

DB/EXPO '91



Section B12
Object-Oriented Development:
Treat 'em like Objects

David McGoveran
President - Alternative Technologies
Principal - Database Associates

**Object Oriented Development
Treat 'em Like Objects
David McGoveran**

ABSTRACT

Development using the object oriented paradigm involves acceptance of a number of assumptions about the nature of the world. Successful development of object-oriented programs that interface to relational databases, perform OLTP and OLCP functions, and work in a client/server or distributed environment, require careful understanding of these assumptions and their impact. Both the benefits and the costs of object-oriented development will be examined, especially in the context of these more complex database applications.

BIOGRAPHY

David McGoveran co-founded and is a Principal of Database Associates, along with Richard Finkelstein, Colin White and Paul Winsberg. He is President of Alternative Technologies, a Santa Cruz, California firm which has specialized in complex relational database applications for over ten years and which is an affiliate of Database Associates.

Mr. McGoveran designed and developed the first commercial CIM system using a relational database. He has authored numerous technical articles and has lectured around the world. Author of the forthcoming 'Guide to Sybase SQL Server' with Chris Date, he serves as an associate editor of InfoDB magazine. Mr. McGoveran is listed in Who's Who in the World and Who's Who in the Computer Industry. He can be contacted at Alternative Technologies - 408-425 1859.

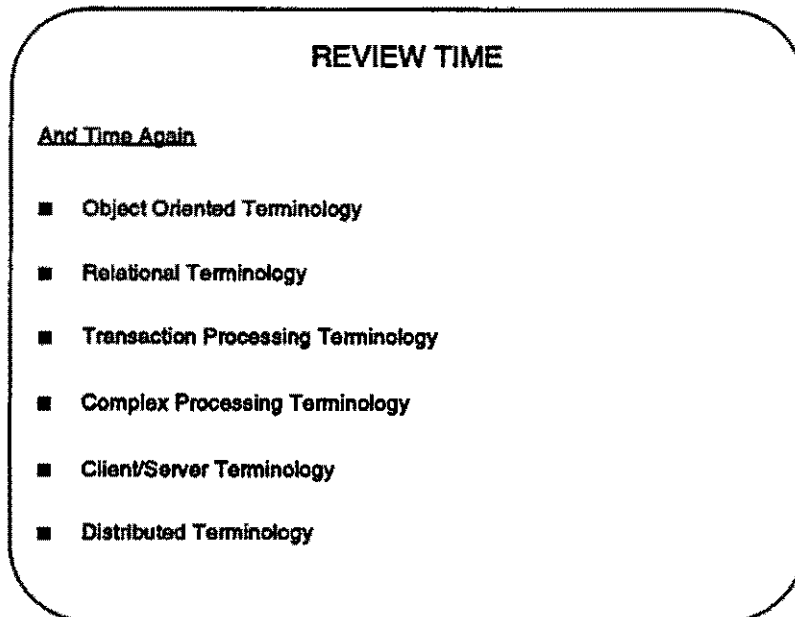
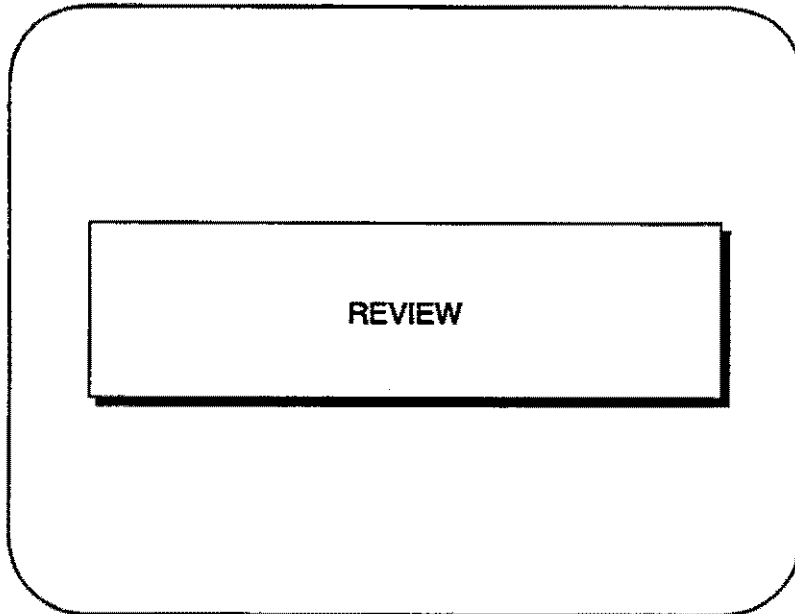
OVERVIEW

OBJECT ORIENTED DEVELOPMENT

- A Brief Overview
- Is It Development Or Design?
- Some Key Object Oriented Principles
- The Ideal Task
- Reality Intrudes
- If I Do It Will It Be GUI?

OBJECT ORIENTED DEVELOPMENT Cont'd

- Can I Do It With Relational?
- Can I Do It With Transactions (OLTP)?
- Can I Do It With Complex Processing (OLCP)?
- Can I Do It With Client/Server?
- Can I Do It With Distributed?



HOPES AND PROMISES

IS IT DEVELOPMENT OR DESIGN?

Some Convenient Definitions

- Object Oriented Analysis
- Object Oriented Design
- Object Oriented Development
- Traditional Versus Non-Traditional Classifications

IS IT DEVELOPMENT OR DESIGN?

Some Key Tasks

- Language Selection
 - What object oriented features are needed?
- What Software Architecture is Required?
- What Are The Boundaries?
 - How to know a brick wall when you hit it!
- How Dynamic Is The Target Environment?
 - Can you detect a dud in mid-flight?
- Goals:
 - Data Structures and Methods
 - Reusability, Extensibility, Robustness, Maintainability

OBJECT ORIENTED PRINCIPLES

Sticking To It

■ Order

Contrary to common opinion, OOP takes a neutral approach toward ordering - there is no top to the call structure, no hierarchy, no application-wide functional decomposition.

OBJECT ORIENTED PRINCIPLES

Sticking To It Cont'd

- **This Assumes A Complete Orthogonal Basis**
A minimal set of classes (called the Basis) exist such that:
 - Every class to be implemented consists solely of properties and services held by the union of this set of classes
 - Each class of the basis shares no properties or services with any other class of the Basis
 - No superclass or any of the basic classes will be created - i.e. the classes to be implemented require only multiple inheritance from classes in the Basis.

THE IDEAL TASK

If I Controlled Everything

- All Objects Well Defined
- All Objects Stable
- All Objects "Universal"
- Everything An "Object"

REALITY INTRUDES

I Don't Control Many Things

- Here A Definition, There A Definition, Everywhere A
- Who Changed These Requirements
- Incompatible Views Of The World
- Are properties "Real"? Are Objects "Real"?
 - Pessimist: "There's nothing here but processes"
 - Optimist: "There's an object in here somewhere"
- The Task Has No Natural Boundaries

IF I DO IT WILL IT BE GUI?

Ways To Encapsulate A User

- What Is Point And Click?
- BLOBs As ICONS
 - Pick large object and place?
- Events, Commands, And Messages
- Natural Language Messages
 - Do we always know what we are talking about?
 - Or even whom we are talking to?

CAN I DO IT WITH RELATIONAL?

Relational Model vs Object Oriented Paradigm

- Different Purposes
- Data Sharing
- Integrity
- Concurrency
- Data Primitives
- Operational Primitives
- Predictable Behavior
- The Impedance Mismatch Problem

CAN I DO IT WITH TRANSACTIONS (OLTP)?

The Purpose of Transactions

- Transactions Preserve Some Property
- Transactions Preserve Some Object
- How Many Transactions Per Second
- Which Transaction Is Longer?
- It Failed! Will We Ever Recover?
- Availability

CAN I DO IT WITH COMPLEX PROCESSING (OLCP)?

Understanding Complexity

- If It's Bleeding Edge
- If It's Volatile
- If It's Big
- If There Are Lots Of Users
- If It's Critical

CAN I DO IT WITH CLIENT/SERVER?

The Ideal Architecture

- What Is Client/Server?
- Treat The Server Like An Object
- Send It Messages
- Can A Client Be An Object?
- Are There Any Problems?

CAN I DO IT WITH DISTRIBUTED?

Or, "Where Oh Where Has My Object Gone?"

- Can Objects Be Distributed?
- Who Owns The Object
- Is Your Object Single Threaded?
- Who Manages Concurrency?
- Who Manages Recovery?
- Are Messages Serializable?

SUMMARY

When It Works, Treat 'Em Like Objects

- It Won't Always Work
- But When It Does

 - It works well
 - The benefits are great
 - And it isn't a silver bullet
 - And you can treat 'em like objects

