. That hasn’t changed but there is this conduit. I think the point you raised is since it is just a conduit, why not just create a brokering site that links directly to all the insurance companies and let them handle the extra bandwidth, right?

Geoffrey: Yeah, exactly, and that goes out to the original strategic architecture, which is why I come back and say, “If you really wanted to point a finger at something here, it’s long overdue. The entire government acquisition process is...
dysfunctional. You’d be hard pressed to find anybody inside the government other than the official acquisition professionals who think otherwise. You do not get taxpayer value out of it. You very rarely, rarely get a fully functional IT system out of it, and you even more rarely get a working IT system that is modern in any particular view.”

Eric: Robin, if you want to comment on that?

Robin: Yeah. The history shows that it’s true in Europe. It’s true in the UK where I come from and seems to be true in America, and it’s not really tailored to actually have good outcomes. Geoff mentioned earlier, the thing that they could usually have done here is to unlink the dependencies. You actually have a broker upfront, but you can also have individual systems for individual areas of the country if you really wanted to. This is not an insoluble volume problem, because it was painted like that in the first instance and, yes, the people that were, I say, in charge of design upfront will have known there will be millions of people going onto the system. If you break the system up and you help breaking in front, then you don’t get very big systems.

It seems to me that the design principle that’s gone here with all the integration of stuff here has utterly and completely ignored that. If that’s the case, then somebody who was in the thinking stage of this knew nothing about computer systems.

Eric: We’ve got a couple other good comments from the audience here. Geoff, I’ll throw this one over to you. One of the attendees writes, “Agile cycle times require the integration of upper case and lower case tools, meaning computer-assisted software engineering, where you can change business rules at higher level and propagate the changes faster and more reliably. I think this does speak to what strategy should be for a site of this size is you want layers of abstraction. You do not want very much to be hard-coated. You want more of a service-oriented architecture such that if you discover a problem in one piece of the puzzle if you will or one piece of the stack, you can address that problem without there being countless other manual integrations that have to be undone, right?

Geoffrey: Yes. Let me clarify because I’m reading some of these things. The term ‘business rules’ means different things to different people. Some of the comments are coming back from a technical view of business rules where there’s actually business rule systems that would execute the business rules, et cetera. I agree with that. I’m not disagreeing with that at all. That higher level is certainly a major value added to any application stack like this.

What I’m talking about is going further up to the human side where a business rule is when you talk to CEO and you say business rule, he’s not thinking about BPEL. He is not thinking about “a business rule system”. He didn’t know they exist. What he’s thinking about is what he wants to see happen in his organization and instantiated in a “IT system”. That’s the business rules I’m talking about, and those business rules have to be part of the day-to-day governance process and, therefore, highly visible. Then when you make a decision, you will go do this on Tuesdays when Sally is not on vacation. We’re going to let everybody buy an insurance policy for $1. Then that has to quickly show up not months later. That should show up in execution no later than a day or two later.

Making that happen, yes, can certainly be added by a “technical business rule system,” but it can also be added by just having a good design of the software stack such that you can go in and pay a mid-level engineer who does some Java programming and say, “Go change method X172 and change that if then loops, that we do this, this, and this,” bada-boom, bada-bing, it’s done, it’s execution. That’s what I’m talking about. There’s lots of ways to get at that end, but it
means that you have some good design like Robin said ahead of time. Again, I’m pointing the finger at government acquisition.

Eric: Here’s actually a really good point from one of the attendees who write, “However, the key to the affordable part is calculating the subsidy and insurance companies cannot do that.” That’s an excellent point. Basically now, this is where we’re talking about connecting to other federal systems that have other data about individual citizens of the United States and to be able to calculate the potential subsidy. It seems to me that we have sort of a circuitous route that we need to take because of this. It would seem to be a separate module does just that piece of the puzzle, meaning you get to a certain point in your process. You decide you like a certain plan in terms of what it offers. Then you get to the point where you have to figure out what that thing is going to cost. The end which determines the subsidies, that have to be pulling from other government systems, right?

Eric: It can and cannot. Again, that could [inaudible 00:46:22] a strategy, so I don’t want to say that those original value added things like partnerships and distributed management are not good things, but that did not require you to go to these architecture that they chose. Anybody who knows, let’s say, two of the biggest CPU cycles, so for CPU cycle intense applications. Authentication is always that way. That’s always a problem.

The other one is like you said where you actually now have to compute something meaningful. All right, that’s an actual application in and of itself, and that’s an excellent example of the business rules that will evolve on an almost daily basis. There has to be an intense visibility, because in some of the contracting I did, some similar business rules which are buried in the data layer, the program managers didn’t even know that they were there. There was no management oversight of the actual operational de facto business rules, which show up in the data, because everyone was so buried. Nobody had any visibility of it.

That’s the thing that should be in a separate application, but then from an execution architecture, you go, “I can pay the person to do this over there, there, and there, but now let’s go look at our transactional latency.” We want a client to have a good use case. Do we want to really call the computers stuck in the IRS data center? What is their history of performance? What latency do they have? They say they’re going to do it and I give them credit for being optimistic, but what can they really do?

Maybe in order to reduce latency and vulnerability, what we’ll do is we’ll have the IRS produce and manage that application, but they have to ship it to us and we will have a version-controlled execution on the same server. There’s no crosstalk. There’s no interfaces. It’ll just be within the same server enclave and, therefore, we’re taking out of the risk equation some of those interfaces. If it was me, that’s me personally, so again some of this is personal opinion. I’m risk-averse. That’s what I would do.

Point 15: List of Government IT Failures

Eric: We have another comment from an attendee who writes, “The list of government IT failures is not helpful. IT failures are not sole providence of government with all project self-worth exceeding magnificently.” That’s true, but I’d like to point it out in just a texted answer to our attendee here. The big difference is that if some corporation sites failed, I don’t get fined because of it. The key to remember with this new legislation is that all citizens must purchase healthcare.
Now, you can go through the usual channels, of course, but if you are one of the people who needs help, meaning you’re one of the people who needs a subsidy to get the healthcare, you must go through this system. One thing I read just the other day, and let’s comment on this for a second, is that even the paper side of this whole process, meaning even if you just go ahead and do the old fashion way where you fill out forms and you mail them in and they are processed by someone on the other end, even the people who process those paper forms are required to use this website. Not even the manual side of the operation can be functional until the website is functional.

I’m just curious to know. From my perspective, any time you have a manual override, you do not make the manual override contingent upon the automatic version. The manual override is supposed to be a whole separate process.

Robin: Unless, of course, you want to run with plan A and there’s no plan B.

Eric: That’s the thing, so it’s really not a manual override at all. By definition, that is not a manual override, right?

Robin: Right.

Geoffrey: I’m responding to that right now as well, and again, this is all respectful. These are opinions, but what I just said is that not only is do I disagree with the list of government IT failures as not helpful. I’m saying it is absolutely required. It is required not from data management or IT. It is required from management. Pure management requires you to assess and analyze and learn from prior actions of similar type of programs and particularly failures. If you’re not doing that, I would make the absolute claim, you are not acting in a good management manner.

Point 16: Opportunity For an Open Source Project

Eric: I just posted the link of the slides again here, folks, so you can download it. We’re going to keep focusing on this to try to better understand what’s going on. Here’s a question I was going to throw out. If you look to the Apache Foundation and you look to all these different open-source projects, of which there are many, many, many. Of course, Apache Web Server has been around, I guess, for 15 or 20 years now, maybe longer. Doesn’t it strike both of you as an awesome opportunity for an open source project? Because that’s the other question here is, if you build a site of this magnitude using technology that is not seriously up-to-date, you run the risk of having all the old problems that you always used to have. Whereas if you take more of the service-oriented approach that we talked about, it’s easier to rip and replace component parts when they fail or when they break or when they’re simply not up to snuff, but this sure strikes me.

Robin, I’ll ask you, it strikes me as an awesome opportunity for an open-source project to go to the Apache Foundation and say, “Hey, guys, could you help us out? We have a site up and running now. It still has glitches and so forth, but we want to architect the best site of the future. Can you help us and launch a project and launch an open-source project where you can get all these great minds from Google and Twitter and LinkedIn and all these different companies out there in the Valley helping out to design much more of a streamlined system for the future?

Robin, what do you think about that idea?
Robin: If you did that, you’re still going out to some contractor of some sort. Geoff is completely correct to focus on this: the process is how this stuff is procured, because at the end of the day, there are a lot of people. This is not an unsolved problem. Let’s not look at this and say, “This is such a hard system because nobody ever tried to do anything like this before.” There’s a system in Holland that does something like this, right? This is not an unsolved problem.

From that point of view, they’re ought to have been that say, choices that could have been made that would reduce the risk. One of the things, this is a very big project. Very big project, high risk. The industry has been telling us that for 40 years. Do a big project, high risk, high risk of failure. To have three and a half years sounds like in a modern day that that would be enough, but high-risk project and … and it’s all turned out badly, and to have the final thing that did me, almost had me burst out laughing is no integration testing until two weeks before you go live, and that’s just like, how could you ever do that with something like this? How could you?

Eric: Yeah, that’s wild. It’s up there right now. Folks, please feel free to send all of your ideas. You’ve got my email address. I sent it in. Just for a couple minutes here, I want to focus on the PSI-KORS model again, and talk about how something like this could be used for these exchanges. Could you maybe just take one piece? Let’s look at processes right here. That’s one of the key considerations. When you look at the rollout of the project, the fact that the web flow was changed shortly before the whole thing went live, that’s obviously not something that you want to do, right? You want to map that stuff out well ahead of time, and then you want to make sure that you don’t change at the last minute, and if you do change close to the last minute, you’ve got to do a mountain of testing, right, Geoff?

Geoffrey: Yeah, and then the PSI-KORS really here is to aid … what this really does in a very simple manner, which is the goal of most metamodels, is to get issues out and to disambiguate where things are confused. Again, we do this mostly for data rationalization. Where this really pays off is people focus on “the metadata” which is almost an irrelevancy because the metadata, if it’s not synchronized with the value stored in the database, then really don’t reflect it. People talk about that year after year after year.

What this does is get that out in the open and scopes it and does all that, but even at a higher level of what you’re talking about, and so let’s say somebody made that decision. I understand the decision. I can’t judge the decision. You will or will not let a consumer do this. Then you go with this and the idea is you will quickly go, we all know we need an impact analysis, but let’s just get down to the big things right now. If I’m going to change that business process, I know it has ripple effects through everything, but let’s just take the big things. What do we know off the top of our head are the technology interfaces and components of the effective? I don’t even need to go past that. Just what are the major effects? Because without something like this, what’s going to happen is they send the data call to each contractor and going, “Give us the change required for this and the time,” and they all write back some small thing. They don’t want to write back too many because then they’ll look like they’re not experts and they won’t get paid, so you underreport the problems.

Here, this going from a no-fault situation. This is not a contractual situation. We just want to get a sense made of what’s really going on and what would happen is that one guy would go, “We’ve got to change these components and these interfaces but it’s no big deal,” and somebody else would go, “It is a big deal. We just spent three months testing that interface because it had to go off to systems X, Y, and Z and it had to pull data. We had all these timing problems. Actually, it really is a big deal.” That information I just said, which takes 10 seconds to say, usually never comes out into the open.
Point 17: Wrap Up

Eric: I'll tell you what, folks, we've had a good time here just learning from some of you talking about what's going on with this website. Obviously, it's a pretty important deal. We all want this thing to work, so if you have any other comments on it, feel free to send an email to yours truly, EK@mobiusmedia.com. We had some fun with our own little technical issues here today, but we got you all back online, so thank you very much for your time and attention.

With that, I guess, we're going to bid you farewell and keep your eyes peeled. We've got some other events coming up. We have one tomorrow with IBM talking about what's going on with data visualization. We have a lot of other events scheduled for the rest of this year.

With that, I'll say, we bid you farewell, folks. Thank you very much for your time and attention. We will catch up to you next time. Take care. Speakers, standby for a quick post-briefing. Bye-bye, folks.

NEXT