More Than Just Models - Developing and Delivering a Working Data Management Service

DM&IQ Data Management & Information Quality Conference Europe 2009

Dagna Gaythorpe

http://www.seshat.com

© Dagna Gaythorpe 2004 - 2009

Agenda

- Introduction
- Services (What are we delivering?)
- Benefits (Why?)
- Tools (How do we deliver it?)
- The Enterprise Data Map
- Selling Data Management
- Where do we start?
- Questions

© Dagna Gaythorpe 2004 - 2009

Introduction - who am I?

Independent consultant - data models for utilities, telecommunications, accountancy, travel, loyalty cards and investment banking.

Member of DAMA UK. (On the committee as International Liaison, so please talk to me if you want to know more.)

Vice-president of Online Services for DAMA International

© Dagna Gaythorpe 2004 - 2009

Disclaimer

This presentation is based on the way I think things should be done.

It may not reflect reality as you know it.

It does reflect the way I do things when I get the chance – unless otherwise stated, I have done all this in real life, working, paid-employment situations.

© Dagna Gaythorpe 2004 - 2009

Data Management is a service

Data Management is a means, not an end.

If we work out what we are trying to deliver, then we can work out how.

It is much easier than working out how and then trying to work out why!

© Dagna Gaythorpe 2004 - 2009

What are we trying to deliver?

Data Management (modelling, architecture) is a service.

It is not an end in itself.

The models and repositories are not deliverables.

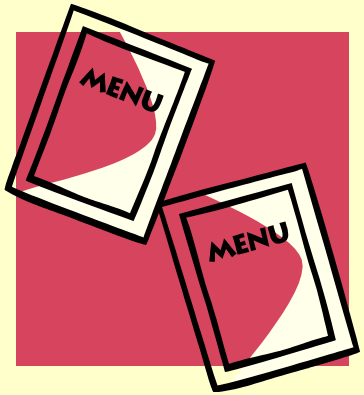
They are tools to enable the delivery of the service.

The service we deliver is the enablement of faster, better and cheaper development (of processes, strategies, products, etc – not just code).

© Dagna Gaythorpe 2004 - 2009

Services

- Project Starter Packs
- Data models/database designs
- Data lineage/Impact assessments
- Test scenarios
- Package evaluation
- Data flows/interfaces/messaging
- Metadata
- Requirements and specification reviews
- Context
- Awkward and stupid questions



© Dagna Gaythorpe 2004 - 2009

Starter Packs

At the start of a project, if we have everything in place:

- First cut logical model (subset of EDM)
- Standard definitions and formats of attributes
- Existing interfaces (for system replacements)
- Data sources (for new systems)
- Relevant rules
- Impact analysis (outgoing interfaces)

The starter pack may not contain everything, but it will deliver everything we already know.

© Dagna Gaythorpe 2004 - 2009

Data models/database designs

Use an Enterprise data model

- Re-use and consistency
- Map physical and logical system models to the EDM
- Cross-reference via EDM
- Repository for standard layouts and definitions

Review physical designs – beware ‘DBA efficiency’!

First-cut logical model in 2 hours.

Use scenarios.

© Dagna Gaythorpe 2004 - 2009

Data lineage

- Where did that come from?
- How is it calculated?
- What happens next?
- Where does it go?

This used to be called ‘Entity Life History’. You need it for BASEL II, compliance, happy auditors, and making changes.

(There is a presentation about it on my web site).

© Dagna Gaythorpe 2004 - 2009

Impact assessments

What happens if we change this?

What is the impact further downstream?

We need to know (or find out) the lineage, so we know who needs to be consulted, warned and informed.

The Impact Assessment should form part of the feasibility study for a project. Even if all it says is ‘go ahead – no problem’.

This can be time consuming if you don’t have the lineage yet.

© Dagna Gaythorpe 2004 - 2009

Test scenarios

When we are assembling a data model (from mostly pre-configured components...), we can work out what components we need from the processes to be supported.

Knowing the processes, we can devise scenarios for the model – add a customer, place an order, launch a product, etc

Use these to test the model – and the application.

© Dagna Gaythorpe 2004 - 2009

Package evaluation

"We don't need architecture – we use packages".

You still need architecture to make them fit together, and to work out which new ones will fit.

Use the scenarios to evaluate a package. Especially if the vendor provides a data model. (Rarely, but always ask.)

© Dagna Gaythorpe 2004 - 2009

Data flows/interfaces/messaging

- Designing and deciphering interfaces and messages
- Just because you can do a lot with XML, it doesn't follow that you should!
- Do you really need XML for internal interfaces?

© Dagna Gaythorpe 2004 - 2009

Metadata

My definition: Metadata is the stuff I need to know about the data I use.

So someone else needs different metadata.

We supply metadata to the business and IT – it is our data. And we supply a lot, to fulfil a lot of different needs.

Don’t give everything to people who don’t need it!

© Dagna Gaythorpe 2004 - 2009

Requirements and specification reviews

- Sanity check
- Find scenarios
- Check existing scenarios are supported
- Rules
- Check context
- Definitions

© Dagna Gaythorpe 2004 - 2009

Context

From Answer.com:

1. The part of a text or statement that surrounds a particular word or passage and determines its meaning.
2. The circumstances in which an event occurs; a setting.

[Middle English, composition, from Latin contextus, from past participle of *contexere*, to join together : com-, com- + texere, to weave.]

For us, the ‘stuff’ that surrounds a data item.

© Dagna Gaythorpe 2004 - 2009

As described by Kipling:

I keep six honest serving-men
(They taught me all I knew);
Their names are What and Why and When
And How and Where and Who.

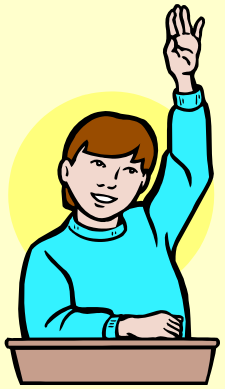
("I keep six honest serving men"
from "The Elephants Child"
by Rudyard Kipling)

© Dagna Gaythorpe 2004 - 2009

Awkward and stupid questions

This can be a tough one...

- Recognise assumptions and challenge them.
- Ask the questions that people think have obvious answers – just in case!
- Be persistent.
- Nit-pick.



Especially vital in vendor presentations!

© Dagna Gaythorpe 2004 - 2009

Benefits

- Repeatable designs
- Consistency
- Improved quality measurement
- Faster, better cheaper development and implementation

© Dagna Gaythorpe 2004 - 2009

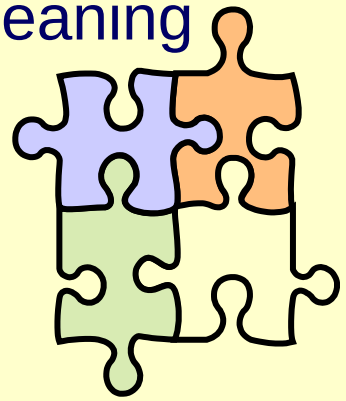
Repeatable designs

- Design it once and re-use it
- Second and subsequent use is ‘free’
  - There is a cost to look it up, but it is less than the cost (and time) of an original design
- Component-based designs can be ‘assembled’ by more (less experienced) people
- Final checking and approval by more experienced (expensive!) people
- Building on existing patterns

© Dagna Gaythorpe 2004 - 2009

Consistency

- Comes from repeatable designs
- Honoré Blanc began making guns with replaceable parts in about 1778
- If a data item (e.g. Net Sales) is always stored the same way and has the same definition:
  - ETL becomes EL
  - Wherever it appears, it has the same meaning
  - Essential for SOA



© Dagna Gaythorpe 2004 - 2009

Improved quality measurement

Answers.com on ‘Quality’:

1.
    1. An inherent or distinguishing characteristic; a property.
    2. A personal trait, especially a character trait: *“The most vital quality a soldier can possess is self-confidence”* (George S. Patton).
2. Essential character; nature: *“The quality of mercy is not strain'd”* (Shakespeare).
3.
    1. Superiority of kind: *an intellect of unquestioned quality.*
    2. Degree or grade of excellence: *yard goods of low quality.*
4.
    1. High social position.
    2. Those in a high social position.
5. *Music.* Timbre, as determined by harmonics: *a voice with a distinctive metallic quality.*
6. *Linguistics.* The character of a vowel sound determined by the size and shape of the oral cavity and the amount of resonance with which the sound is produced.
7. *Logic.* The positive or negative character of a proposition.

We generally use ‘quality’ in the sense of the third definition.

© Dagna Gaythorpe 2004 - 2009

Improved quality measurement (2)

Data management services and tools won't improve data quality.

They will help us to:

- identify what we should look at,
- find problems
- measure the level of quality

Data quality comes from people and processes.

© Dagna Gaythorpe 2004 - 2009

Faster, Better, Cheaper

Pick any two...

© Dagna Gaythorpe 2004 - 2009

Faster, Better, Cheaper

But if you have already done a lot of the basic work,
and can reuse it
and only have to invent the new stuff…

Then you can have it all for new projects.

© Dagna Gaythorpe 2004 - 2009

Faster, Better, Cheaper

Faster - because you have one you did earlier.

Better - because it has already been checked by other projects, and is probably already in use somewhere in the organisation.

Cheaper - because the project only has to pay for the new stuff (plus a bit for getting the old stuff out of the central model).

© Dagna Gaythorpe 2004 - 2009

Faster, Better, Cheaper

And if you are reusing data formats, then you reduce the amount of data conversions, checking, reformatting and converting.

Simpler interfaces, faster development, better quality…

Faster, better and cheaper again!

© Dagna Gaythorpe 2004 - 2009

Tools

- Enterprise Data Models
- Application Register
- Business Process Model
- Physical Data Models
- CRUD Matrix
- Application Data Flow Diagram
- Data Glossary
- Application and Process Entity Life Histories
- Entity/Application/Process Cross reference Matrices

© Dagna Gaythorpe 2004 - 2009

Enterprise Data Models(1)

Two models – conceptual and logical

Conceptual:

- Holds the business concepts – high-level nouns
- 20-30 concepts
- Definitions
- No attributes
- Expand to subject areas

© Dagna Gaythorpe 2004 - 2009

Enterprise Data Models(2)

Two models – conceptual and logical

Logical:

- The ‘bones’ of the business
- The underlying structure of the data
- Attributes
- Name and define everything – entities, attributes, relationships, rules, source, etc
- Include standard formats

© Dagna Gaythorpe 2004 - 2009

Data flow diagram

Data flows between applications

- Number each flow – link to a repository/spreadsheet
- Show each application once only!
- Two-way flow is probably two flows

Links to Application Register and Enterprise Data Model

The data flows can be used to start/derive an application data model.

© Dagna Gaythorpe 2004 - 2009

Application register

List of all known applications:

- Formal name
- Aliases
- Brief description
- People/department/team who own/maintain/use/know about it
- Supplier (for packages)
- DBMS
- Language(s)
- Platform

© Dagna Gaythorpe 2004 - 2009

Lineage (aka Life History)

Build these for major entities.

Add new ones as you need them.

Two types – process and application.

They are not Use Cases!

© Dagna Gaythorpe 2004 - 2009

Scenarios

- Record what we do with the data.
- Based on process lineage.
- If you have process modellers, there is an overlap.
- Include reporting requirements.
- Use to test data models (virtual SQL).

© Dagna Gaythorpe 2004 - 2009

What are we trying to deliver?

Money.

- Getting more out of the data for the same or less money.
- Developing things (applications, reports or services) for the same or less money.
- Not paying fines for non-compliance

But it is very hard to measure how much we deliver.

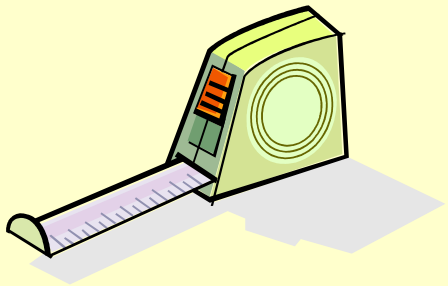
© Dagna Gaythorpe 2004 - 2009

Can we measure it?

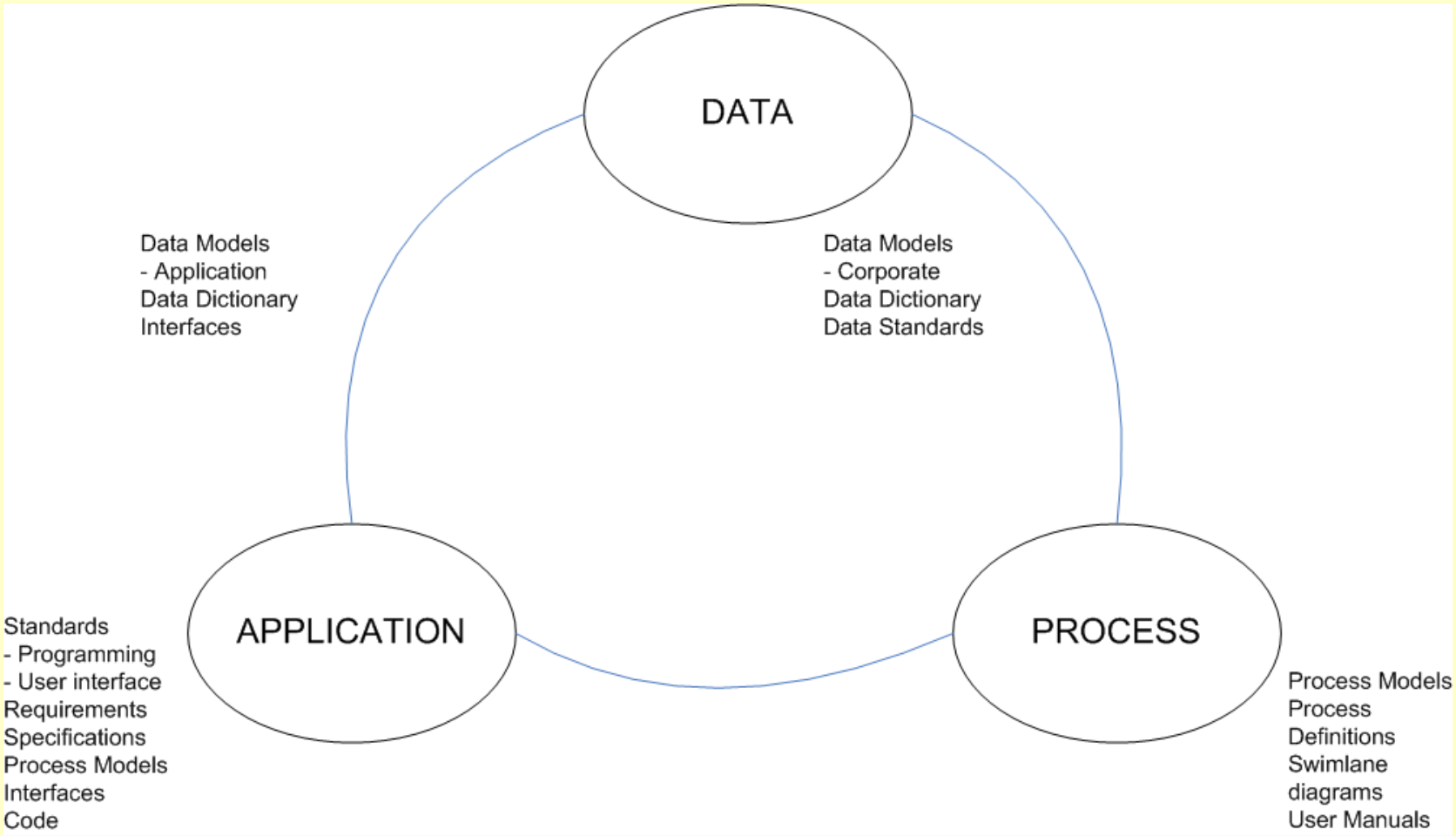
We can measure how often the work we have done is re-used (entities, attributes, tables, columns, definitions...), and work out the money saved by re-use.

Maybe we can measure how much effort (time and resources) it takes to deliver each function point (if we have base figures).

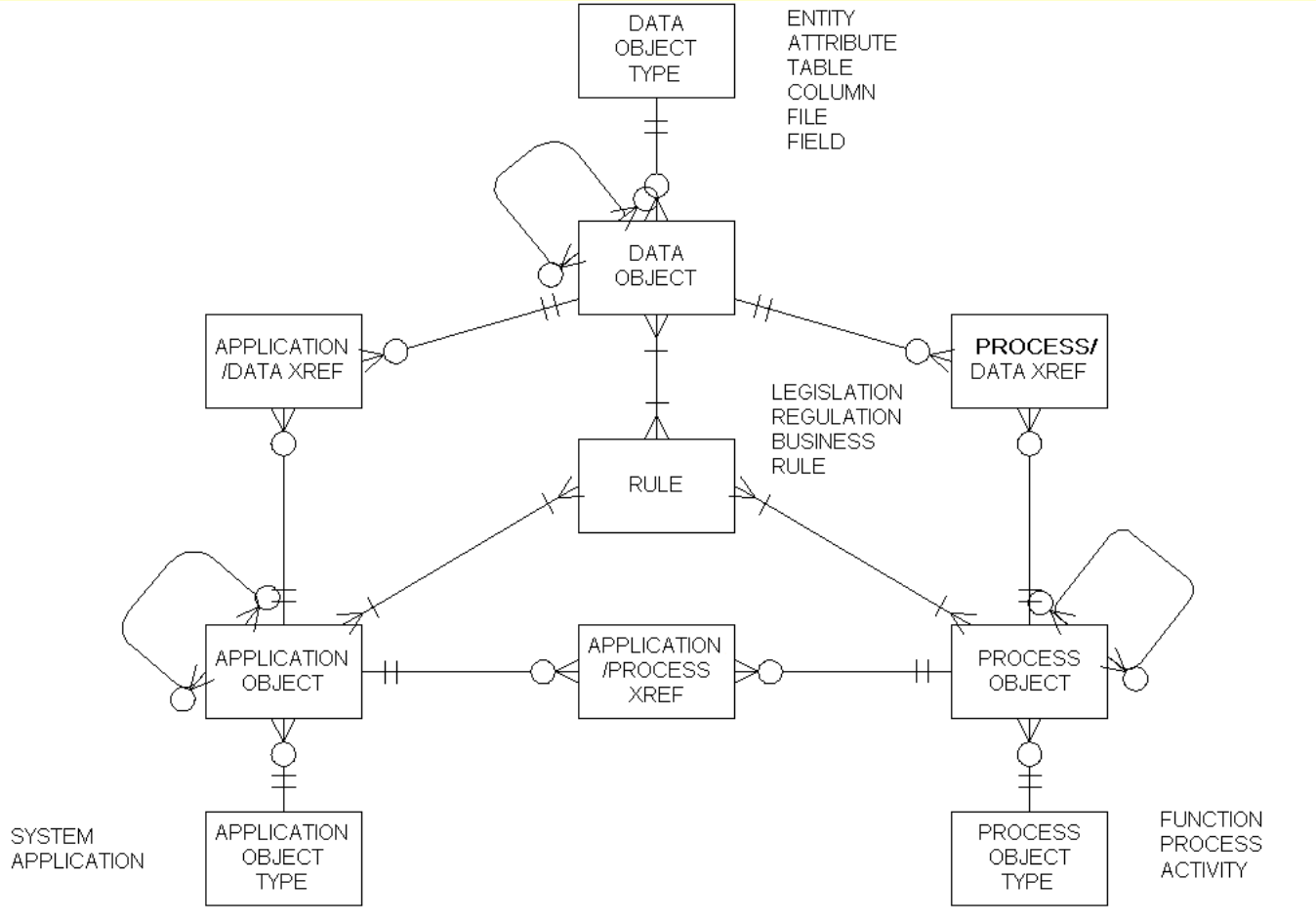
Will we?



The Enterprise Data Map



The Enterprise Data Map



© Dagna Gaythorpe 2004 - 2009

Selling data management

A digression...

Data management benefits are hard to measure.

This can make it hard to sell.

© Dagna Gaythorpe 2004 - 2009

Selling data management (2)

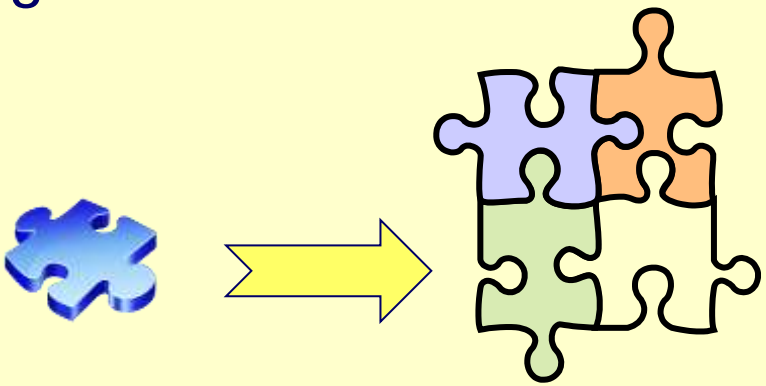
Some workable approaches:

- Re-use (applies to both software and data)
- Mass production – data as components
- Speed limits and road signs

© Dagna Gaythorpe 2004 - 2009

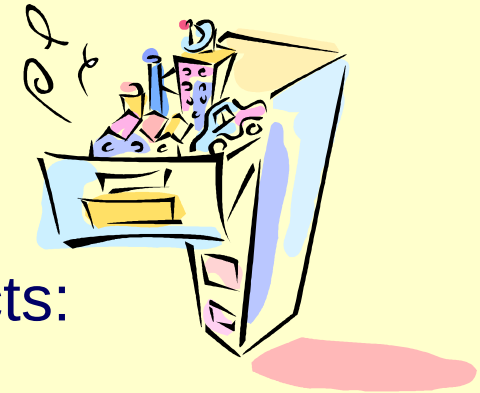
Sales pitch – Re-use

- Design it once, use it many times.
- Each time we re-use it, we get the benefit of not spending time (re)designing
- If we have consistent data designs, definitions, and structures, it makes passing data around much easier by reducing transformations
- It also makes reporting easier and more consistent



© Dagna Gaythorpe 2004 - 2009

Sales pitch – mass production



- This is, basically, advocating SOA
- Remember objects?
- Design and deliver standard data artefacts:
  - standard meaning
  - everyone uses the same thing
  - everyone gets the same information
  - everyone gets consistent results
- You need a fairly high level of (CMM-type) data management maturity to do this.

Sales pitch – speed limits

Governments have spent millions getting the benefits of speed limits across.

- Reduce accidents
- Reduce serious accidents
- Reduce congestion

If we equate data management to speed limits:

- Reduce “wrong data” incidents
- Reduce legal breaches (wrong numbers filed, etc)
- Reduce time waiting to use data

The value comes from what we don’t get!

© Dagna Gaythorpe 2004 - 2009

That's a great idea.

But where do we start?

- Application register and data flow diagram when talking to IT.
- Entity Life Histories/lineage works well with business people.

Avoid the 'What data do you need to run the business?' trap.

© Dagna Gaythorpe 2004 - 2009

Contact Information

Dagna Gaythorpe

http://www.seshat.com

dgaythorpe@seshat.com

@dagna

© Dagna Gaythorpe 2004 - 2009