SOFTWARE DEVELOPMENT

Paradigms And Processes

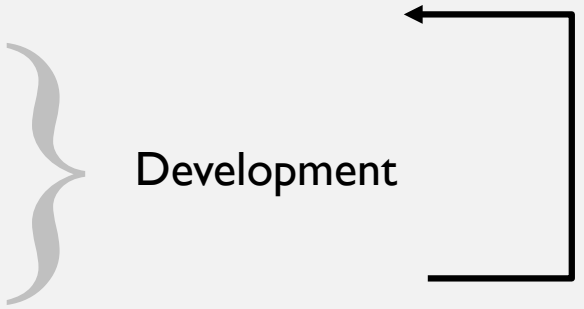
Delroy A. Brinkerhoff

THE SOFTWARE LIFE CYCLE

- "A life cycle is a collection of phases that divide an effort into several more manageable and controllable subordinate efforts."
  - Sinan Si Albir, UML In A Nutshell
- "The choice of the software development process has a significant influence on the project's success. The appropriate process can lead to faster completion, reduced cost, improved quality, and lower risk. The wrong process can lead to duplicated work efforts and schedule slips, and create continual management problems."
  - Jaak Jurison, Software Project Management: The Manager's View, p. 12

SOFTWARE LIFECYCLE PHASES

- Requirements
- Analysis
- Design
- Implementation
- Validation
- Maintenance
- Retire



ANALYSIS

- Emphasis is placed on the problem, not the solution
- Creates an external model of the problem/application domain by abstracting essential aspects or features
- Results should be understandable by customers, domain experts, and implementers
- Language/system independent
- Verifies that the requirements are sufficiently complete to proceed
- Called OOA when applied to the object model

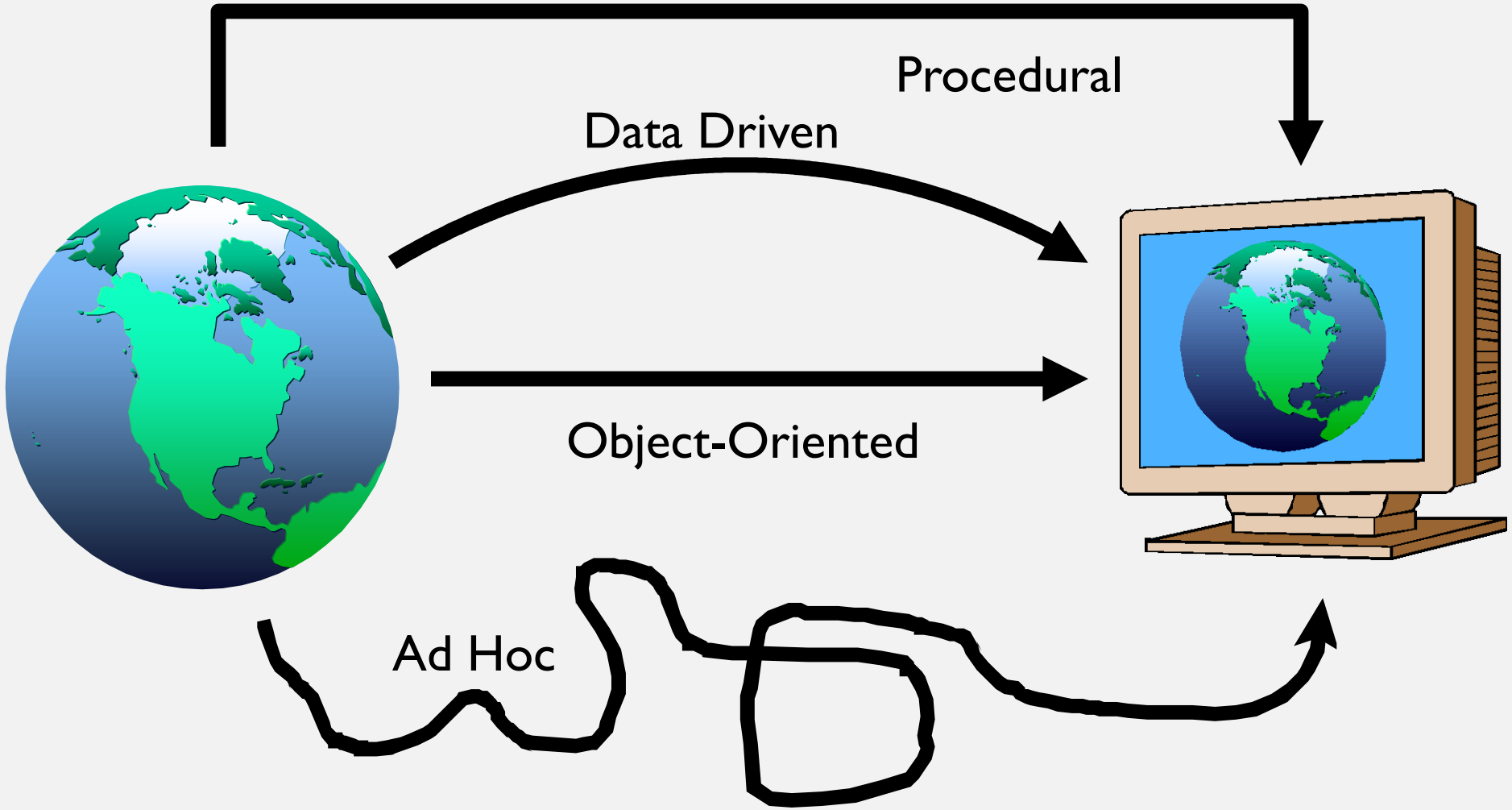
DESIGN

- Creates a solution architecture or framework by transforming the analysis results into a form that can be implemented
- Forms a bridge between analysis and implementation
  - Adds data structures and other implementation features
  - Describes user interface
  - Describes data management
  - Describes task management
- Maintain language/system independence
- Called OOD when applied to the object model

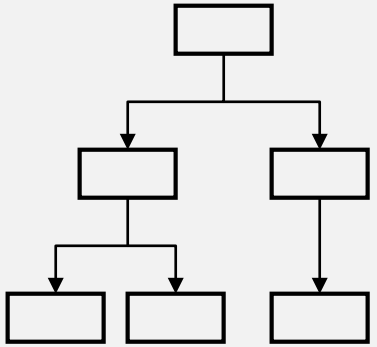
IMPLEMENTATION / PROGRAMMING

- Creates or forms a usable tool or system
- Forms the most significant part of a project's deliverables
- Final result may be represented as
  - Hardware
  - Software
  - Combination
- Called OOP (programming) when the implementation is in software based on the object model

THE DEVELOPMENT GOAL



THE PROCEDURAL MODEL



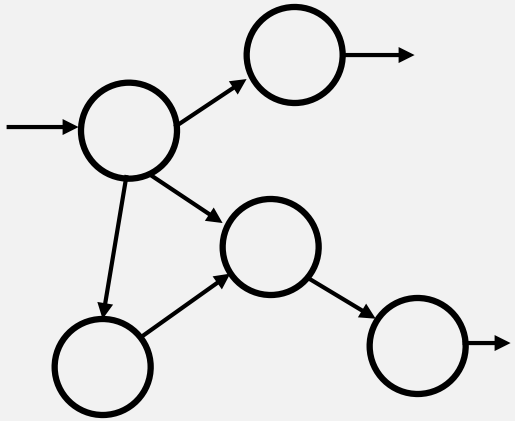
```
updateInventory( )
{
}

printReport ( )
{
}
```

Semantic Gap
(analysis & design)

Analysis
and
Design
Results

Implementation

THE DATA FLOW MODEL



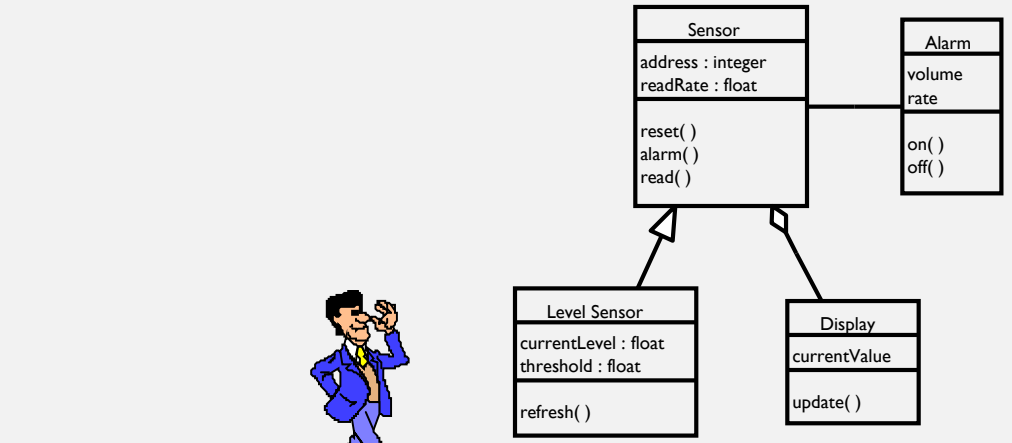
Analysis
and
Design
Results

Paradigm shift
(programming)

```
updateInventory( )
{
}

printReport ( )
{
}
```

Implementation

THE OBJECT-ORIENTED MODEL




```
class Sensor { }

class LevelSensor : public Sensor { }

class Alarm { }

class Display { }
```

O-O

Analysis and Design Results

O-O

Implementation

JAVA VS. C++

- Java is a pure object-oriented language, all features (variables, constants, and methods) are contained in a class
- C++ is a hybrid language – supports both the procedural and object-oriented paradigms
  - Features may be contained in a class
  - Features may be independent of any class
  - Some library components are object-oriented