

...the art of the sexy database

Christopher F. Burns, Sr.

GEMKO Information Group - Buffalo, NY

cburns@gemko.com



Why are we here ?.....

RUNDOWN

Y ASK Y

LABATT'S

TOM PETTY

SANDBOX

U.S.G.S.

MR. BUBBLE




THREESOME

GANG

IN TOUCH

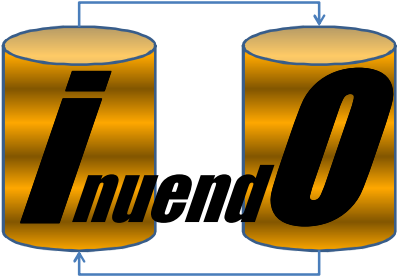
MORE...

- To get a fresh perspective of the art of data architecture
- To seek an escape from the tired old conventions rooted in the 1970's
- To shift responsibilities from the program stack to the database
- To make our data friendlier to other platforms and applications
- 'Cause there's food involved
- Others ?



On tap.....


RUNDOWN	
Y ASK Y	• Dirty little secrets about data
LABATT'S	• Prism model of supply chain interaction
TOM PETTY	• The entity concept
SANDBOX	• Time travel technology
U.S.G.S.	• Implementing as part of a modernization strategy
MR. BUBBLE	• Why GEMKO ?
THREESOME	• Q & A
GANG	
IN TOUCH	
MORE...	





...the art of the sexy database

Dirty little secrets about data







Your database...

RUNDOWN

Y ASK Y

LABATT'S

TOM PETTY..

SANDBOX

U.S.G.S.

MR. BUBBLE


THREESOME


GANG


IN TOUCH

MORE...

- Efficient or deficient ?
- If asked what comprises your data, or more specifically, a typical database file, what would your response be ?
- Perhaps a breakdown like this ?







Database fields by purpose (perception)

RUNDOWN

Y ASK Y

LABATT'S

TOM PETTY.

SANDBOX

U.S.G.S.

MR. BUBBLE

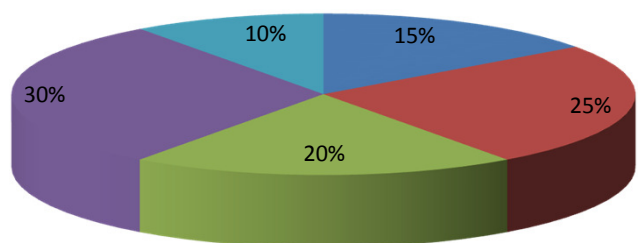
THREESOME

GANG


IN TOUCH


MORE...


■ Key Fields
 ■ Descriptors
 ■ Buckets
 ■ Flags/Codes
 ■ Misc.



Virtually 100% productive 😊







The dirty little secret (reality)

■ Weak Data Types

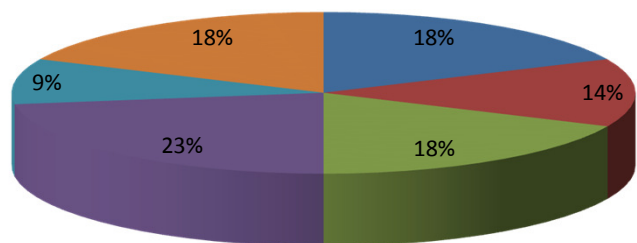
■ Clunky Keys

■ Duplicity

■ Repetitive


■ Unused Space


■ Truly Strategic




Maybe 18% productive ☹️

RUNDOWN
Y ASK Y
LABATT'S
TOM PETTY
SANDBOX
U.S.G.S.
MR. BUBBLE
THREESOME
GANG
IN TOUCH
MORE...










Your legacy data is made up of...

- Weak, inflexible data types
 - Why ? Because legacy tools like AS/400 DDS were still used religiously even after SQL became available.
 - Application code required to address shortcomings.
 - Doesn't play nicely in the sandbox with other platforms.
- Clunky complex keys
 - Why ? Because these systems were built piecemeal.
 - Application coding required to facilitate.
 - Major root canal when corporate standards change.
- Fields available elsewhere (duplicity)
 - Why ? Because in the early 90's, even though disk was cheap, CPU was not. And many HLL's had file limits.
 - Application code required to synchronize, in many places
 - But it still gets out of synch (then fix-it coding required).




RUNDOWN
Y ASK Y
LABATT'S
TOM PETTY
SANDBOX.
U.S.G.S.
MR. BUBBLE
THREESOME
GANG
IN TOUCH
MORE...

Your legacy data is made up of...

RUNDOWN
Y ASK Y
LABATT'S
TOM PETTY
SANDBOX
U.S.G.S.
MR. BUBBLE
THREESOME
GANG
IN TOUCH
MORE...

- Repetitive fields
 - Why ? Because then we could use arrays in HLL's.
 - Requires application code to parse and assemble.
 - Has hard limits, else major root canal to increase limits
- Unused space
 - Why ? Because that's the way it was always done.
 - Traditional filler areas, obsolete fields
 - Empty instances of repetitive fields
 - Descriptors sized to accommodate the oddballs
- Truly unique and productive fields
 - Why ? Because the file was created for a reason.
 - Differentiates the file from the rest of the application.
 - Mission critical to the business model.







What does this mean (in English) ?.....


RUNDOWN
Y ASK Y
LABATT'S
TOM PETTY
SANDBOX
U.S.G.S.
MR. BUBBLE
THREESOME
GANG
IN TOUCH
MORE...

Chances are your home grown database will not support a significant seismic event:

- New business units
 - Different fiscal calendars
 - Apps often assume calendar year
 - Different security guidelines
 - Scripts with hardcoded user ID's. Yikes !
 - Different specialty data
- Consolidation or elimination of business units
- New product lines
- New enterprise level customers
- Shift in strategic vision and mission
- A change in platform. Yikes !







RUNDOWN

Y ASK Y

LABATT'S

TOM PETTY

SANDBOX

U.S.G.S.

MR. BUBBLE

THREESOME


GANG

IN TOUCH

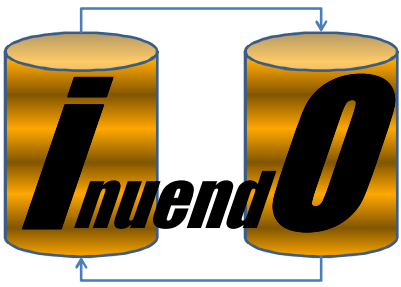
MORE...

Then what happens ?.....

- Baby, bath water – you do the math.





- Or major patching and add-on.
 - Only adds to the problem.
- That's the "why". Next the "how".





...the art of the sexy database


Prism model of supply chain interaction











Supply chain basics

RUNDOWN

Y ASK Y

LABATT'S

TOM PETTY

SANDBOX

U.S.G.S.


MR. BUBBLE

THREESOME


GANG.


IN TOUCH


MORE...



Your business is an ecosystem of relationships, all of which interact on a day to day basis.







Supply chain basics

RUNDOWN

Y ASK Y

LABATT'S

TOM PETTY

SANDBOX

U.S.G.S.

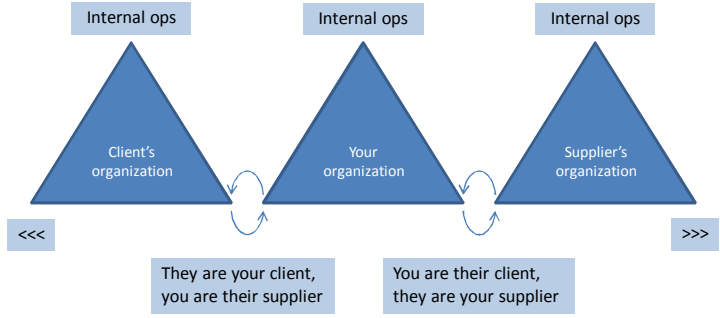
MR. BUBBLE

THREESOME


GANG


IN TOUCH


MORE...



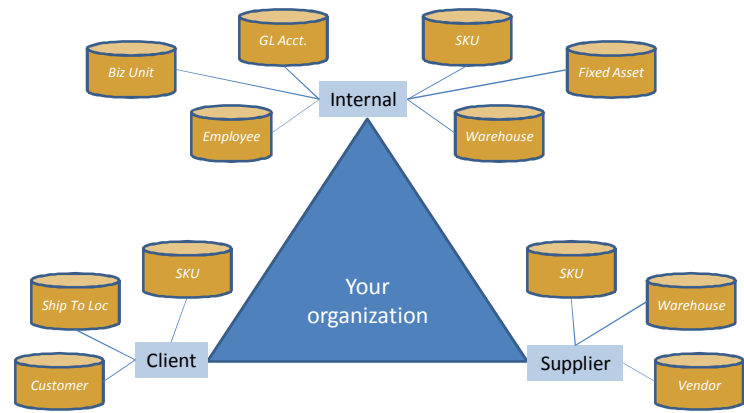
In a nutshell, the supply chain is simply the limitless cascade of these linked ecosystems.










RUNDOWN
 Y ASK Y
 LABATT'S
 TOM PETTY
 SANDBOX
 U.S.G.S.
 MR. BUBBLE
 THRESOME
GANG
 IN TOUCH
 MORE...



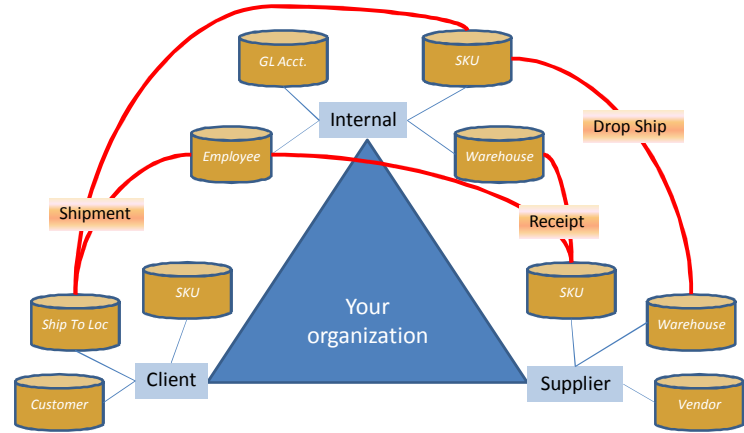
Each point on the prism is supported by a series of business objects which we represent with data.










RUNDOWN
 Y ASK Y
 LABATT'S
 TOM PETTY
 SANDBOX
 U.S.G.S.
 MR. BUBBLE
 THRESOME
 GANG
IN TOUCH
 MORE...



Everything we do is merely a conversation between two or more of these objects. For example...







Simple enough.....

RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS

COLONEL FLAG

McFLY


EAT FRESH


SWIFFER


MORE...

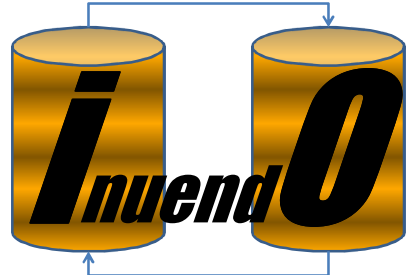
But where did it go wrong ?

- Most home grown systems were assembled little by little, by multiple individuals using multiple approaches over multiple years.
- Design decisions based on the concept of IT being purely cost, not an investment.
- Useful life of the application underestimated.
- “If it ain’t broke, don’t fix it” is NOT a strategic mission statement, but was treated like one.
- A less constrained approach was needed.









RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS

COLONEL FLAG


McFLY


EAT FRESH


SWIFFER

MORE...

- First concepts developed in 2007 during insurance application modernization.
- Full model white boarded in 2009.
- Multiple implementations in 2010, additional ones planned.










What is meant by Inuendo ?.....

RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...

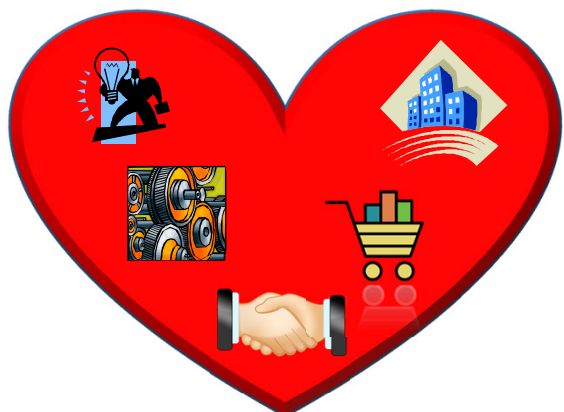
- An Italian suppository ? Not quite.
- Rather, it is a database model designed to treat all components of the supply chain equally, regardless of their role.
- Not platform or DB engine specific.
- Provides unlimited growth and flexibility.
- Provides comprehensive audit trail.
- Lessens dependency on IT personnel for ordinary requests.










At the heart of Inuendo.....






...are its principles

Inuendo principles.....

RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...


- Objects and transactions alike are individual, independent entities.
- Each entity has a database generated enterprise wide identity and a well known (to the user) legacy identifier (example – customer number).
- Each entity has a class (what type of entity).
- Each class has a distinct set of properties, each with its own nickname an associated data type.
- Data is organized vertically by data type and accessed in an associative manner by nickname.
- Data is time, user and program stamped.
- I/O is performed using centralized functions.







Inuendo principles (cont.).....

RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...

- Any of the following could be considered an entity:
 - Direct points on the prism
 - Business objects supporting the direct points
 - Permanent relationships between business objects
 - Temporarily relationships between business objects
 - Increments of cumulative statistics
- Many business objects and transactions have common properties. For example:
 - One or more addresses, phone numbers, etc.
 - One or more dates, flags, etc.
 - One or more comments, monetary values, etc.
 - One or more parent objects from which it was spawned.
- Just because an object doesn't have a particular property today, doesn't mean it never will.
- Should have an uncapped number of instances of each property as well as an uncapped number of properties.







Inuendo principles (cont.).....

RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS

COLONEL FLAG


McFLY


EAT FRESH


SWIFFER

MORE...

- Data should be retained indefinitely.
- A piece of information associated with an entity should exist once and only once in the database.
 - Includes the legacy “well known” identifier.
- No aggregate fields should exist, only the detail that supports them.
 - Why introduce the risk of disjointedness ?
- Should be able to recapture the value of any property at any moment in time.
 - More comprehensive than journaling
- Should leverage the power of the database engine first before creating any application code.







The heart of the Inuendo data model

RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS

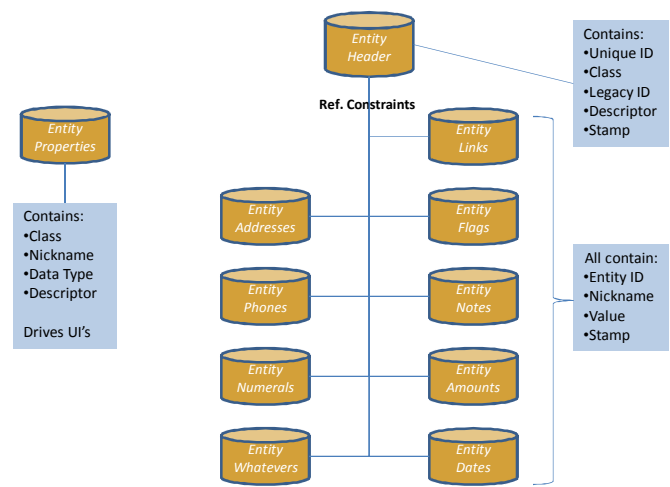
COLONEL FLAG

McFLY


EAT FRESH


SWIFFER


MORE...



The diagram illustrates the Inuendo data model structure. At the top is the **Entity Header**, which contains: Unique ID, Class, Legacy ID, Descriptor, and Stamp. Below it is the **Ref. Constraints** section, which includes tables for Entity Links, Entity Flags, Entity Notes, Entity Amounts, and Entity Dates. To the left of the Ref. Constraints are tables for Entity Addresses, Entity Phones, Entity Numerals, and Entity Whatevers. On the far left is the **Entity Properties** table, which contains: Class, Nickname, Data Type, and Descriptor, and is noted as 'Drives UI's'. A note on the right states that all these tables contain: Entity ID, Nickname, Value, and Stamp.







Entity example.....

RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS

COLONEL FLAG

McFLY

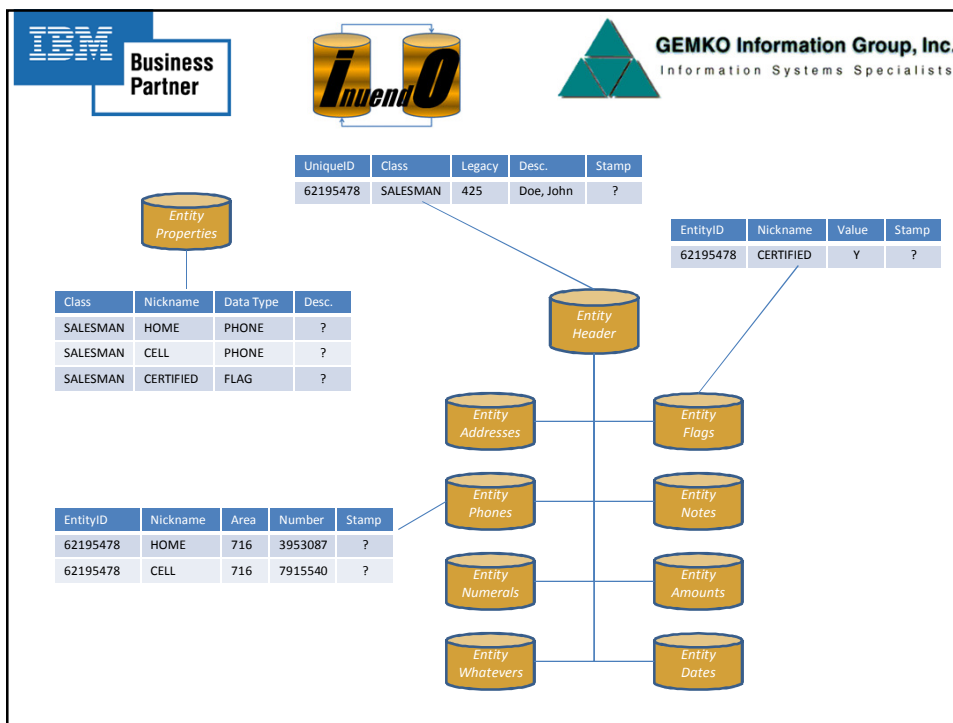
EAT FRESH




SWIFFER

MORE...

Consider a Salesman entity. Aside from other properties, every Salesman has:

- A legacy salesman number.
 - Would apply to almost any entity class.
 - Probably just looks different, depending on class.
- A full name / description.
 - Would apply to almost any entity class.
- A home phone and a cell phone number.
 - Would apply to many entity classes.
- A flag indicating whether he is certified.
 - Probably only applies to salesman class.
- The date he was added to the system.
 - Would apply to almost any entity class.






Database engine components....

RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...


- The only identity column is in the Entity Header.
- Referential constraints link the EntityID field in each sub table to the UniqueID in the Header.
- The Value field in each sub table may be based on a DISTINCT TYPE defined through SQL based on the needs of the business.
- During transition, triggers on legacy physical files replicate effects of I/O operations real time.







Application components....

RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...

- Standard I/O functions to get, put property values.
 - One per data type (better than Java getters/setters).
- If a “data type” is complex (example – address), then standard functions to “cast” to it.
 - In HLL’s, might be done with data structures.
 - Isolate all mashing/finagling/scrubbing in one place.
- Function to create a new entity header.
 - Or “instantiate a new entity” for you Java buffs.
- Function to reconcile legacy ID to entity ID based on entity class.
 - In earlier models, created one per class ☹.
 - Proper uniqueness is important on legacy side.







Standard I/O functions example:

RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS


COLONEL FLAG

McFLY

EAT FRESH

SWIFFER

MORE...




EntityID	Nickname	Value	Stamp (3+ columns)
62195478	CERTIFIED	N	2009-06-14-08.24.27.680583 JDOE SLS465R1
62195478	CERTIFIED	Y	2010-04-21-15.58.05.149804 CSMITH SLS465R1


Function `getFlag(EntityID, NickName)`


- Uses SQL to retrieve the most recent row for the EntityID and Nickname combination from the Entity Flags table, and returns the Value.
- Example: `getFlag(62195478, 'CERTIFIED')`
- This would imply the value "Y" because it is the most recent.

Function `putFlag(EntityID, NickName, NewValue)`

- Uses `getFlag` to determine the current value, and if it changes, writes a stamped record to the Entity Flags table.
- Example: `putFlag(62195478, 'CERTIFIED', 'Y')`
- Could imply some type of successful/failure indication as a result value.







ID reconciliation example:

RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS


COLONEL FLAG

McFLY

EAT FRESH

SWIFFER

MORE...





UniqueID	Class	Legacy	Desc.	Creation Stamp (3+ columns)
62195478	SALESMAN	425	Doe, John	2009-05-22-09.32.17.401031 FRED SLS465R1
65800351	SALESMAN	189	Doe, Jane	2009-11-05-12.55.20.795306 HARV SLS465R1


Function `getEntityID(Class, LegacyID)`

- Uses SQL to retrieve the unique ID of the entity for the matching Class and LegacyID from the Entity Header and returns the database generated value. A value of zero indicates a "not found" or "invalid seek" condition.
- Example: `SalesmanID = getEntityID('SALESMAN', 189)`

- Handy anywhere the user might key in a legacy identifier, for verification and definitive linkage with other database tables. Remember, the only place the legacy ID will EVER exist is in the Entity Header. This is a BIG yet powerful deviation from many traditional methods.
- Suppose your company decided to change the (legacy) salesman number scheme. No biggie if the salesman number is only in one place. BIG biggie if the salesman number is in 200 places. Get the picture ?



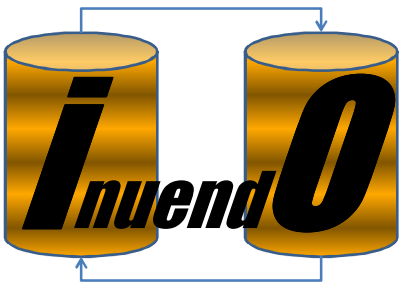




Isn't that a lot of data ?.....


RUNDOWN	
DYSFUNCTION	
INUENDO	
CANNED TUNA	
SALESMAN	
TOYS	
COLONEL FLAG	
McFLY	
EAT FRESH	
SWIFFER	
MORE...	


- Not as much as you'd think !
- Most highly volatile DB fields tend to be aggregates, which are prohibited by Inuendo.
- Descriptive and strategic data tends to be static for long periods of time.
- Sub table rows are generally narrow.
- Indexes are simple and uniform
- Apps will only retrieve the info they need.
- Hence, overhead is light.
- But wait until you see what's next...






...the art of the sexy database

Time travel technology












Time travel technology....

RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...

There are probably things we'd like to do in the past:

- Go back and put big money on Butler to make the 2010 NCAA finals.
- Unsend that ill-advised e-mail from 5 months ago.
- Or perhaps, more realistically, see what an agitated end user really saw on their screen when they told you – “it was broke, but now it works, but fix it anyway”.
 - That one we can help you with.

Time travel technology.....

RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...


Let's reconsider a couple of basic Inuendo principles:


- All entities are created equal and have a creation time stamp (and user stamp, and perhaps others).
- All properties of all entities are stored by data type in sub tables, with each change in a property's value indexed with a time stamp (plus other stamps).


Now reconsider the last two examples:

- Retrieving a property value for a given entity.
- Retrieving the unique ID for an entity .

Let's make one tweak to each function...







Standard I/O functions example:

RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS


COLONEL FLAG

McFLY

EAT FRESH

SWIFFER

MORE...




EntityID	Nickname	Value	Stamp (3+ columns)
62195478	CERTIFIED	N	2009-06-14-08.24.27.680583 JDOE SLS465R1
62195478	CERTIFIED	Y	2010-04-21-15.58.05.149804 CSMITH SLS465R1


Function `getFlag(EntityID, NickName, TimeIndex)`


- TimeIndex is an OPTIONAL argument of type TIMESTAMP.
- Uses SQL to retrieve the most recent row for the EntityID and Nickname combination from the Entity Flags table, whose recorded time stamp is LESS THAN the one provided as an argument, and returns the Value.

•Example: `getFlag(62195478, 'CERTIFIED', '2009-06-19-11.25.56.228509')`
 •This would imply the value "N", because the first row precedes the time index argument.

•Example: `getFlag(62195478, 'CERTIFIED', '2009-04-28-15.02.44.311606')`
 •This would imply a blank value (or null if you so desire), because there are no entries which proceed the time index argument.







ID reconciliation example:

RUNDOWN

DYSFUNCTION

INUENDO

CANNED TUNA

SALESMAN

TOYS


COLONEL FLAG

McFLY

EAT FRESH

SWIFFER

MORE...




UniqueID	Class	Legacy	Desc.	Creation Stamp (3+ columns)
62195478	SALESMAN	425	Doe, John	2009-05-22-09.32.17.401031 FRED SLS465R1
65800351	SALESMAN	189	Doe, Jane	2009-11-05-12.55.20.795306 HARV SLS465R1


Function `getEntityID(Class, LegacyID, TimeIndex)`


- TimeIndex is an OPTIONAL argument of type TIMESTAMP.
- Uses SQL to retrieve the unique ID of the entity for the matching Class and LegacyID from the Entity Header, if its recorded time stamp is LESS THAN the one provided as an argument, and returns the database generated value. A value of zero indicates a "not found" or "invalid seek" condition.

•Example: `getEntityID('SALESMAN', 189, '2009-06-19-11.25.56.228509')`
 •Would imply a zero value because Jane did not exist at that time index.

•Example: `getEntityID('SALESMAN', 425, '2009-06-19-11.25.56.228509')`
 •Would imply 62195478 because John existed at that time index.





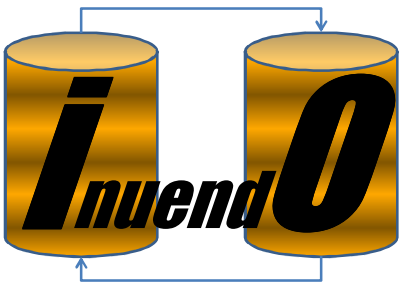


What could we do with this ?.....

RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...


By changing our system time or session time (depending on your operating system):


- See an inquiry screen the way a user would have seen it back on February 28th, before month end.
- Recognize only transactions as of a given date.
- Generate a missing year end report for the auditors.
- Recreate an order acknowledgment for a customer when your application no longer lets you do it because the order has since been shipped and invoiced.
- Help a user figure out where they took a wrong turn during last week's A/P check run.
- Getting the picture now ?





...the art of the sexy database


Implementing as part of a modernization strategy
















The Modernization Meal.....


RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...

- A repeatable, best practice and educational solution.
- Low risk, low disruption approach that leverages the resources you already have, and at your own pace.
- Delivering this message since the 2005 Fall COMMON.
- Technology changes, but the mission is still the same.
- Presentation available to your group live or on webcast.

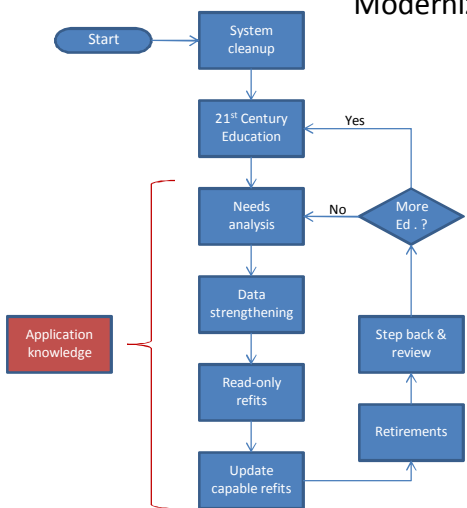









Modernization meal process


RUNDOWN
DYSFUNCTION
INUENDO
CANNED TUNA
SALESMAN
TOYS
COLONEL FLAG
McFLY
EAT FRESH
SWIFFER
MORE...





```

graph TD
    Start([Start]) --> SC[System cleanup]
    SC --> CE[21st Century Education]
    CE --> NA[Needs analysis]
    NA --> DS[Data strengthening]
    DS --> ROR[Read-only refits]
    ROR --> UCR[Update capable refits]
    UCR --> R[Retirements]
    R --> SBR[Step back & review]
    SBR --> ME{More Ed. ?}
    ME -- No --> NA
    ME -- Yes --> CE
    
```



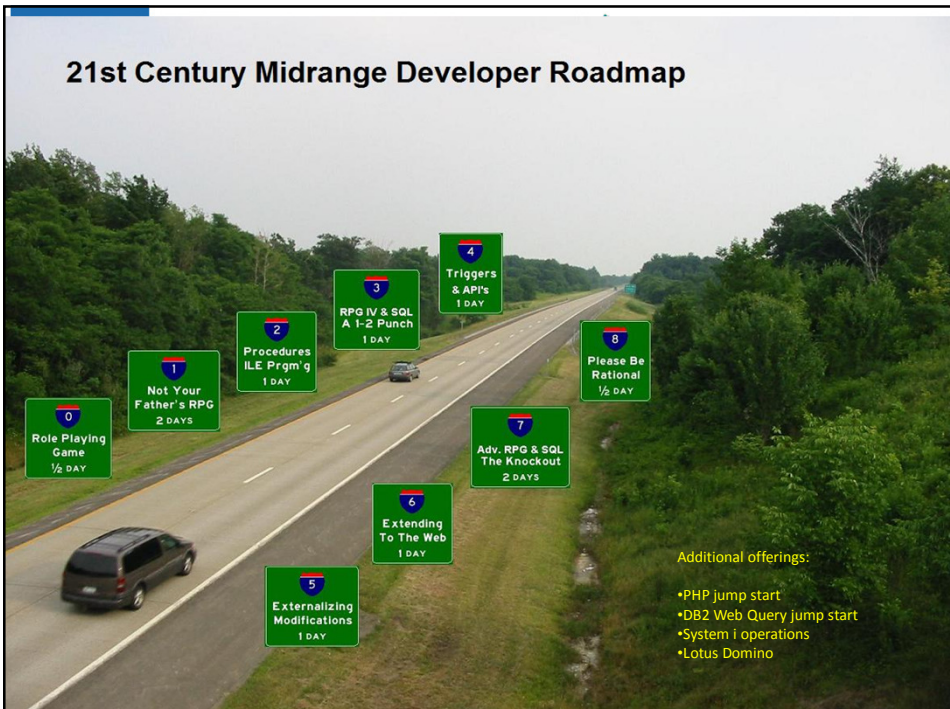




System cleanup....

RUNDOWN	
DYSFUNCTION	<ul style="list-style-type: none"> • Deploy native, home grown analysis tools. <ul style="list-style-type: none"> – Open source in nature. • Identify unused or unreferenced objects. <ul style="list-style-type: none"> – Retire to designated archive. – Reduces scope of refit steps • Cleanse environmental bad practice. <ul style="list-style-type: none"> – Example: user, library/schema hard coding. • Establish proper test environment. <ul style="list-style-type: none"> – Including regularly scheduled refreshes.
INUENDO	
CANNED TUNA	
SALESMAN	
TOYS	
COLONEL FLAG	
McFLY	
EAT FRESH	
SWIFFER	
MORE...	


21st Century Midrange Developer Roadmap





- 0 Role Playing Game ½ DAY
- 1 Not Your Father's RPG 2 DAYS
- 2 Procedures ILE Prgm'g 1 DAY
- 3 RPG IV & SQL A 1-2 Punch 1 DAY
- 4 Triggers & APIs 1 DAY
- 5 Externalizing Modifications 1 DAY
- 6 Extending To The Web 1 DAY
- 7 Adv. RPG & SQL The Knockout 2 DAYS
- 8 Please Be Rational ½ DAY

Additional offerings:

- PHP jump start
- DB2 Web Query jump start
- System i operations
- Lotus Domino










21st Century education.....

RUNDOWN	
US-219	<ul style="list-style-type: none"> • Series of hands on, instructor led workshops. <ul style="list-style-type: none"> – Live, customer site or public facility. • Covers skills required during later steps. <ul style="list-style-type: none"> – Many of which are cross platform. • Many of the skills are cross platform. <ul style="list-style-type: none"> – Example: SQL, DB concepts, Eclipse workbench • Consistent with IBM certification standards. <ul style="list-style-type: none"> – Instructor was co-author of latest exams. • “You don’t know what you don’t know”. <ul style="list-style-type: none"> – Aging applications tend to promote aging skills.
ANDRO	
RETRO	
EXPOSE	
Y AGAIN	
S & M	
THANKS	
BOTTOM	

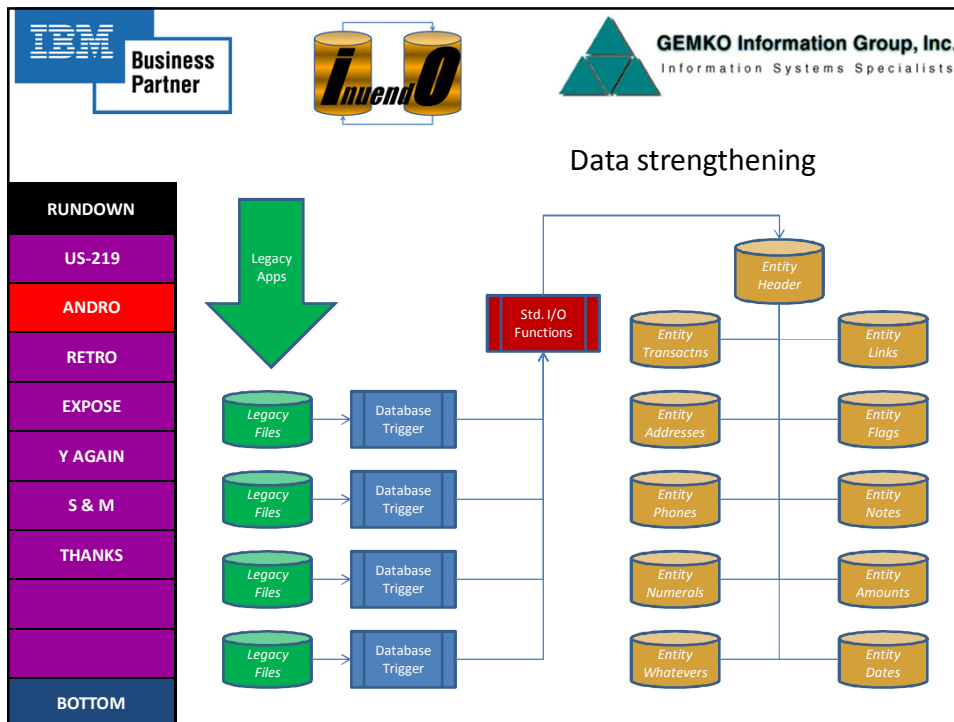






Data strengthening....

RUNDOWN	
US-219	<ul style="list-style-type: none"> • Establish core Inuendo tables. <ul style="list-style-type: none"> – Includes referential integrity rules. – First cycle heavy, subsequent cycles lighter. • Construct standard I/O functions. <ul style="list-style-type: none"> – Includes unique ID resolution. – First cycle heavier, subsequent cycles lighter. • Turn on real-time replication to Inuendo tables. <ul style="list-style-type: none"> – Legacy user experience unaffected. – Verification of data. • Publish technical playbook for staff to use during refits.
ANDRO	
RETRO	
EXPOSE	
Y AGAIN	
S & M	
THANKS	
BOTTOM	

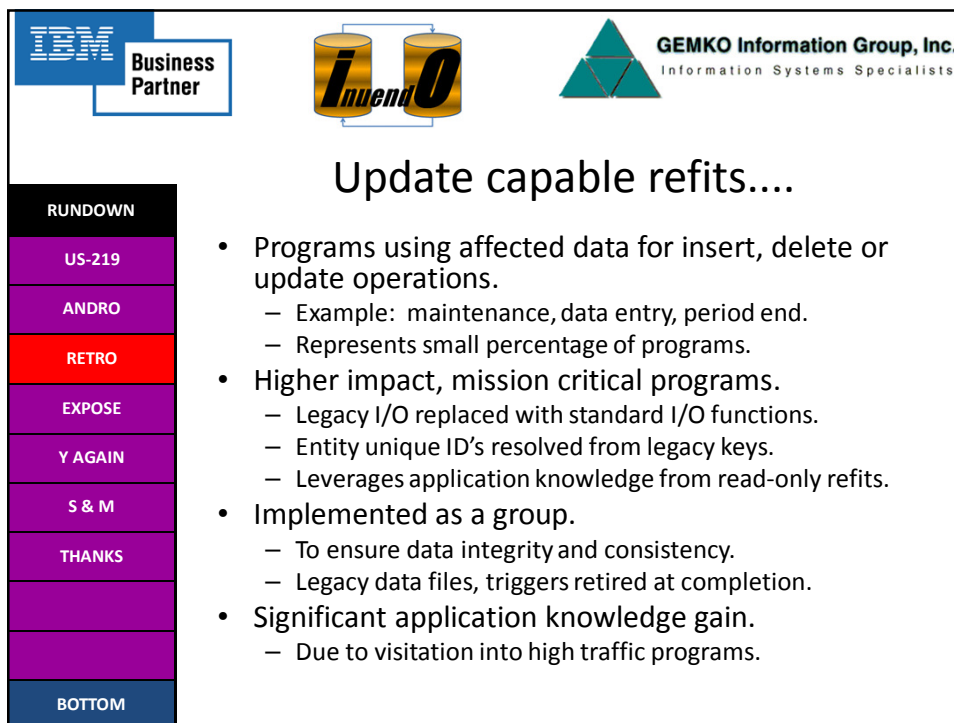
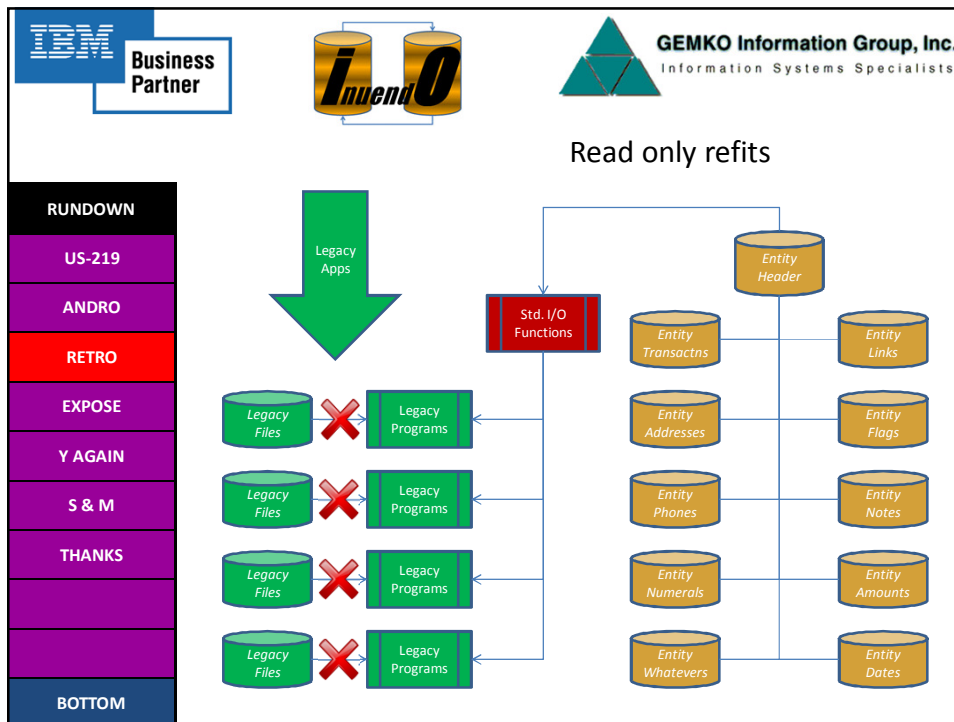


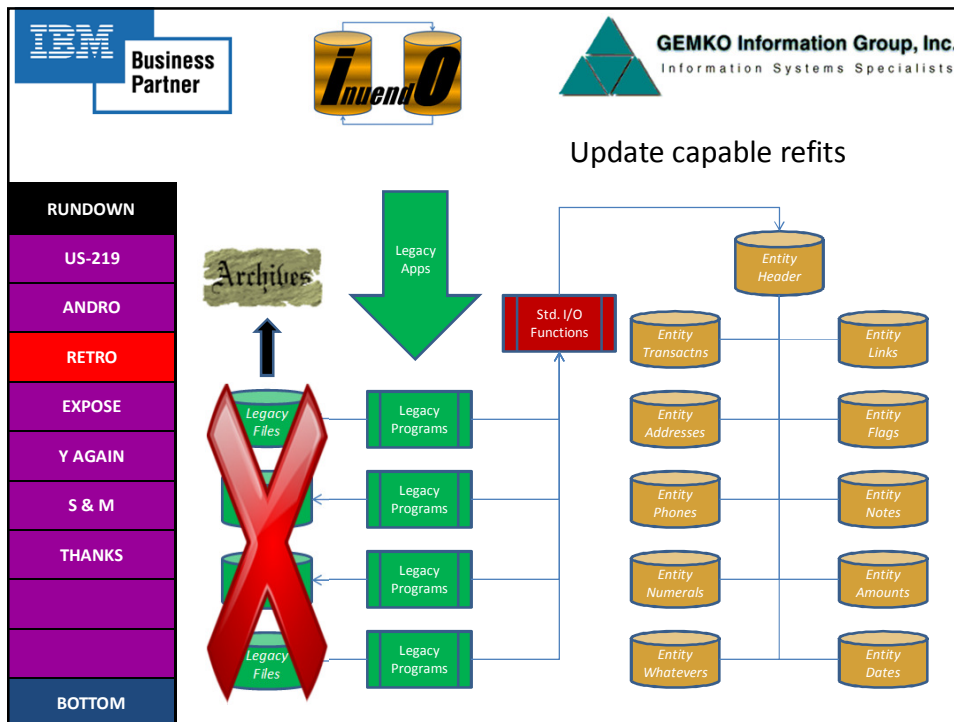
IBM Business Partner | **Invent 0** | **GEMKO Information Group, Inc. Information Systems Specialists**

Read only refits....

- Programs using strengthened data for input operations only.
 - Example: inquiries, reports, etc.
 - Represents largest percentage of programs.
- Low risk, low impact, low disruption
 - Legacy I/O replaced with standard I/O functions.
 - Entity unique ID's resolved from legacy keys.
- Discrete “behind the scenes cycle”
 - Refit, test, implement.
 - One at a time or small groups, not a big bang.
- Significant application knowledge gain.
 - Due to visitation into program set.

RUNDOWN
US-219
ANDRO
RETRO
EXPOSE
Y AGAIN
S & M
THANKS
BOTTOM






IBM Business Partner **InuendO** **GEMKO Information Group, Inc.**
Information Systems Specialists


Step back and review....


RUNDOWN

US-219
ANDRO
RETRO
EXPOSE
Y AGAIN
S & M
THANKS
BOTTOM

- Some things went well, some perhaps did not.
 - Example: new skills inexperience, conventions, etc.
 - May demonstrate need for additional education.
- Fine tune estimated effort metrics.
 - Several one time tasks now behind us.
 - Better clarity on how long refits take on average.
- Procedural improvements.
 - Testing practices, user signoff, etc.
 - Update playbook guidelines but not crippling standards.
- May expose previously unrecognized constraints.
 - Could change priority of next cycle candidates.



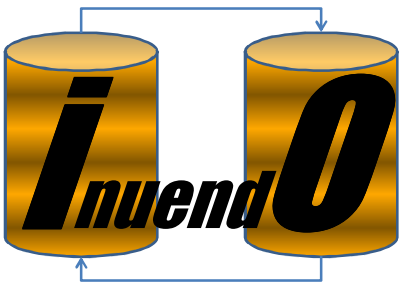




Then what ?....


RUNDOWN	
US-219	
ANDRO	
RETRO	
EXPOSE	
Y AGAIN	
S & M	
THANKS	
BOTTOM	


- Process repeats indefinitely.
 - New set of constraints, legacy data files.
 - Smarter each time through.
- Sporadic big bang efforts are disruptive, unpredictable and frequently deferred.
 - Often leads to stagnant applications.
- A POOGI can be forecasted and budgeted for on an annual basis.
 - Promotes more dynamic applications.
 - Perpetual improvement in more manageable chunks.
 - Investment in previous milestones is protected.
- Seismic events no longer as ominous.
 - Thanks to flexibility of applications and data.
 - Acquired knowledge.





...the art of the sexy database


Finally, why GEMKO ?














Why GEMKO ?.....

RUNDOWN	
US-219	
ANDRO	
RETRO	
EXPOSE	
Y AGAIN	
S & M	
THANKS	
BOTTOM	

- IBM business partner since inception in 1990.
- Rich portfolio of solutions and skills.
 - IBM System i platform.
 - Microsoft Dynamics platform.
 - Infor ERP platform.
 - LAMP technology development.
 - Annual customer care plans.
- Numerous alliances with product partners.
- Loyal customer base across many industries.
- Partly owned by Gaines, Kriner, Elliott LLC.
 - Accounting & financial services.
 - Business process improvement (TOC).
 - Marketing and relationship acquisition.
- Seek to earn our place as a Trusted Advisor for your organization.











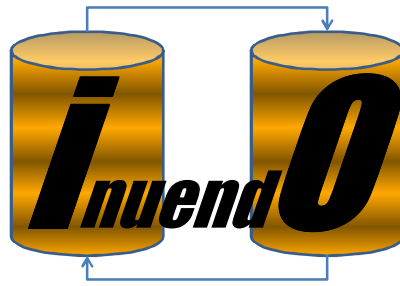
Solutions & More

(in other words, ask questions now)

RUNDOWN	
US-219	
ANDRO	
RETRO	
EXPOSE	
Y AGAIN	
S & M	
THANKS	
BOTTOM	



...the art of the sexy database

Thank you for your attention

Christopher F. Burns, Sr.

cburns@gemko.com



GEMKO Information Group, Inc.
Information Systems Specialists