Adapting Zachman for Business Objects

An Inside-Out Look from the Outide-In...

Open Engineering Inc. www.openeng.com

TERMS OF REFERENCE

About This Presentation

- Presented by...

  Robert E. Shelton, President & CEO

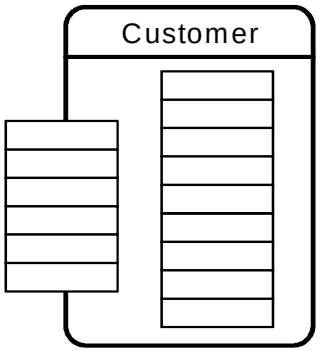
  Open Engineering Inc.
  50 California Street, Suite 860
  San Francisco, California, USA 94111
  (v) 415-989-9050 (f) 415-989-9055
  (w) www.openeng.com

- Presented to...

  OMG Object-Reference Model Subcommittee
  June 1997 Meeting, Montreal, Quebec, Canada

What is a Business Object

- A business modeling or software ***package of business procedure***, ***policy*** & ***rules wrapped around data***
  - Each business object represents a single defined business concept
- A way to organize the ***right data and right procedure in the right place***
- ***Sharable*** (i.e. business-reusable)
- ***Independent of applications***

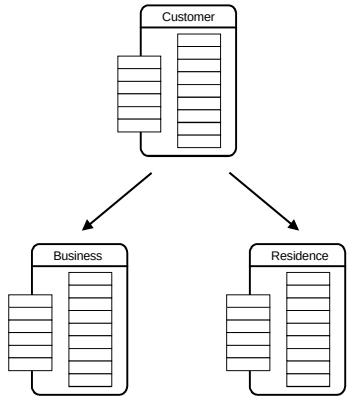


The Understanding Gap

- In business modeling...
  - ***Business objects abstract domain concepts***
  - Not the concept in the context of any one application
  - Not some concept from the technology infrastructure
- In implementation...
  - ***The interface is not necessarily technically different*** from an application or any CORBA object in general
- The ***difference is in purpose, content, subject***
  - The shape at the interface does not have to be different for the concept to be different, valid and important

Why Business Objects?

- ***Systems mirror the business***
- ***Internal & external sharing***
  - Business data & procedures
  - Business rules & integrity constraints
- ***Managing differences and change in business rules, procedures, data***
  - Put divisional/local business rules in the specializations
  - Maintain corporate definitions, rules and data in the generalization
- ***Specialize CBOs to industry domains and regionalization-nationalization***



The Understanding Gap

- The motivation for business objects is to...
  - ***close the gap between business understanding and software implementation***
  - by producing system components that closely resemble recognizable business concepts
- Thus...
  - ***Business objects are differentiated by their intent rather than by their interface***
  - ***Business objects are considered a specialization of object in order to achieve architectural separation of concerns***

Extensions of Terms...

- Common Business Object (CBO)

  A business object that exists in 2 or more domains
- Domain

  An industry sector (approximately a S.I.C. grouping)
- Domain Business Object (DBO)

  A CBO that has been specialized for one domain
- Locale

  Region, political jurisdiction
- Localized Business Object (LBO)

  A CBO or DBO that has been localized

MAPPING TO ZACHMAN

Taxonomy of Business Objects

- There are ***3 types*** of business objects...
  - ***Entity Business Object***
    - Person, place, thing or concept
    - Business noun

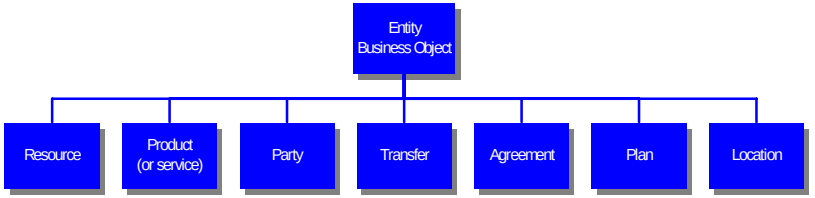
  - ***Process Business Object***
    - Business process, workflow, activities
    - Business verb that require multiple nouns
    - Structured collection of entities, interactions, events

  - ***Event Business Object***
    - An event that causes or results from processes or actions
    - Occurrences, interruptions, passage of time

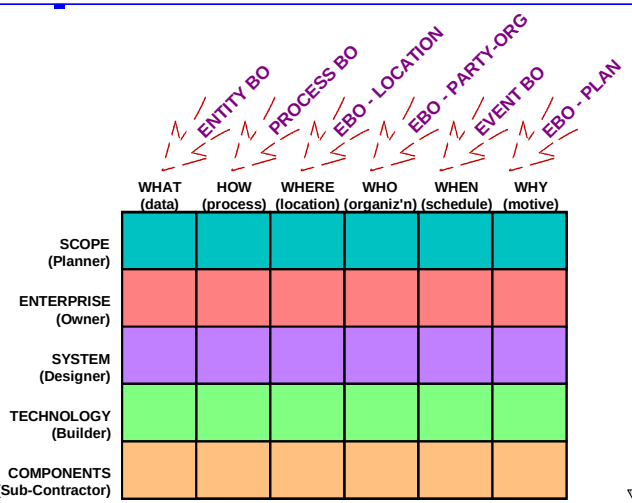
Zachman-Sowa Framework

	WHAT (data)	HOW (process)	WHERE (location)	WHO (organiz'n)	WHEN (schedule)	WHY (motive)
SCOPE (Planner)						
ENTERPRISE (Owner)						
SYSTEM (Designer)						
TECHNOLOGY (Builder)						
COMPONENTS (Sub-Contractor)						

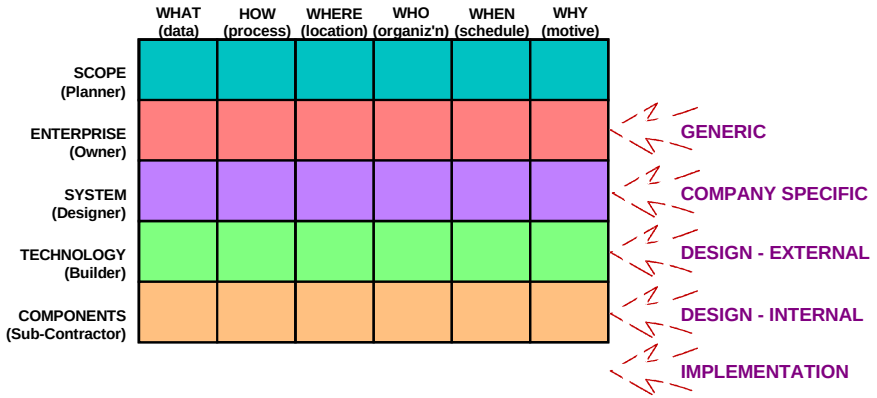
7 Base Entity Business Objects



Map Taxonomy to Zachman



Map Levels to Zachman



Levels of Abstraction

- ***Business Abstractions***
  - ***Generic***
    - ***Common - horizontal, cross domain***
    - ***Domain - vertical, industry***
    - ***Localized - regional, national***
  - ***Company specific***
    - ***Enterprise - company wide***
    - ***Business area - business unit, department***
    - ***Individual - working group***
- ***Software Abstractions***
  - ***Design***
    - ***External - public interface protocol, class structure***
    - ***Internal - methods, attributes, constraints, mappings***
  - ***Implementation***
    - ***Source code - "human readable" target language***
    - ***Executable code - format determined by run time VM***

CONCLUSIONS

Implications

- The separation of...
  - ***Generic from business-specific*** supports the idea of selling generalized, abstract type libraries that each company will specialize to its own needs
  - ***Horizontal from vertical*** positions OMG vertical-industry TF work for specialization of more abstract CBOs
  - ***Vertical from regional*** positions economic or national interest areas (i.e. JSIG) for specialization of industry or common business objects
- Why architectural separation of concerns?
  - ***The business and technology-based implementations are different things and change independently of one another***

Implications

- Concept (business) to code (interface)
  - The same business object can "exist" at different levels
  - A business object does not have to be coded to be useful
  - CBOs are powerful tools for business engineering/planning
- ***Within this framework we can organize OMG to deliver consistent business components based on a core set of concepts in common***

