

Clariteq

Data Modeling: New Uses for New Times

DAMA Minneapolis – May 21, 2003

Presented by:

Alec Sharp

Clariteq Systems Consulting Ltd.

206 – 2438 Marine Drive

West Vancouver, BC

Canada V7V 1L2

604 925-2440

asharp@clariteq.com

www.clariteq.com

To start the discussion...

Data modeling is largely unchanged over 15-20 years...

✓ Do you agree?

... but the environment is very different.

✓ In obvious and not-so-obvious ways

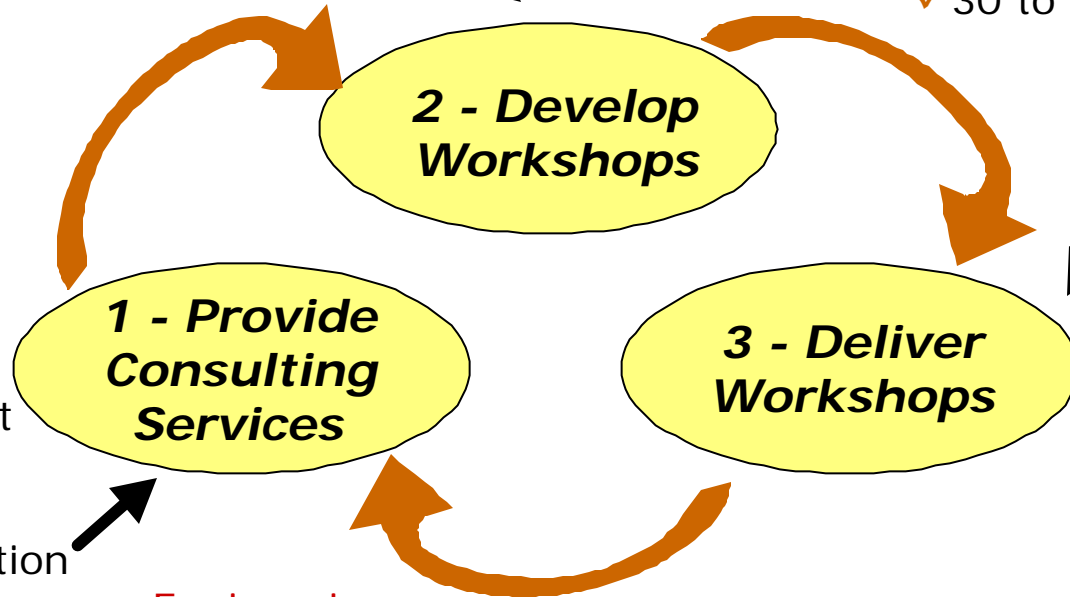
1 -What's different?
2 - How have you used data modeling in a new way?

✓ Have you used data modeling in a way you wouldn't have predicted 10 or 15 years ago?

What I do...

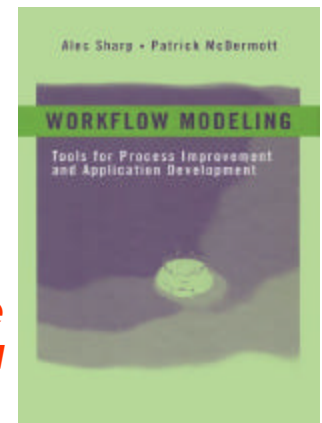
- ✓ Requirements Analysis Overview
- ✓ Workflow Process Modeling
- ✓ Use Cases and Application Logic
- ✓ Data Modeling
- ✓ Advanced Data Modeling

- ✓ In-house at large organizations
- ✓ Mostly 2 days each
- ✓ 30 to 40 per year



- ✓ System Development or Acquisition using Model-Driven Requirements Definition
- ✓ Data Modeling and Reverse-Engineering
- ✓ Process Mapping and Redesign
- ✓ Facilitation
- ✓ Business Planning & Data/Process/Application Architecture

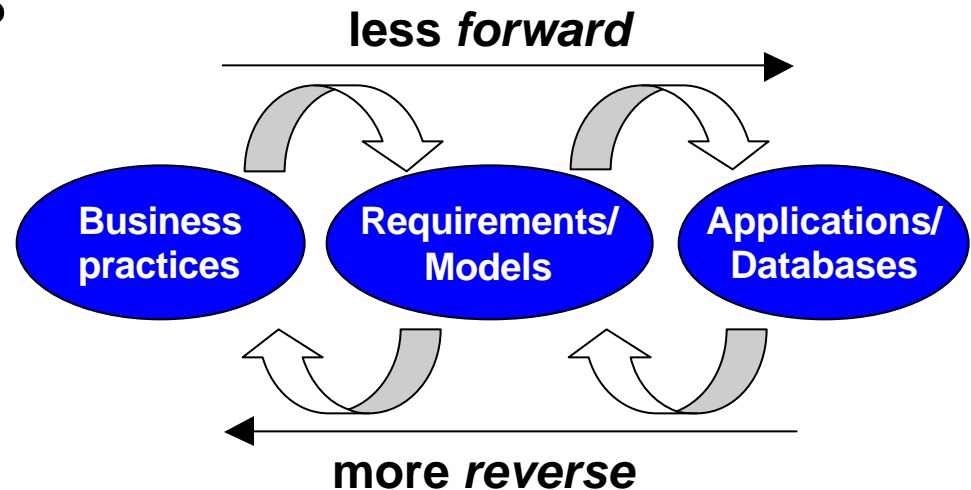
My mother is impressed because I wrote a book!



That's because everything's different!

- ◆ *Less “new” modeling, more dealing with existing structures*
 - purchased / ERP / COTS
 - custom
 - legacy

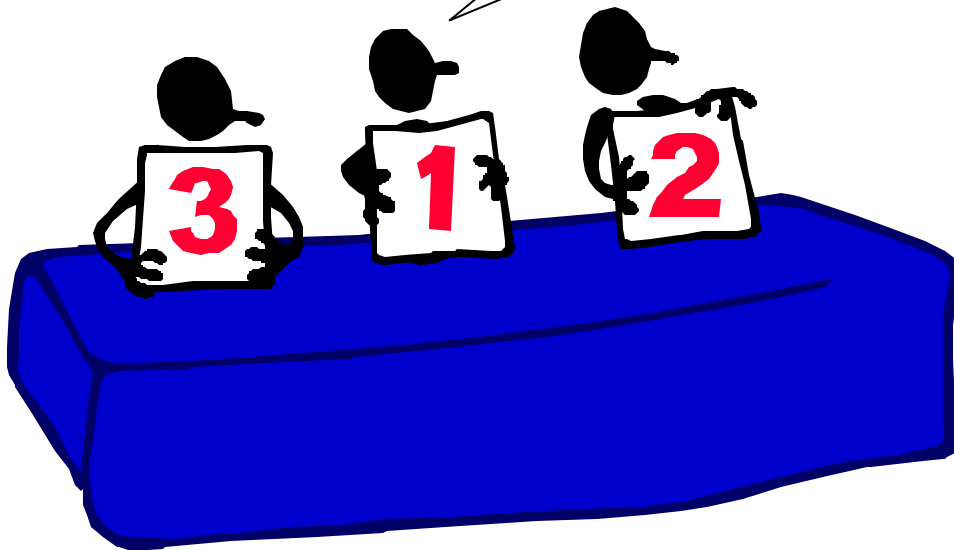
The new reality:



- ◆ *Less “gorilla” modeling, more “guerilla” modeling to support*
 - BPx
 - training
 - understanding policy & practice
 - ...?

Origins of the problem

“Data modelers won’t be needed anymore, because the software company has already done it!”



Various commentators on my career as a data modeler, mid-1990s



Coming to grips with it

- ◆ ~1991 - Max Hopper CW article
- ◆ early 90s –
“Modeler’s won’t be needed anymore!”
- ◆ mid 90s - Manufacturing client:
“Help! Our COO chose a package and...”
- ◆ Since then... “Help! We need to...”
 - select a package
 - adapt a package
 - implement a package
- ◆ Maybe Max was right...



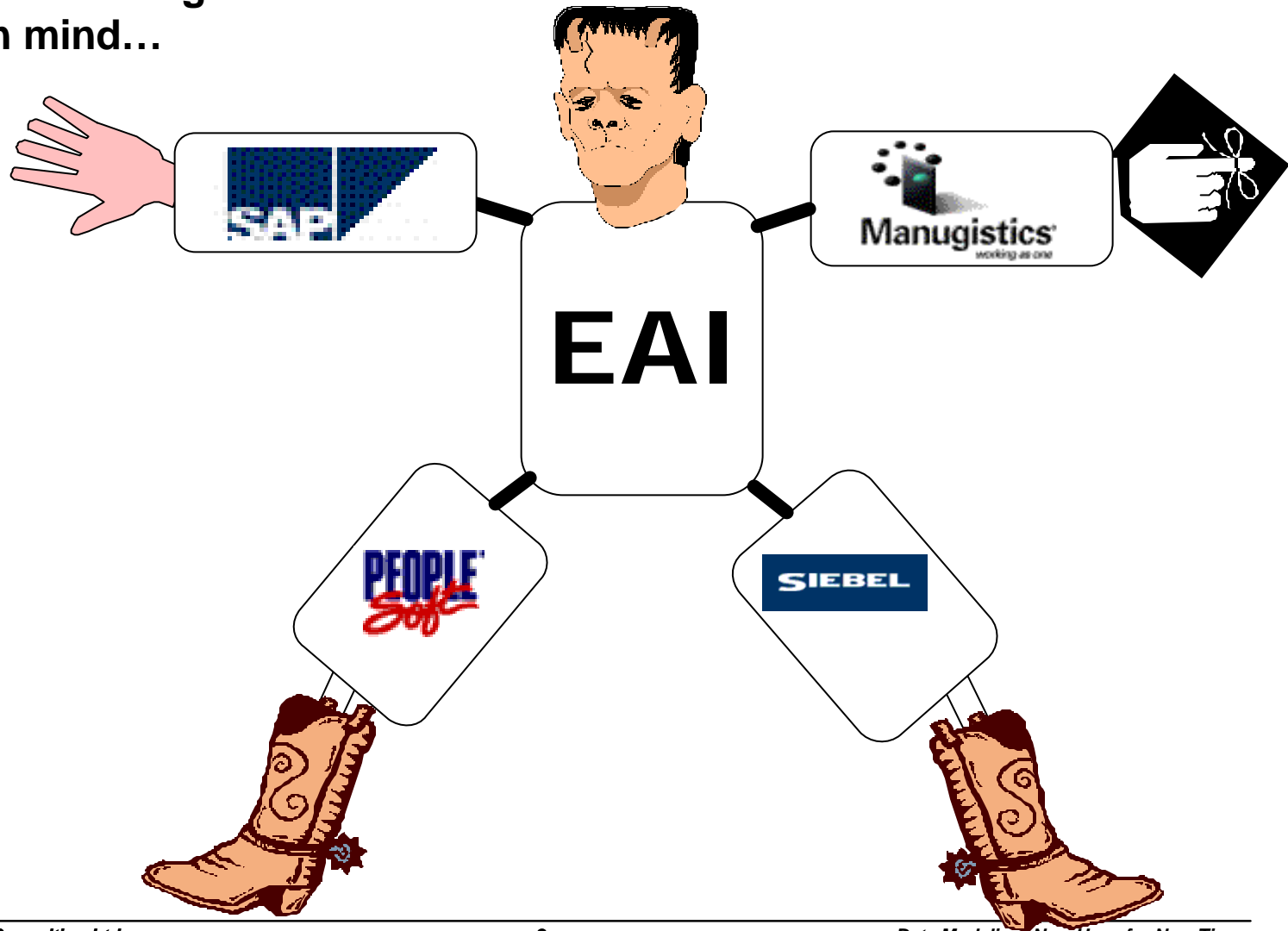
The state of the whirled

- ◆ The “E-whatever” hangover - another savior bites the dust and process – “BPx” - stages a comeback
- ◆ The Object-Oriented Analysis & Design hangover...
“It *should* have worked... we just needed smarter people”
- ◆ The Y2K hangover...
“All those legacy apps you wanted to replace...
they got a whole new lease on life!”
- ◆ The eternal quest for the perfect methodology...
SSAD, IE, RAD, RUP, and now...
lightweight methods!
- ◆ Major internal development is on hold,
and purchased applications have taken over
- ◆ Integrated architectures are suspect – “M&A means our
business regularly disintegrates and reintegrates”
- ◆ Thankfully – *XML and (fill in the blank) will save the day!*
(with assistance from the
Mary Shelley Consulting Corporation)



The Mary Shelley reference explained

Not quite the “integration”
we had in mind...



So what we have is...

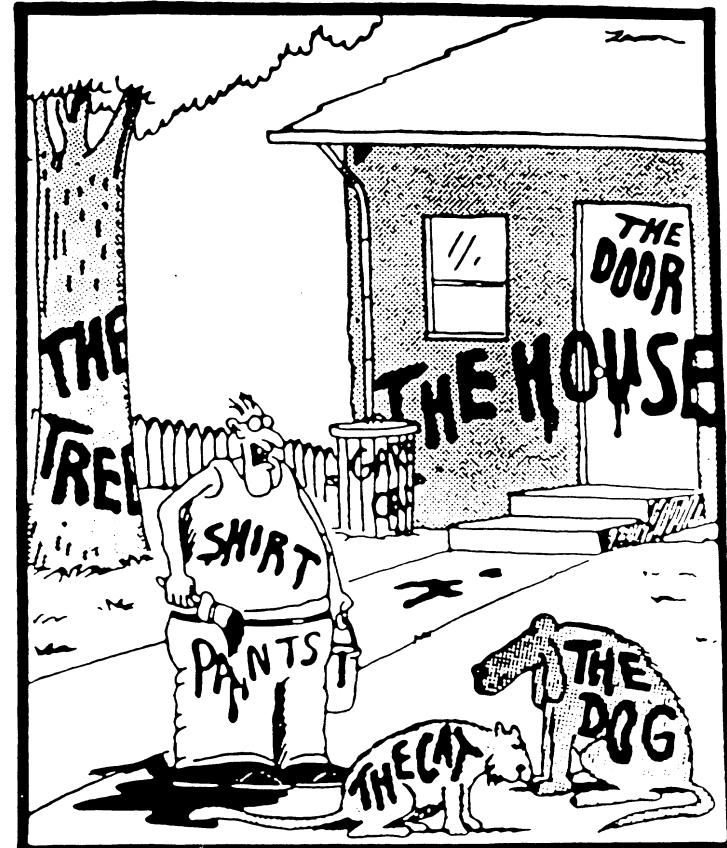
- ◆ **Bad integration, which has kindly been renamed “loosely coupled”**
- ◆ **Ad hoc planning (business as usual)**
- ◆ **A struggling internal I.T. (Alert the media!)**
- ◆ **Vendors and other providers in ascendancy**
- ◆ **A bunch of legacy, custom, third party, purchased, and ERP applications**
- ◆ **A few tools, no integration, and dubious prospects**

At least this is familiar territory!

So, what can we grab hold of?

- ◆ The business needs a *lingua franca* more than ever

- ◆ “Data” represents the things that
 - processes act on
 - applications revolve around
 - businesses want information about



“Now! *That* should clear up a few things around here!”

The basic message...

Data modeling is largely unchanged over 15-20 years...

- ✓ **Well defined** components and standards
- ✓ **Industry support** - tools, training, literature
- ✓ Widely seen as *the fundamental technique* (of course, "completely new" methods like OOA and UML come along from time to time, but...)

... but the environment is very different.

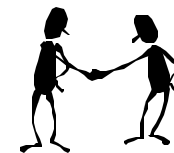
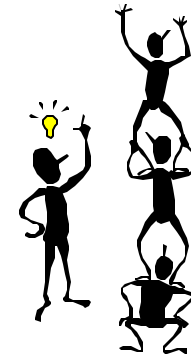
- ✓ **ERP, packages, COTS, ...**
- ✓ **Immortal legacy apps**
- ✓ **Less development, more plastic surgery**
- ✓ **A new take on "integration"**
- ✓ **New "partners" to deal with**

Adapt, or be a bystander. New times call for new approaches.

- ✓ **Show the business what they have**
- ✓ **Show the business what they're getting**
- ✓ **Reverse-engineering**
- ✓ **Presentation and exposition**
- ✓ **Acceptance... packages, "integration," etc.**

Let's hear some stories

- ◆ Preparing for configuration of ERP application
- ◆ Selection of G/L system
- ◆ Assessing the impact of implementing acquired software (simple example)
- ◆ Assessing the impact of implementing custom software (more complex example)
- ◆ “Why do we hate this application?” and “How should we influence vendor’s plans?”
- ◆ Determining if an application does what it should
- ◆ Securing support for development plans



#1

~1994 – redemption!

The client...

Could you come on over and do that thing you do?

That entity data stuff with the boxes and lines

We're implementing something called SAP. Our COO told us to!

They say it's a terrible idea and a waste of time.

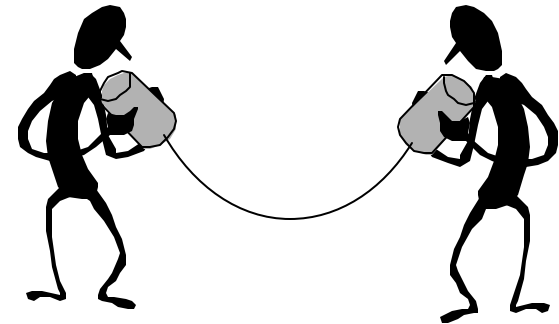
Alec...

I guess. What thing in particular?

Oh, data modeling. Sure - what's the project?

Uh-huh. What do your SAP consultants say about this?

I'm on my way.



#1

The details of my redemption

The problem:

- application selected by decree
- need to understand as-is business to map to package and decide on configuration options
- a list of 100s of requirements wasn't helping

The approach:

- team of 7 builds 45 entity conceptual data model over two days
- identify “what’s good, what’s not” about current business rules
- move on to configuration activities

- Vendor
- Country
- Plant
- Plant Location
- Equipment Item & Type
- PO, PO Line Item
- Req, Req Line Item
- Release, Release Line Item
- Work Definition, WD Item
- etc. etc. etc.

The key points:

- client-initiated, not IT
- “More value from those two days than anything else we did!”
- “I’m not irrelevant!”

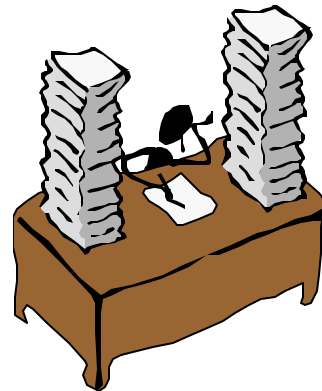
#2

Example 2: Selecting an application

Selecting of new Financials app is hopelessly bogged down

- Considerable effort in building a BDM*
- Two problems:
 - matrix points to the app no one likes
 - want vendor demos with focus and control

Requirements	D&B	Oracle	SAP	Coda	etc.
1	Y	Y	Y	Y	
2	Y	Y	Y	N	
3	Y	Y	Y	Y	
4	N	Y	N	Y	
5	N	N	Y	Y	
6	Y	Y	Y	Y	
7	Y	Y	Y	Y	
8	Y	Y	Y	Y	
9	Y	N	Y	N	
10	N	Y	N	Y	
11	Y	Y	Y	Y	
12	Y	Y	Y	Y	
13	Y	N	Y	Y	
14	Y	Y	N	N	
...					
...					
858	N	N	N	Y	
859	Y	Y	Y	Y	



* Big Dumb Matrix

BDM issues

- time consuming
- most apps meet most criteria
- still can't tell if an app will work well in your environment

#2

The dangers of over-analysis

CORRECT-POSTURE DOG FEEDER.



#2

Selection - focusing on what matters

The approach:

- small team builds “thing model” (~60 things total, 15 “core”)
- for each core thing (okay...entity) identify 3 to 5 life cycle events
- for each event, develop scenario
 - pre-conditions and test data
 - desired handling and outcome (“use cases”)
- turn over to app vendors – “Show us!”
 - how you support the model
 - how you handle scenarios

“Things we track” -

- Project, Work Order
- Plant, Plant Equipment
- Product Type, Product Lot
- Product Inventory
- Sale, Transfer
- Location, Ledger Entity
- Financial Category
- Responsibility Center
- Account, Sub-Account
- Fixed Asset

↓
“What happens to them” -

Fixed Asset is

- Acquired or Constructed
- Depreciated
- Transferred
- Disposed Of

The key points:

- it worked! - much clearer how well an app would support the business
- didn’t call it “data modeling” until late in the game
- left vendor some room - “Here’s how we’d do it.”

#2

A pattern you may have noticed...

A minimalist methodology for extending and validating a data model, and discovering further requirements.

1 – build a simple data model

2 - for each significant entity

- ✓ brainstorm events that target it, e.g.
Fixed Asset “is acquired”, “is depreciated”,
“is transferred”,
“is retired”
- ✓ arrange events in typical “life cycle” order
- ✓ add missing events

3 - for each event

- ✓ discuss “expected handling” -
 - communicating it to the system - use cases
 - validation, rules, data updates - application logic
- ✓ check to see if the data model handles it
- ✓ revise the data model as necessary

#3

What does this app. mean for us? (simple)

- ◆ **Situation**
 - **Blue Cross affiliate purchases “SuperBlue” from another (no study, no assessment – “divine intervention”)**
 - **VP of IS wants implementation impacts understood**
 - ◆ improvements & drawbacks
 - ◆ adjustments to business processes and procedures
 - **Initial plan – send 60+ business people through my data modeling class (\$\$\$!!!)**
- ◆ **Ethics prevents me (reluctantly!) from agreeing**
- ◆ **A better approach**
 - **Reverse-engineer something understandable**
 - **10 minute course on “data models” (unavoidable)**
 - ◆ what they are
 - ◆ how they impact systems and businesses
 - **Review SuperBlue model and capabilities with business folks**
 - **Let *them* identify impacts**

#3

How not to review a model



Models should aid understanding by:

- ✓ **Abstracting**
- ✓ Using **visual cues** consistently
- ✓ **Masking** unnecessary detail

#3

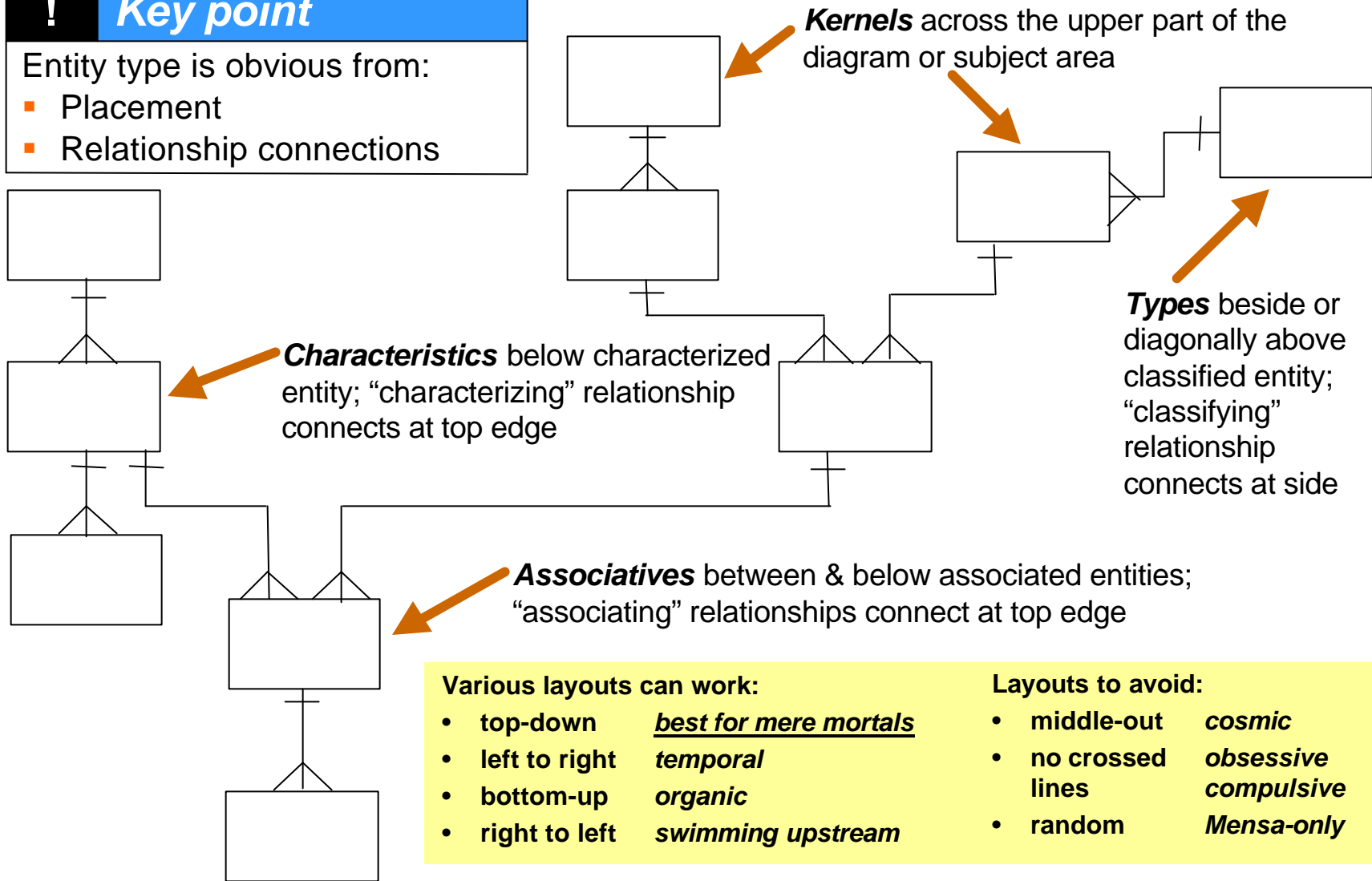
Drawing the model - "No Dead Crows!"



Key point

Entity type is obvious from:

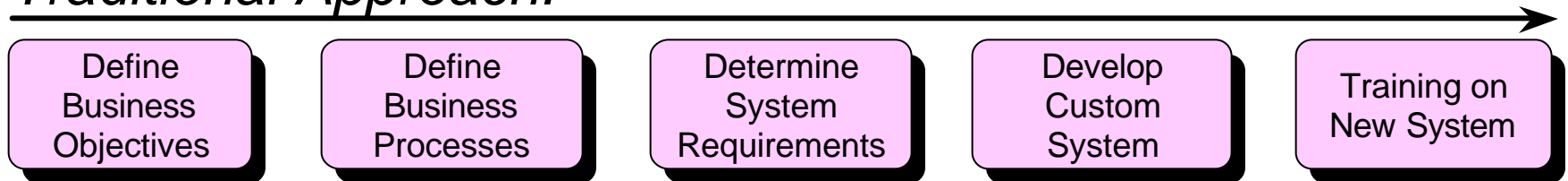
- Placement
- Relationship connections



#3

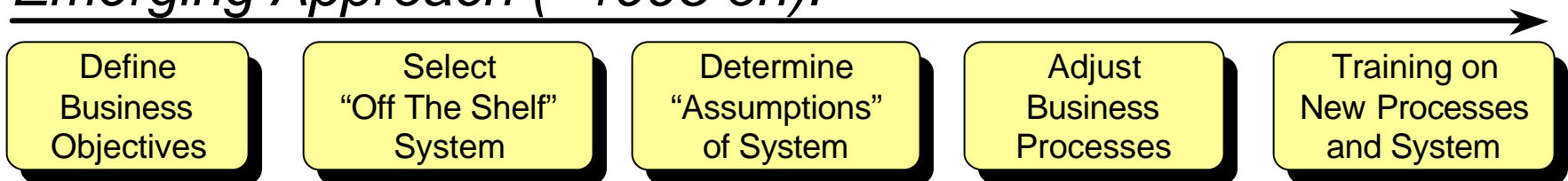
Some elements of the presentation...

Traditional Approach:



But systems have become very complex, expensive, and time-consuming to develop, so ...

Emerging Approach (~1995 on):



#3

The essence of Data Modeling...

- ◆ “A description of a business, in terms of the things it needs to know about.”

- ◆ There are many ways to describe a business...
 - How it works - Process Model
 - How it's organized - Organization Chart
 - Where it operates - Location Map
 - Whether it's making money - P&L Statement
 - What it needs information about – *Data Model*

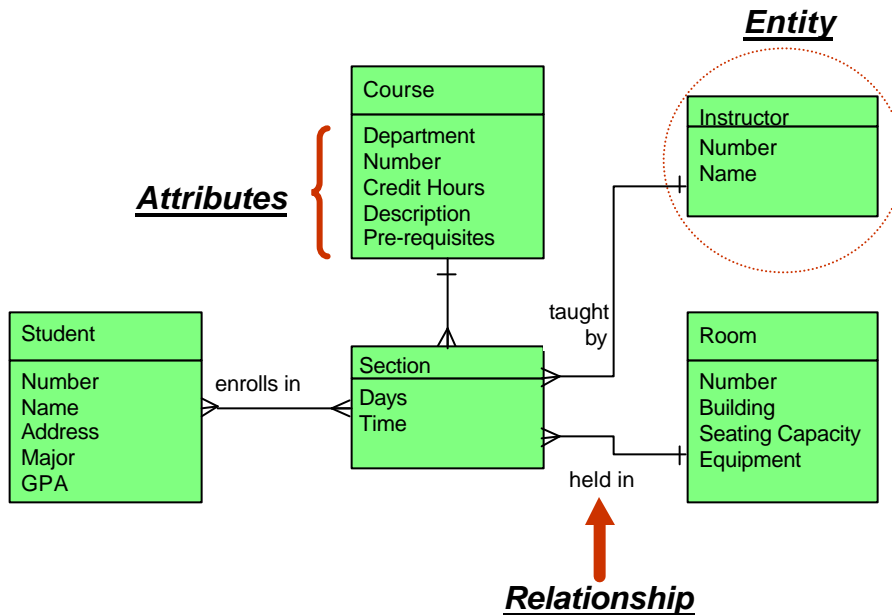
- ◆ Followed (reluctantly) by 10 minutes of data modeling education, and examples of how data models impact application capabilities

#3

What is a data model?

“ A depiction of the **things** the business needs to know about and the **facts about those things**”

Graphic Component



Entity-Relationship Diagram

Narrative Component

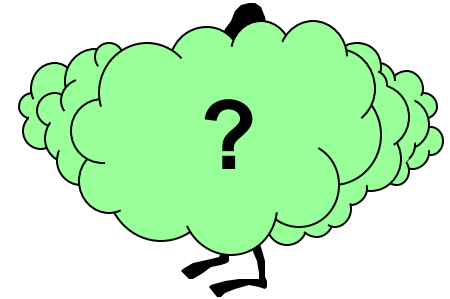
Student: A Student is any person who has been admitted to the University, has accepted, and has enrolled in a course within a designated time. Faculty and staff members may also be students

Entity Definition

#3

Why do we care about Data Models?

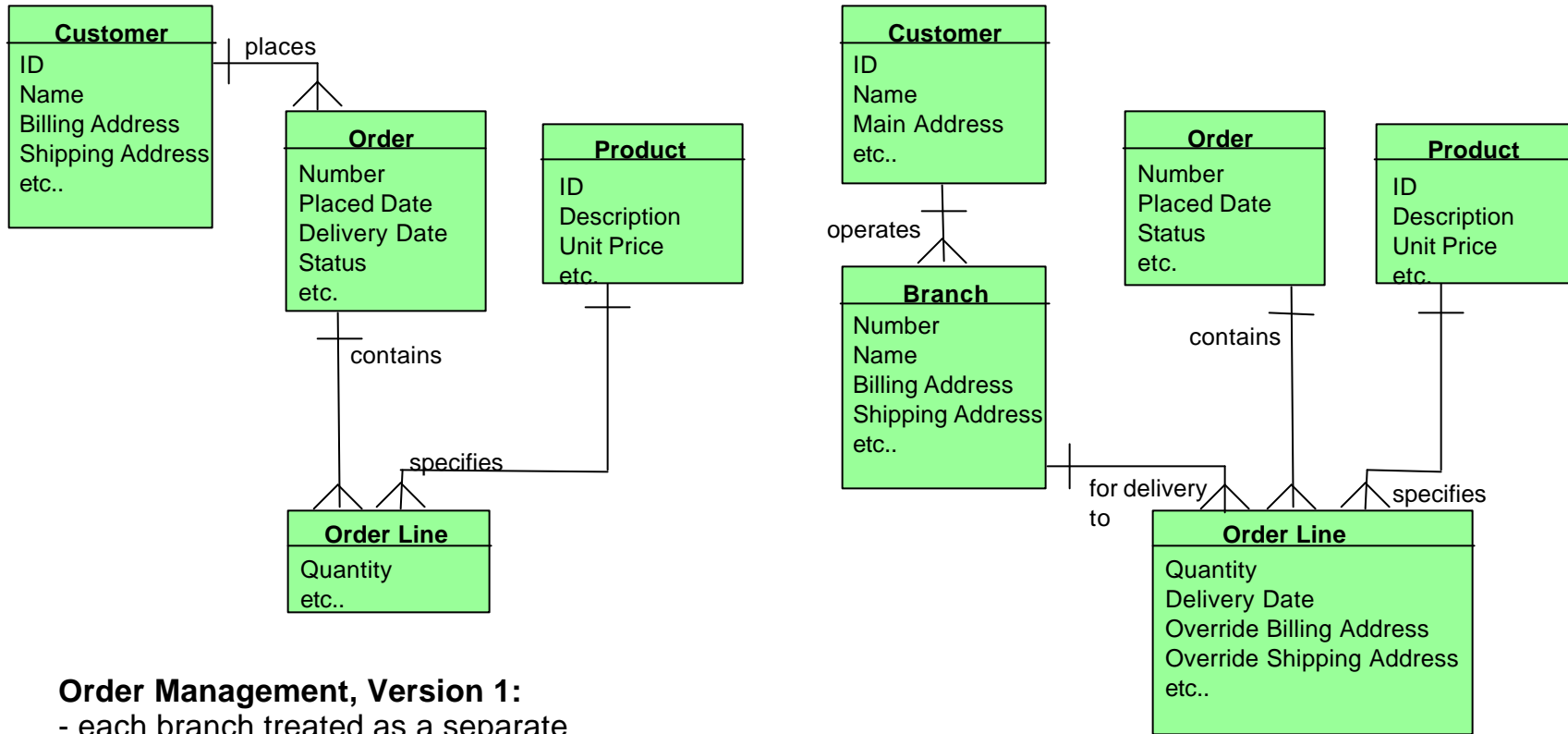
- ◆ The Data Model represents everything a system “knows” about the world
 - “world view” would be a better term
 - “things” - terminology and definition
 - “facts about things” – what data is maintained
- ◆ Fundamental impact on:
 - terminology
 - system capabilities
 - business processes
 - reporting capabilities
 - data that can be collected or displayed, and its format
 - *and almost everything else...*



#3

The critical point...

Different data models, different business capabilities!!!



Order Management, Version 1:

- each branch treated as a separate Customer - no easy way to "roll up"
- one delivery date and location per Order
- can't override Addresses for an Order

Order Management, Version 2:

- easy to "roll up" business across all Branches
- each Order Line can be for a different Delivery Date or Branch, and can override the Billing and Shipping Addresses.

#3

Presentation guidelines

- ◆ Draw it on a whiteboard while you present it, even if you have a laptop presentation.
“If it’s too complicated to draw, it’s too complicated to present.”
- ◆ Draw it top down, adding a few entities at a time.
- ◆ Constantly illustrate the model with sample instances, definitions, schematics, etc.
- ◆ Regularly highlight features and constraints of the model, in business terms. E.g.,
“Currently we can allocate a Product to one Product Category, but this model enables us to allocate a Product to multiple Product Categories at a time, and to record changes in categorization over time.”
- ◆ Encourage participation - *the more questions and comments, the better!*

#3

SuperBlue “Provider” - 1

Individual

Institution

Area

Sorry –

I’ve had to delete the rest of the “progressive build” of this data model presentation, because it’s my client’s proprietary material.

Just remember...

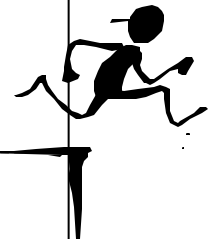
“Start simple, at the top, and add detail in layers.”

#3

A note on different types of models

Different levels of detail support different perspectives

	<i>Type of data model</i>	<i>The need</i>
1	Subject Area (Scope)	✓ Agreements on scope, context, primary terms and definitions
2	Conceptual (Concept)	✓ Agreements on basic concepts, interrelationships, and rules
3	Logical (Specification)	✓ Complete detail for physical design, and detailed application specs



Higher levels are often lost because...

“We didn’t know it was important”

“Our tool doesn’t support multiple perspectives”

“We started at the bottom level...”

#4

What does this app. mean for us? (complex)

The situation:


- major new application involving significant business process change
- each site currently has its own process
- how to identify and address “variances?”

The approach to date:

- team descends on site, does exhaustive as-is analysis
- team “returns to base,” does gap analysis
- much later, team returns to site to negotiate

The problem:

- time-consuming and mind-numbing
- process drove the sites crazy
- focus on minor issues, many important issues missed entirely



**“The sites don’t
want to see us
ever again!”**

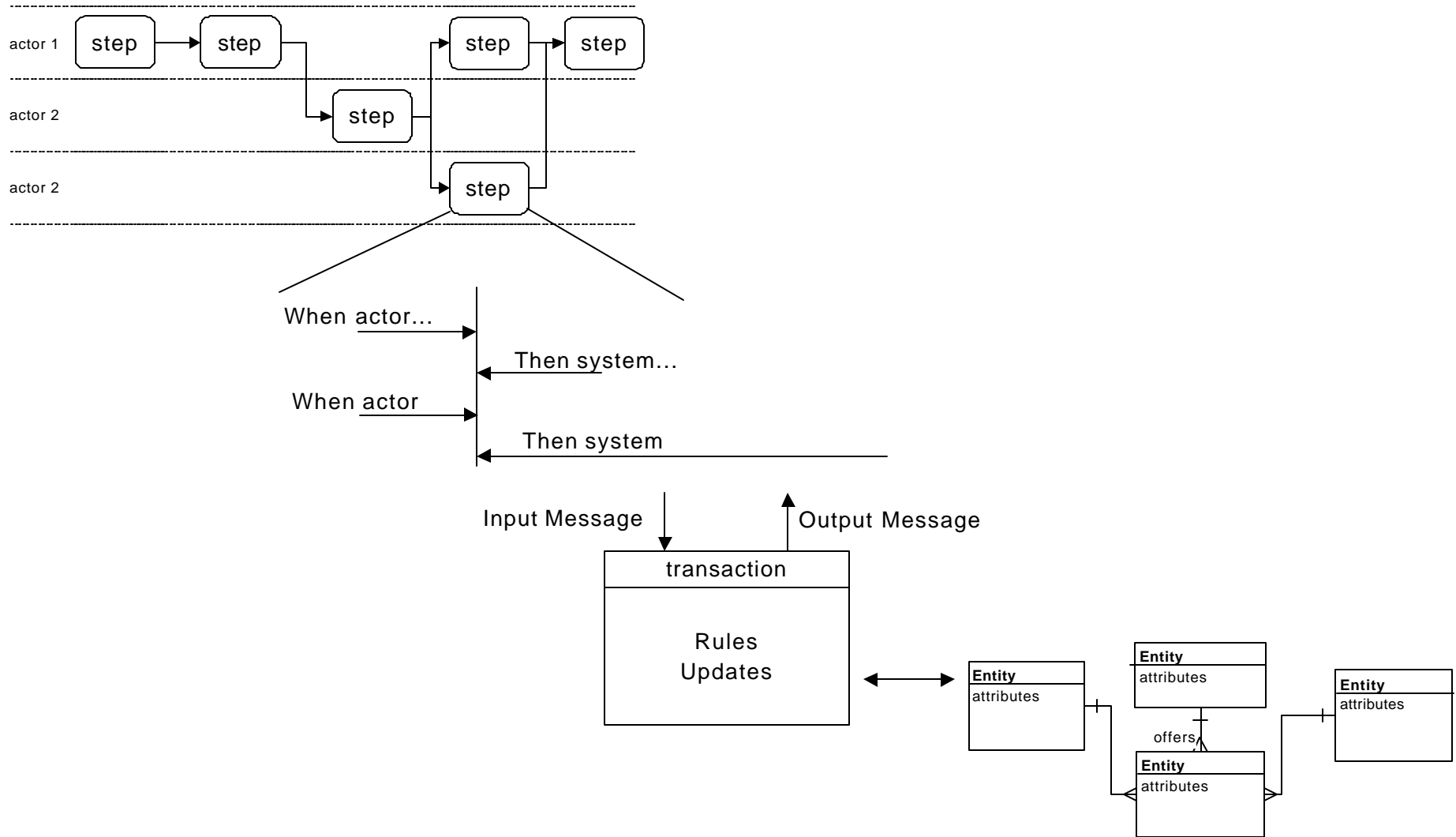
#4

An interlude on interlocking techniques

	Framework Layer	What it covers...	The Technique
Business	Business Objectives	The mission, strategies (customers/markets, products/services, differentiators), goals, objectives, CSFs, KPIs, etc. for the organization. (MSGO – Mission, Strategies, Goals, Objectives)	Project Charter
	Business Process	The activities the business carries out in order to meet its objectives. Includes the actors involved, the sequence of steps they carry out (workflow), and the result(s) produced	Workflow Process Modeling
Information Systems	User Interface	A mechanism through which an actor in a business process interacts with a system. Usually a GUI (graphical user interface) and reports, but could involve scanners, IVR (telephone) systems, etc.	Use Cases
	Application Logic	A “transaction” offered by a system – a specific function. Includes the business rules and data updates it is responsible for. Requires Event Analysis, State Transition Analysis, etc.	Application Logic Modeling
	Data Management	Files and databases that provide a system’s record-keeping functions. Determines the things a system “knows” about, and the data that is maintained about those things.	Data Modeling

#4

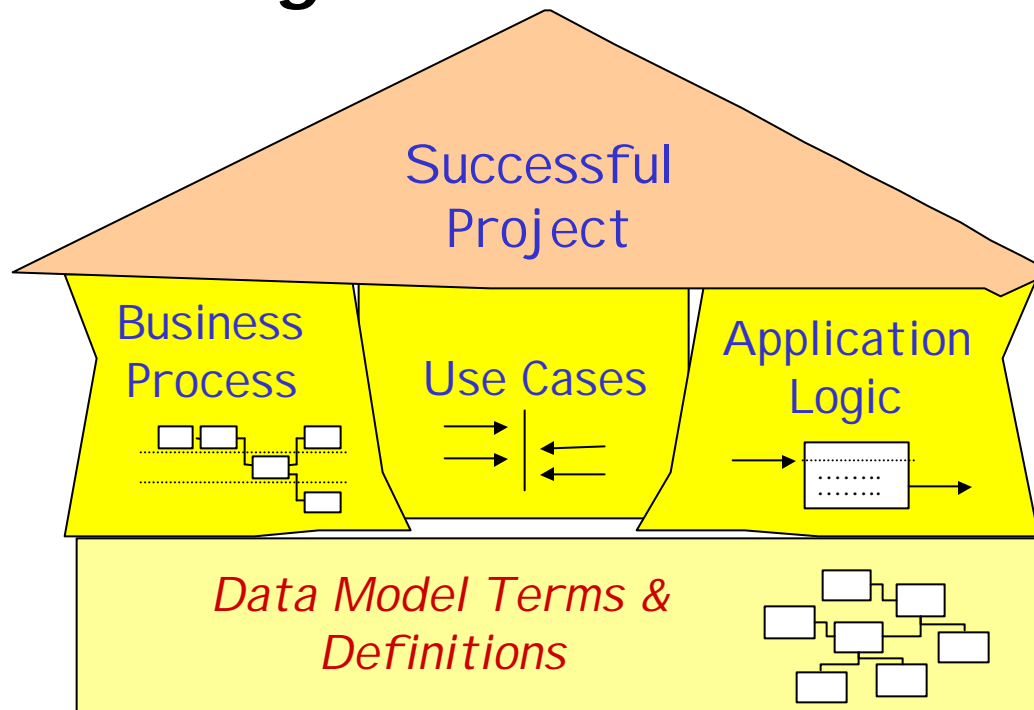
How they all hang together...



#4

It all rests on the data model...

Data Modeling....



.. is the **foundation** technique. It **supports** and provides a **common vocabulary** for all the other analysis techniques, but they **extend** and **validate** it.

#4

A better approach to “gap analysis”

The new approach

- ◆ Build ~10 “variance packages” based on package features, and previous work
- ◆ Organize “core team” of 15 to 25 reps at each site
 - cross-functional
 - included impacted staff and management
- ◆ Conduct facilitated walkthroughs of variance packages
- ◆ Core team highlights “good news!” and gaps



Each variance pack included

- ◆ Application area overview (main application functionality)
- ◆ Expected “good news!” and expected gaps
- ◆ Data Model
 - fairly detailed attributes
 - main business rules
- ◆ Workflow Model (“swimlanes”)
 - actors
 - steps
 - flow – sequence, branching, handoffs
- ◆ Use Case Scenarios
 - 5 to 12
 - progressive complexity
 - exercise entire workflow and core app. functionality

#5.1

Why do we hate this application?

A client request...

"Help!!! We totally modified SuperApp 1.0 (screens, reports, code) but we never *really* liked it. SuperApp 2.0 is now available - what should we do?"

The approach...

Behind the scenes, build conceptual data models by studying SuperApp and talking to clients. Present "world views," not "data model."

At the session

How you see the world...



How SuperApp sees the world...

Draw it Live!

#5.2

Influencing the vendor

The approach for assessing app and negotiating with the vendor:

- ◆ categorize issues into 3 Tier framework
- ◆ assign severity
- ◆ focus *only* on “High” severity data issues



Presentation	
Severity	Description
H	
H	
M	
M	
M	
M	
L	
L	

Application Process Logic	
Severity	Description
H	
H	
H	
M	
M	
L	
L	

Data Management	
Severity	Description
H	
H	
H	
M	
M	
L	
L	

#6

A quick story – the “Loopholes” project

- ◆ **“ISD messed up! This app doesn’t work right!”
(In fact, “Oh no! You did it just like I said to!”)**
- ◆ **A familiar approach**
 - **Key entities**
 - **Definitions**
 - **Events, leading to State Transition Diagrams (a useful form of business rules)**
- ◆ **Two key “revelations”**
 - **the “Account” concept didn’t work out quite as expected**
 - **Most of the “loopholes” were from clever exploitation of the “Transfer” rules
(a la www.screwtheDMV.com)**

#7

A quick story – the “New Customer DB” project

Road show version 1

“Here’s the new model. Isn’t it great?”

“So what?” “Yawn”
“ZZZZZZzzzzzzzzzz....”

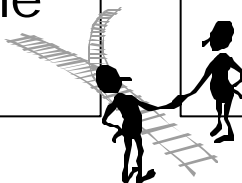
VP of IS & Finance

“You’re dyin’ out there, kid! I want you to drag them through the pain and misery of our current files and databases.
Then show them the new model.”

Road show version 2

“Let’s try answering some important questions using the current model, and then the new model”

“Fantastic!” “When do we get it?” “Do you need \$\$\$?”



***In
closing***

Many more are possible...

- ◆ **Identifying business processes**
- ◆ **Securing funding for start-up**
- ◆ **Defining semantic and syntactic translation rules for EAI undertakings**
- ◆ **New employee orientation**
- ◆ **...and many more!**

**In
closing**

Your turn...

Alec Sharp

Clariteq Systems Consulting Ltd.

206 – 2438 Marine Drive

West Vancouver, BC

Canada V7V 1L2

604 925-2440

asharp@clariteq.com

www.clariteq.com (soon!)

