

**Digital
Consulting
Inc.**



Digital Consulting, Inc.

presents

Friday Seminars
MARCH 3, 1995

Seminar Workbook

**Digital
Consulting
Inc.**



presents

SOFTWARE WORLD USA

**Friday
March 3, 1995**

© Copyright and Presentation Rights Reserved

No copy of presentation use may be made of any material
in this book without the express written permission of
DCI and George Schussel

This Book Belongs to:

Name: _____

Company: _____

Address: _____

Phone: _____

Table of Contents

Digital Consulting, Inc. Section A

Full Day Seminars

9:00 am - 5:30 pm

Pieter Mimno
Twelve Technologies That You Should Know in the 90s..... Section B

Steve Barsh
Secrets to Success with OO, Client/Server, and CASE..... Section C

Dr. Roger Pressman
A Process Improvement Workshop..... Section D

Ken Orr
Data in Jail Section E

Carl Argila
Finding & Keeping Goods Object..... Section F

Roger Burlton
Managing Business Process Reengineering: The IS Challenge Section G

Finding & Keeping Good Objects

SOFTWARE WORLD / USA

Chicago

3 March 1995

Carl A. Argila, Ph.D., Inc.

SOFTWARE ENGINEERING CONSULTANT

P. O. Box 1219-B

Pico Rivera, CA 90660-1219

310-699-3196

carl@acm.org

Seminar Objectives

- ▼ Familiarize participants, as necessary, with the basic concepts of object-orientation.
- ▼ Introduce techniques which can be used to identify appropriate, re-usable object-classes.
- ▼ Incorporate these object-classes into complete, documented object-oriented information model.
- ▼ Discuss issues related to the successful application of these techniques into project environments.

Finding & Keeping Good Objects

Seminar Topics

1. Today's Challenge
 - Introduction & Motivation
 - Identifying the Problem
 - Ever Increasing Software Complexity
 - Ever Increasing Market Pressures
 - Seeking Solutions
 - Automation of the Software Process
 - The Methods Revolution

2. Establishing a Starting-Point
 - Identifying Problem Domain Sources
 - The Role of Application Domain Experts
 - Resource Techniques
 - Interviewing Techniques
 - Classification of Resources
 - Capturing Requirements
 - Capturing "Business Rules"
 - Requirements Definition Modeling

3. Information Analysis Techniques
 - What's in a Name?
 - First-Order Analysis
 - Identification of Initial Concepts
 - Second- and Higher-Order Analyses
 - Identification of Complex Concepts
 - Two-Dimensional and Multi-Dimensional Analyses
 - Identification of Relationships

4. Modeling Techniques
 - Graphical Methods
 - Associating Concepts with Icons
 - Notation & Conventions
 - Correctness Criteria

5. Establishing a Repository
 - Automation: Pros & Cons
 - Repository Access Models
 - Boot-Strapping the Repository

- Living vs. Dead Repositories
 - Assimilating into the Project Culture
6. Correctness Criteria
 - Syntactic vs. Semantic Correctness
 - Verification Rules
 - Internal Consistency
 - Validating Business Rules
 - Establishing Requirements Traceability
 7. Refining the Information Model
 - "Fine Tuning" Informal Standards & Conventions
 - Establishing a Formal Review Process
 - Establishing an Iterative Mechanism for Change
 - Predicting Problem Domain Change
 - Establishing Multiple Sub-Domains
 8. Documenting the Information Model
 - Arguing Against It!
 - Living vs. Dead Models
 - Documentation Techniques
 - Useful Reports and Matrices
 9. Maintaining the Information Model
 - Maintenance "Paradigms"
 - Establishing Maintenance as part of the Project Culture
 - Arguing for CASE
 - Evaluating CASE
 - Maintaining Requirements Traceability
 10. Moving into Object-Oriented Analysis & Design
 - What's Missing?
 - Elements of Object-Oriented Methods
 - Establishing Requirements Traceability at the Object Level
 - How Analysis & Design "feedback" to the Information Model
 - Refining the Information Model
 11. Getting Started
 - Process Maturity Self Assessment
 - Tactical vs. Strategic Change
 - Recognizing the Problem Before Providing the Solution
 - Identifying Strategic Objectives
 - Managing a Changing Process

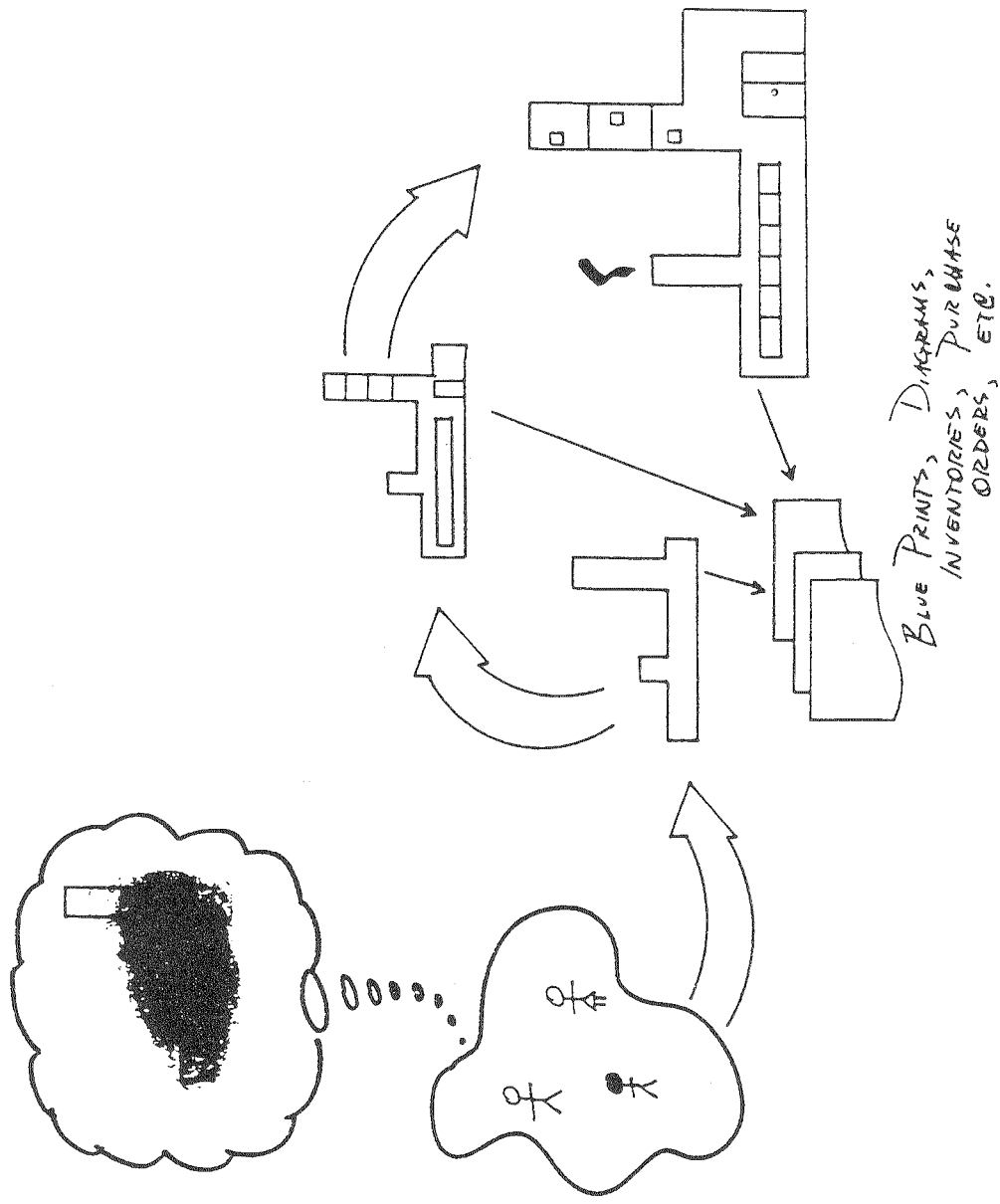
Key Point

- ▼ Much of what is presented in this seminar is akin to a "chaos effect" for software -- small changes today having dramatic effects in the long term.

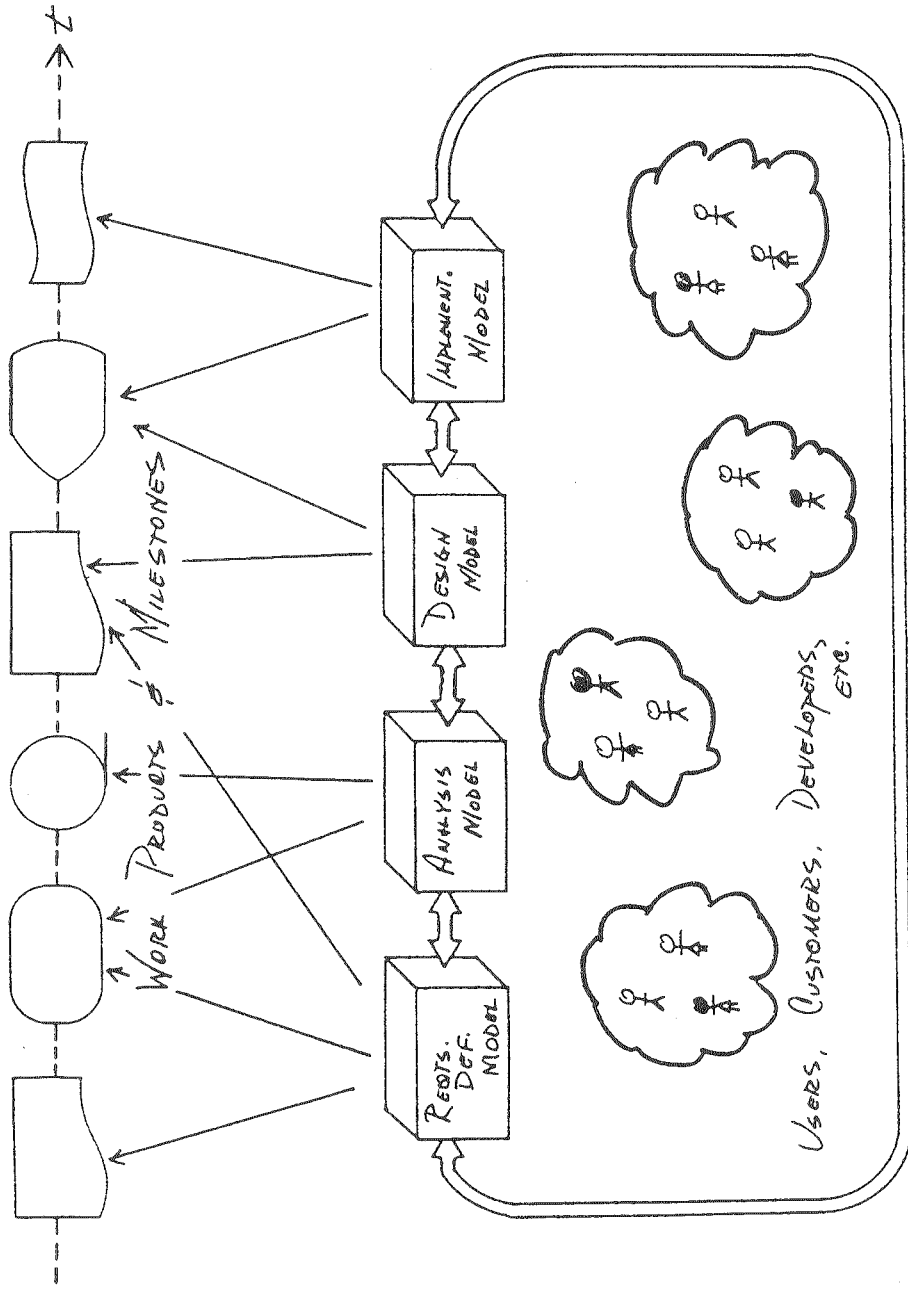
Today's Challenge

- Introduction & Motivation
- Identifying the Problem
- Ever Increasing Software Complexity
- Ever Increasing Market Pressures
- Seeking Solutions
- Automation of the Software Process
- The Methods Revolution

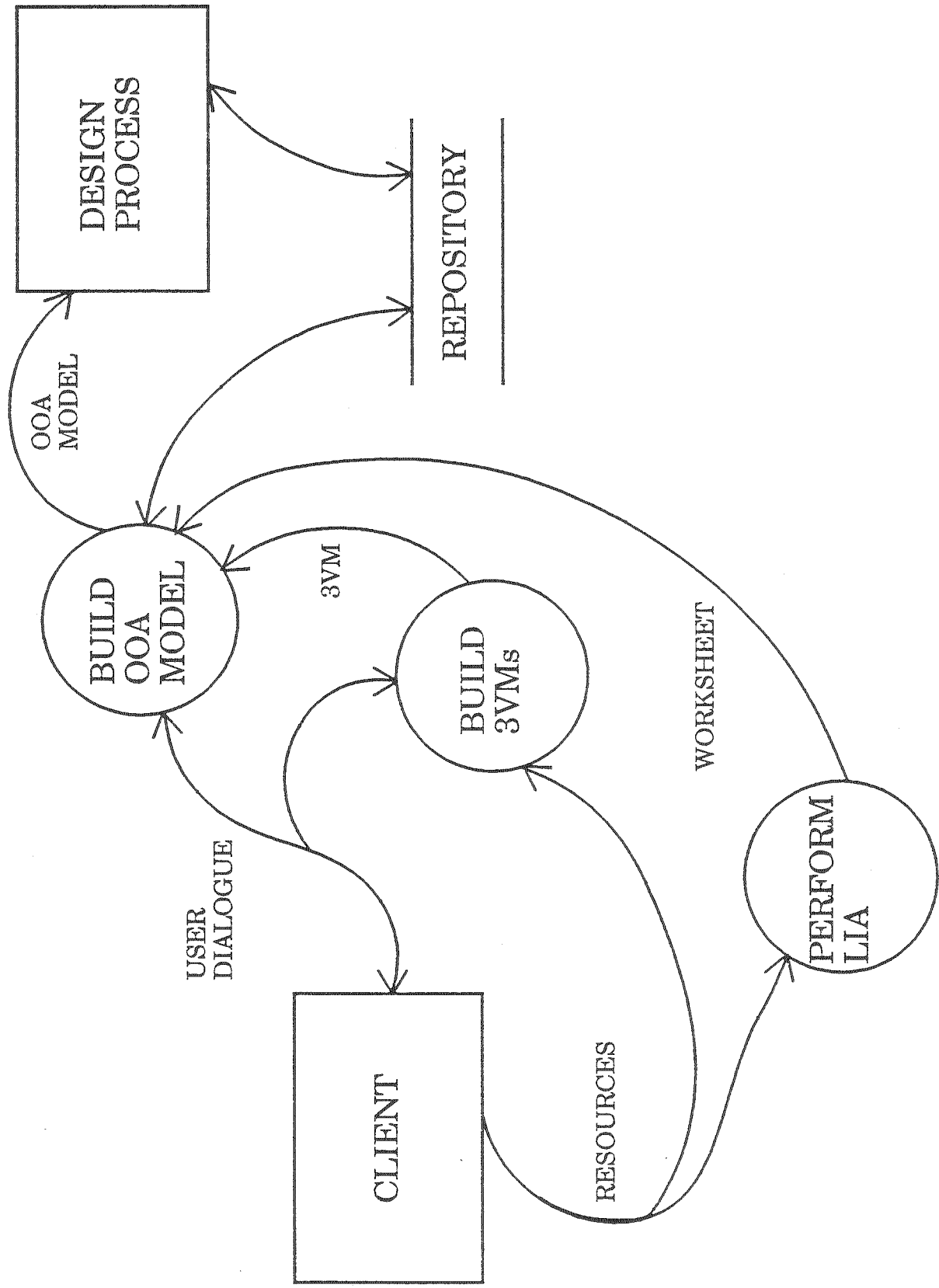
Building Models of Software Systems



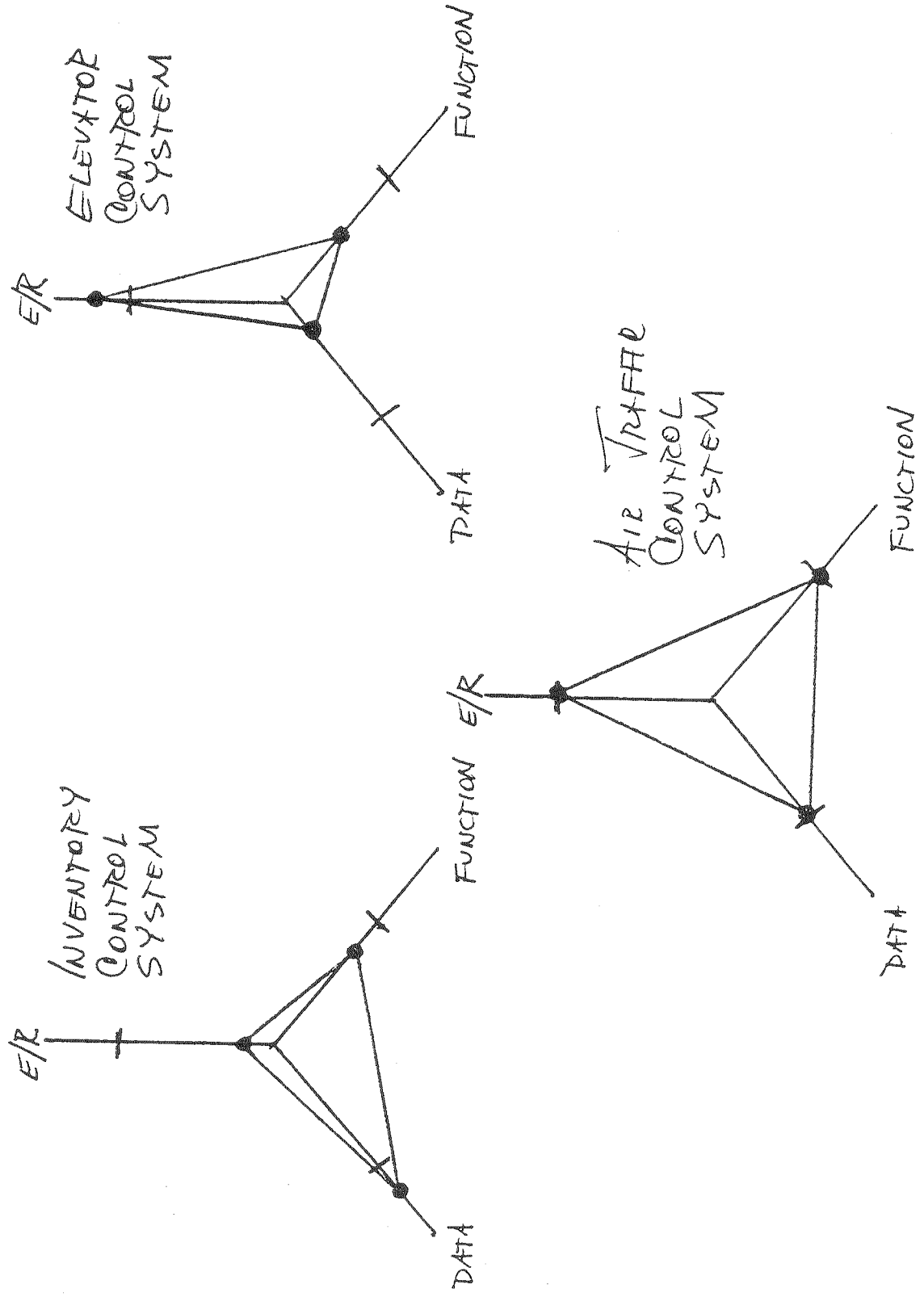
An Overall Modeling Life-Cycle



Overview of Object-Finding Techniques



"System Views"



Key Points

- ▶ Systems may be modeled from different "perspectives" -- consider, for example, the "elevator control system" vs. the "inventory control system."
- ▶ Which "view" is modeled depends upon the intrinsic nature of the system being specified. More than one view may be modeled.
- ▶ O-O methods must be capable of modeling the predominant "system view(s)" or they are useless!

Key Points

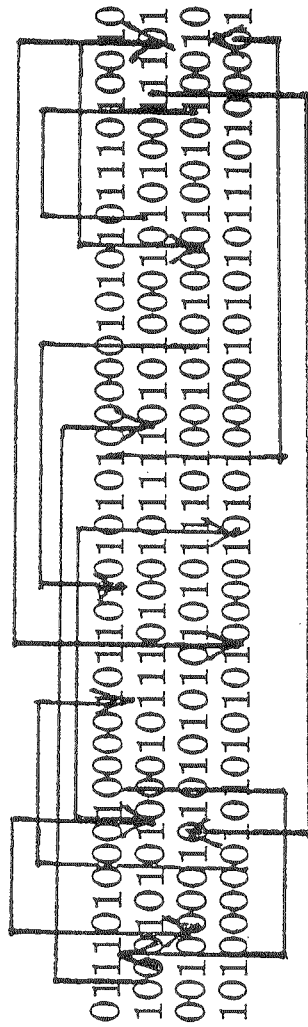
- ▼ The "Holy Trinity" of Systems Modeling:
 - Process Models (DFD's)
 - Data Models (ERD's)
 - Event/Response (State) Models (STD's)
- ▼ Events:
 - Occur at an "instant" in time,
 - Occur in the system's "perception/action space," and
 - The system has a pre-planned response.
- ▼ Event/Response models are related to (but more basic than) "use cases."

Key Points

- ▼ Events must:
 - Occur at an "instant" in time,
 - Occur in the system's "perception/action space," and
 - The system must have a pre-planned response.

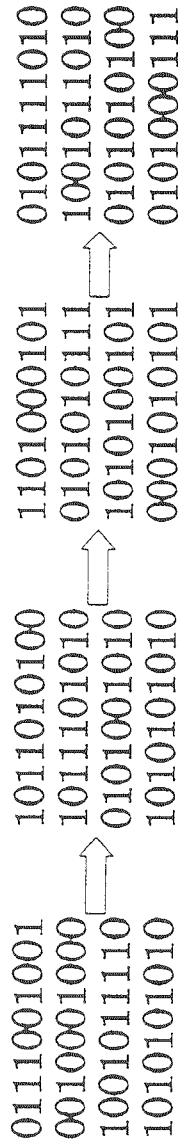
- ▼ Event/Response models are related to (but more basic than) "use cases."

Object Orientation -- What's The Point???

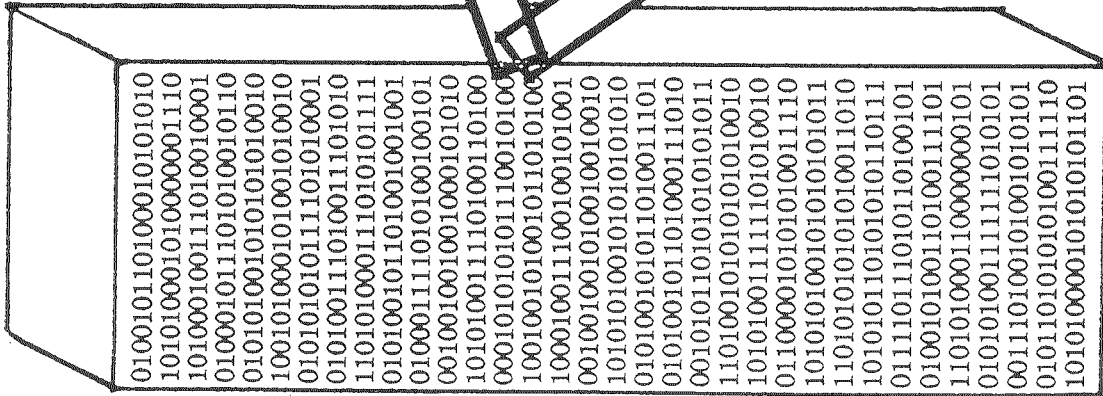


LARGE MONOLITHIC

MODULARIZED



Modularization Goals: Independence, Autonomy, Reuseability, Understandability...



What You Should Know About O-O

- ▼ The basic concepts have been around since the 1960s (*Simula* and *Smalltalk*), the 1970s (Parnas' article "On the Criteria to be Used in Decomposing Systems into Modules) -- if not since 5000BC (the Egyptian pyramids).
- ▼ "O-O is still a tower of babble."
Ed Yourdon at CASE World 2/18/92
- ▼ "I hate the way the word 'object' is being used in the industry. It's an internal technology, not a market opportunity."
Bill Gates, Microsoft (*Byte*, 12/92)
- ▼ Software's equivalent of *oat bran*:
 - few truly know if it's any good
 - few truly know if too much can kill you
 - everyone feels compelled to consume it

Nicholas Wybolt at CASE World 2/18/92
(continued)

What You Should Know About O-O (continued)

- ▼ Support for O-O is far behind that available for traditional methods.
- ▼ Managing an O-O project is very different from managing a traditional software development project.
- ▼ The fundamental issue in object-oriented systems development is: what are objects and how do they collaborate.

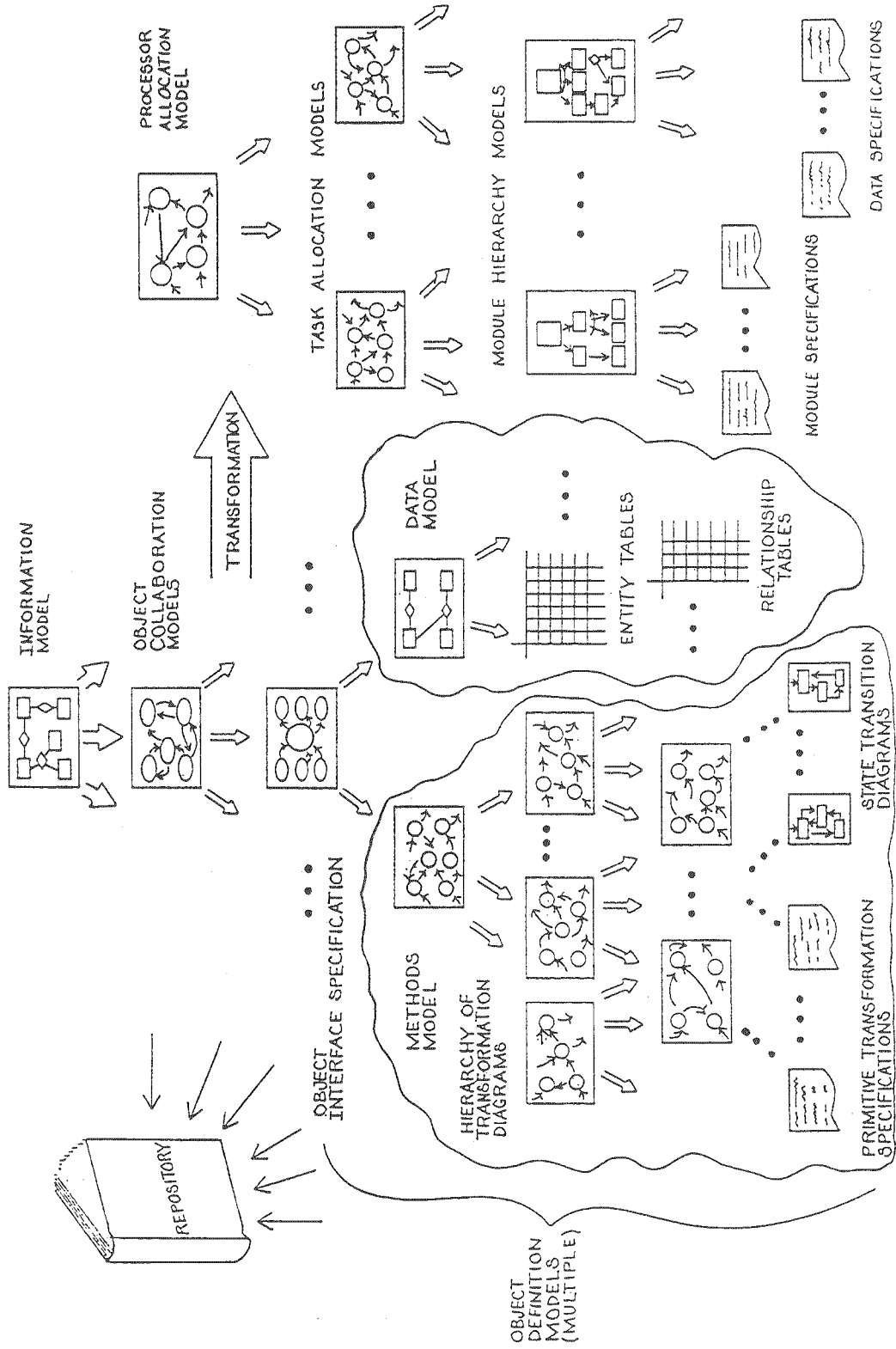
What is an Object???

- ▶ An object is an abstraction of a "real-world" entity or thing within a given application domain.
- ▶ Objects represent sets of things.
- ▶ Objects do work (they encapsulate "method").
- ▶ Objects store information (they encapsulate "data").
- ▶ Objects have "secrets" (they hide information).
- ▶ Objects are "building blocks" from which systems are built.
- ▶ Objects collaborate with other objects to perform some designated activity or function.
- ▶ Objects have "life-cycles."

(continued)

What is an Object?? (continued)

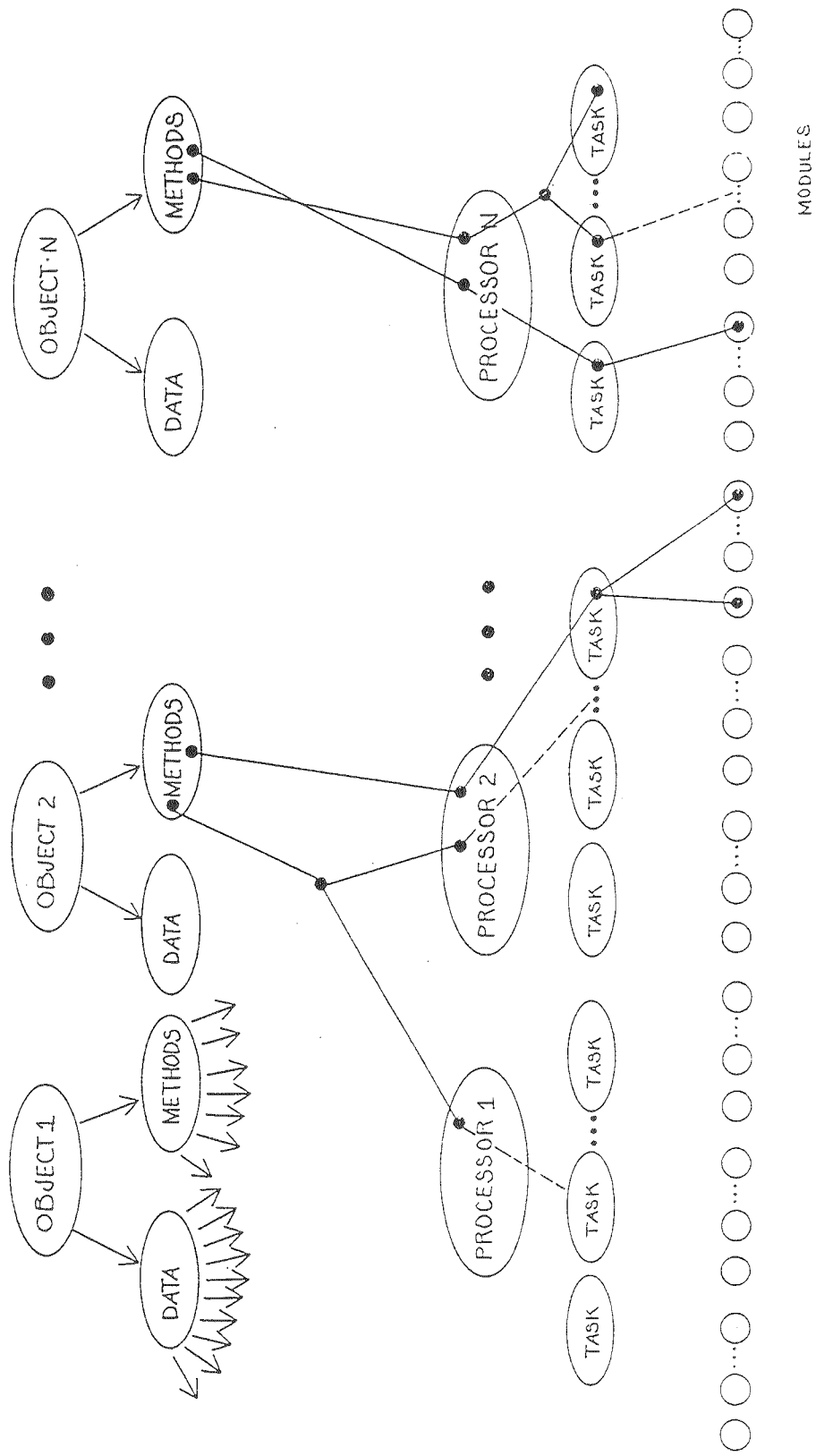
- ▶ Objects may belong to classes of other objects and may inherit their properties.
- ▶ Objects may be of widely varying types; they may be active or passive. They may even be degenerate!
- ▶ Objects communicate and coordinate with other objects by exchanging messages. Messages may be of various types.
- ▶ Objects may involve concurrency.



Object Collaboration Model

- ▶ Represents how objects "work together" for the purpose of performing some activity or activities within the application domain.
- ▶ Objects and Messages
- ▶ Synchronous Interfaces
- ▶ Asynchronous Interfaces
- ▶ Event-Driven Interfaces
- ▶ Transaction/Response Interfaces
- ▶ The System Level Event/Response List
- ▶ Executing the Object Interface Model

Object Allocation to the Implementation Model



Key Points

- ▼ The two fundamental aspects of object-orientation are:
 - The identification of objects, and
 - The specification of systems as collaborations of objects.

- ▼ The first is necessary for the second. Few formal techniques exist, however, to do it well and with confidence.

Key Points

- ▼ Establish a "knowledge bank" from which you can draw throughout the O-O analysis phase.
- ▼ The Knowledge Bank will include:
 - People
 - Documents (including interviews)
 - Models
 - Software
 - ... (be creative!)
- ▼ Establish the Knowledge Bank before you need it!

Establishing a Starting-Point

- Identifying Problem Domain Sources
- The Role of Application Domain Experts
- Resource Techniques
- Interviewing Techniques
- Classification of Resources
- Capturing Requirements
- Capturing "Business Rules"
- Requirements Definition Modeling

Establishing a Starting-Point

- ▶ Distinguish between sources which relate to the application domain vs. those which relate to the specification of a system.
- ▶ Most application domains have a body of reference materials (text books, practices and procedures, etc.).
- ▶ System specifications may be formal or informal.
- ▶ Interviews may be used to secure application domain knowledge or system specifications.
- ▶ A system synopsis is essential.
- ▶ System change requests are invaluable.

Establishing a Starting-Point (continued)

- ▶ Establish a resource roster.
- ▶ Identify and "credential" experts.
- ▶ Document key findings, policies, rules, etc. and seek expert concurrence.
- ▶ Recording and transcribing interview is very useful.
- ▶ Interviews should be directed towards gathering application domain "background" or system specification. Interviewer and interviewee should be sensitive to this distinction.
- ▶ Interviews may be "driven" by "what if" scenarios ("use cases").
- ▶ Look at printed forms -- they capture "business rules"

Key Points

- ▼ It is extremely useful to create traditional S/A models based on the collected information:
 - Context Diagrams
 - DFD's -- if they already exist
 - Data Models
 - E/R or STD models

- ▼ This is especially useful in re-engineering existing systems.

Key Points

- ▶ Context Diagram's external entities and flows will "suggest" objects.
- ▶ DFD's will suggest object methods (mini-specs).
- ▶ Data Models will suggest objects and attributes.
- ▶ E/R models will suggest objects AND methods.
- ▶ They're ALL important!

Information Analysis Techniques

- What's in a Name?
- FirstOrder Analysis
- Identification of Initial Concepts
- Second and HigherOrder Analyses
- Identification of Complex Concepts
- TwoDimensional and MultiDimensional Analyses
- Identification of Relationships

Key Points

- ▶ The goal of Linguistic Information Analysis is to identify:
 - Problem domain concepts, and
 - Relationships between those concepts.
- ▶ The concepts and relationships we identify may become objects (or other constructs) in O-O analysis.

Information Analysis Techniques

- ▼ The essence of the problem is to examine gathered information and identify concepts.
- ▼ First-order analysis involves performing simple word-frequency counts. Adequate supports exists for this technique. Use first-order analysis to identify an initial set of concepts.
- ▼ Second (and higher) order analysis involves phrase-frequency counts. **NO** support exists for this technique. Use higher-order analysis to identify complex concepts.
- ▼ An initial iteration will produce an initial set of concepts. This initial set of concepts can then be used to perform matrix analysis.

Information Analysis Techniques (continued)

- ▼ Matrix analysis will reveal relationships.
- ▼ This is an iterative process. Plan on several iterations before your first review.
- ▼ Information analysis should be supplemented with data, process and event analyses.
- ▼ Final selection based on object selection criteria.

Information Analysis Techniques (continued)

- ▼ Support tools include: word processors, "phrase processors," concordance generators and permuted index generators.

Moving into Object-Oriented Analysis & Design

- What's Missing?
- Elements of ObjectOriented Methods
- Establishing Requirements Traceability at the Object Level
- How Analysis & Design "feedback" to the Information Model
- Refining the Information Model

Getting Started

- Process Maturity Self Assessment
- Tactical vs. Strategic Change
- Recognizing the Problem Before Providing the Solution
- Identifying Strategic Objectives
- Managing a Changing Process