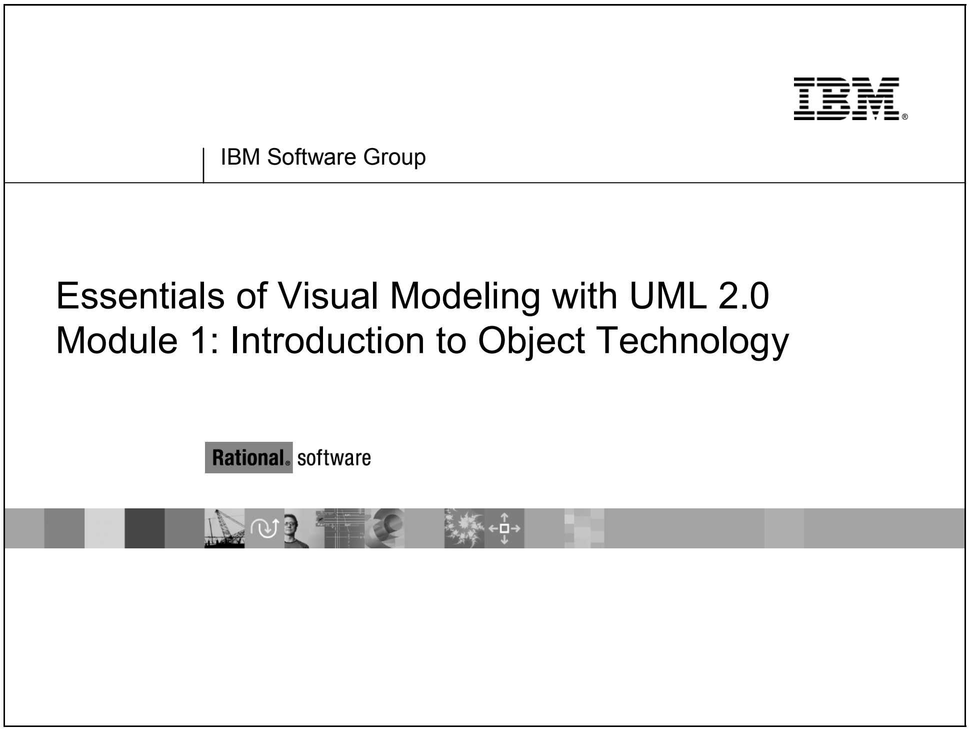
▸ ▸ ▸ Module 1
Introduction to Object Technology



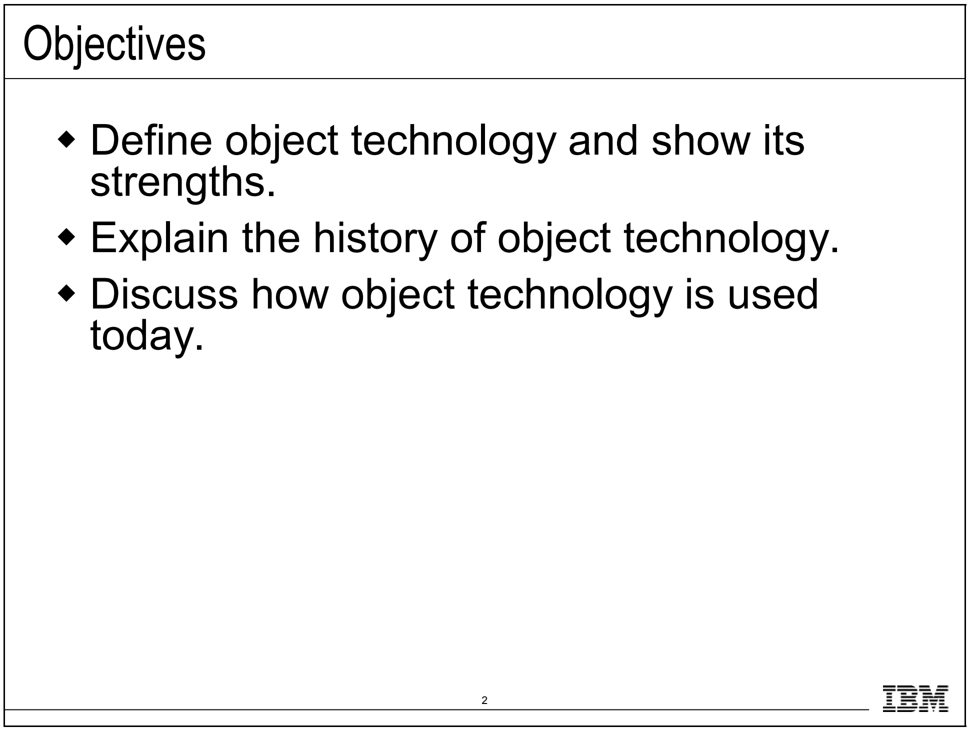
IBM Software Group

Essentials of Visual Modeling with UML 2.0
Module 1: Introduction to Object Technology

Rational. software

Topics

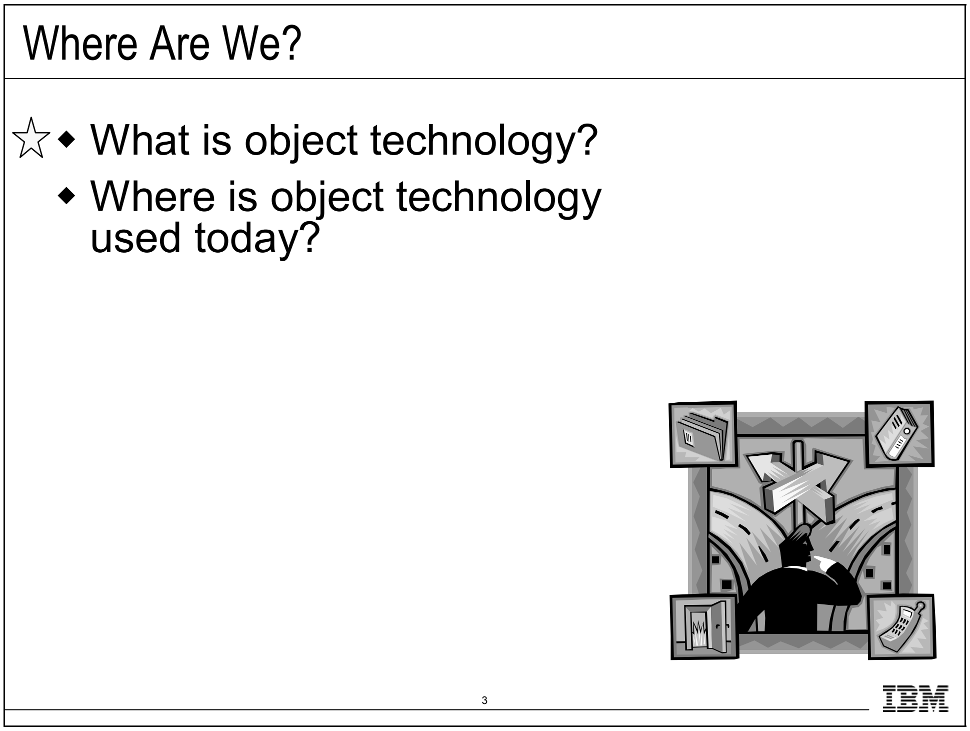
Objectives



Objectives

- Define object technology and show its strengths.
- Explain the history of object technology.
- Discuss how object technology is used today.

Where Are We?

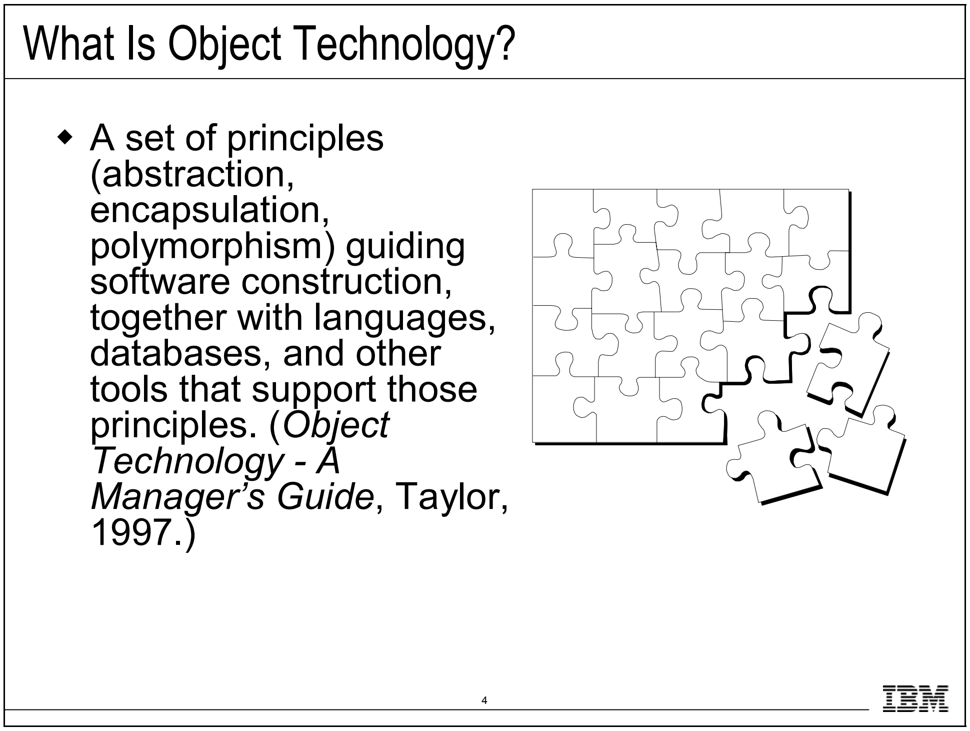


Where Are We?

- ☆ What is object technology?
- Where is object technology used today?

3

What Is Object Technology?



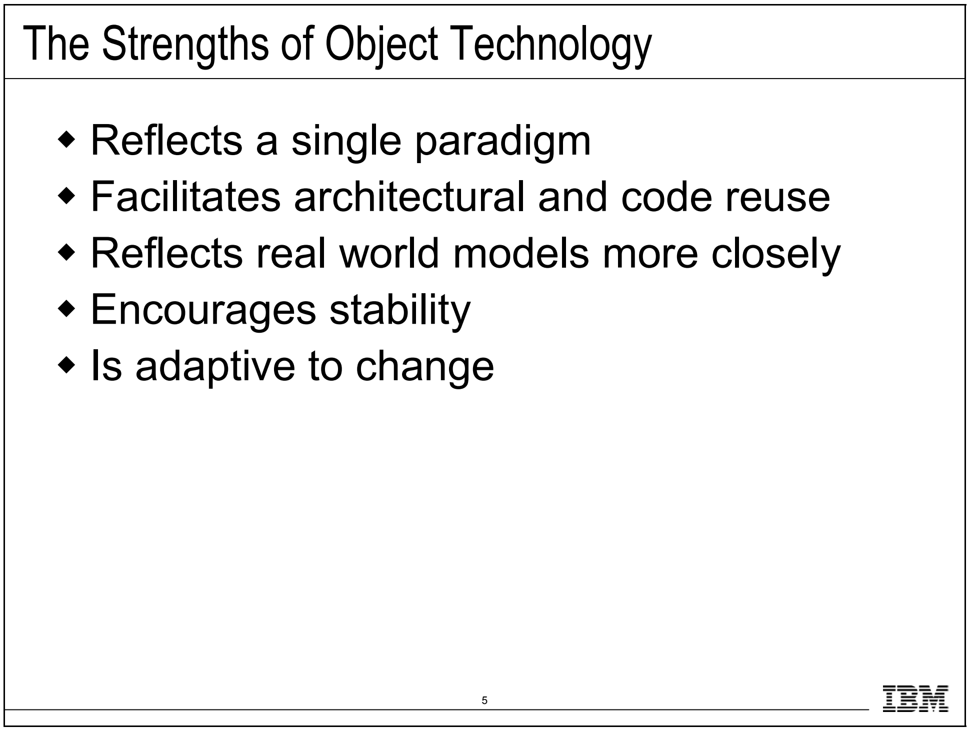
What Is Object Technology?

- A set of principles (abstraction, encapsulation, polymorphism) guiding software construction, together with languages, databases, and other tools that support those principles. (*Object Technology - A Manager’s Guide*, Taylor, 1997.)

4

- Object technology is used for creating models that reflect a specific domain using the terminology of the domain.
- Models created using object technology should be easy to create, change, expand, validate, and verify.
- Systems built using object technology are flexible to change, have well-defined architectures, and have the opportunity to create and implement reusable components.
- Models created using object technology are conveniently implemented in software using object-oriented programming languages.
- Object technology is not just a theory, but a well-proven technology used in a large number of projects and for building many types of systems.
- Successful implementation of object technology requires a method that integrates a development process and a modeling language with suitable construction techniques and tools. (*UML Toolkit*, Eriksson and Penker, 1997.)

The Strengths of Object Technology



The Strengths of Object Technology

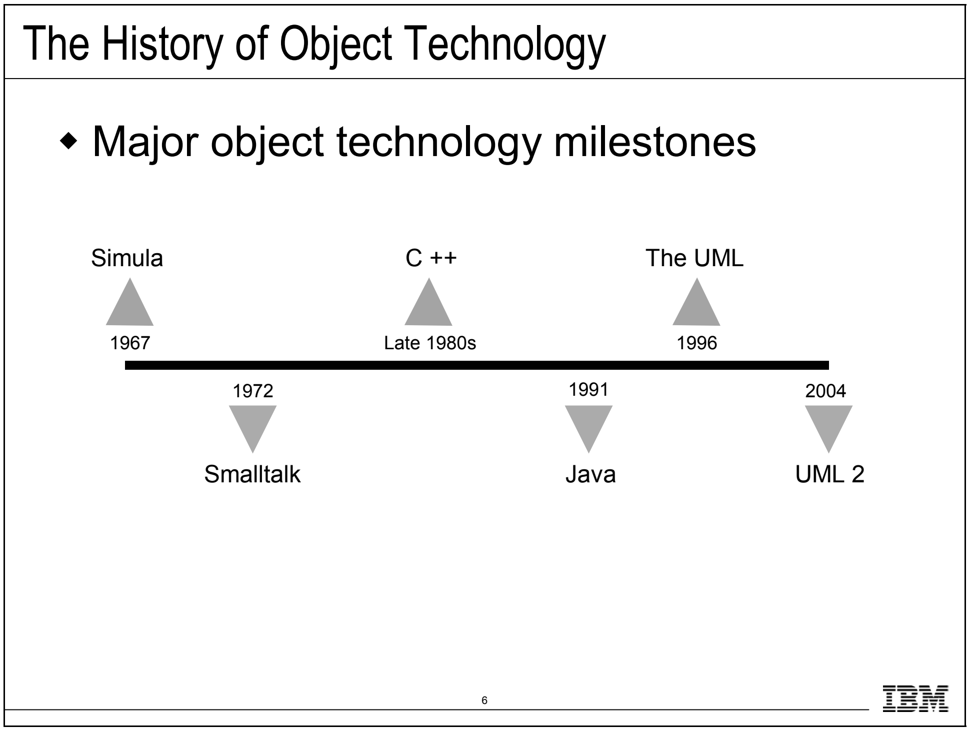
- Reflects a single paradigm
- Facilitates architectural and code reuse
- Reflects real world models more closely
- Encourages stability
- Is adaptive to change

5

So what's the big deal about object technology? Why has the software industry made such a noticeable shift to object-oriented languages like Java? Much of the answer has to do with the strengths of object technology. There are some people who believe that object technology is a passing fad and will quickly go away, while others believe that object technology is the silver bullet that answers all software development problems. The truth is that the answer probably lies somewhere in between. Object technology is a powerful and challenging way to develop software. The result of this hard work can be a software system that:

- Reflects a single paradigm. It provides a consistent language that can be applied for both system and business engineering.
- Facilitates architectural and code reuse by clearly articulating the major components and the critical interfaces between them.
- Reflects real world models more closely. The objects themselves often correspond to phenomena in the real world that the system is to handle.
- Is more stable, a change to the system can be localized to a small part of the system.
- Is adaptive to change -a small change in requirements does not mean massive changes to the system.

The History of Object Technology



The History of Object Technology

- Major object technology milestones

Simula 1967

1972 Smalltalk

C ++ Late 1980s

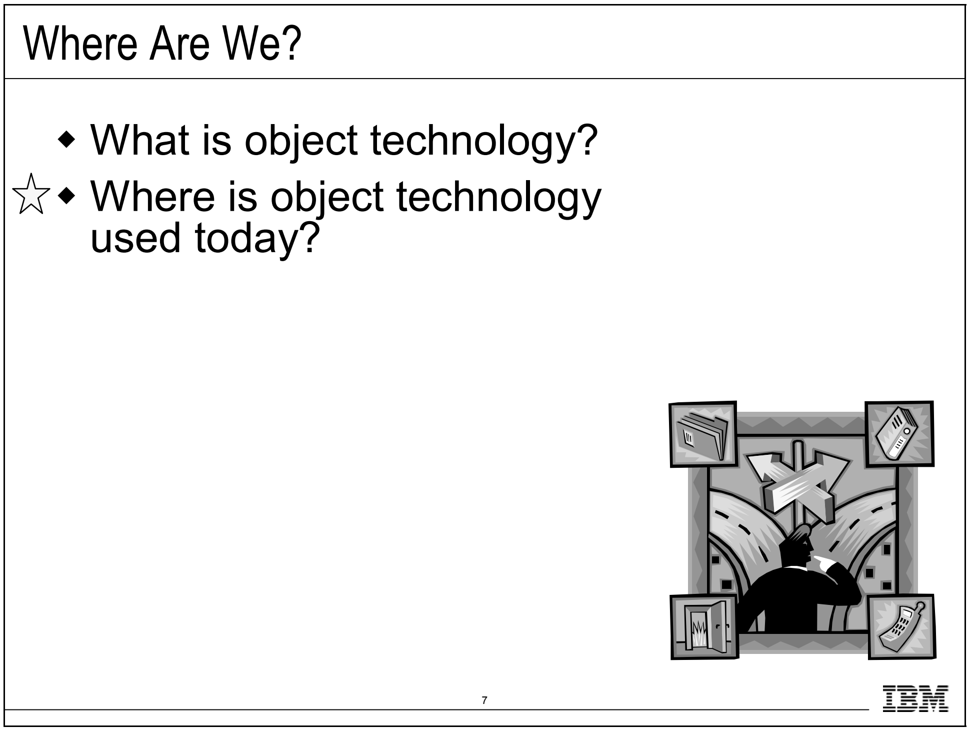
1991 Java

The UML 1996

2004 UML 2

- Object technology is not a new idea. It has been around for over 30 years. Below is a brief history of the major milestones in the history of object technology.
- In 1967, Simula was designed and became the first language to use objects and classes.
- In 1972, Alan Kay and others at Xerox PARC created Smalltalk whose roots were tied to Simula. In 1980, Smalltalk became the first commercial release of an object-oriented programming environment.
- Bjarne Stroustrop, the originator of the C language, released C++ to the public in the late 1980's. C++ was not an entirely new language, but an extension of C abilities.
- In 1991, James Gosling created a language named Oak that was the predecessor to Java. It was created because his development team at Sun Microsystems was writing software for information appliances. He found that C++ was too complex and insecure for the job. Therefore, he created the Oak language.
- UML 2.0 replaces UML version 1.4. Released in 2004, the Object Management Group (OMG) developed two complementary specifications called the Infrastructure and Superstructure specification. The Infrastructure defines the foundational language constructs required for UML 2.0 and serves as an architectural foundation. The Superstructure defines the user level constructs. Together, they constitute a complete specification for the UML 2.0 modeling language.

Where Are We?

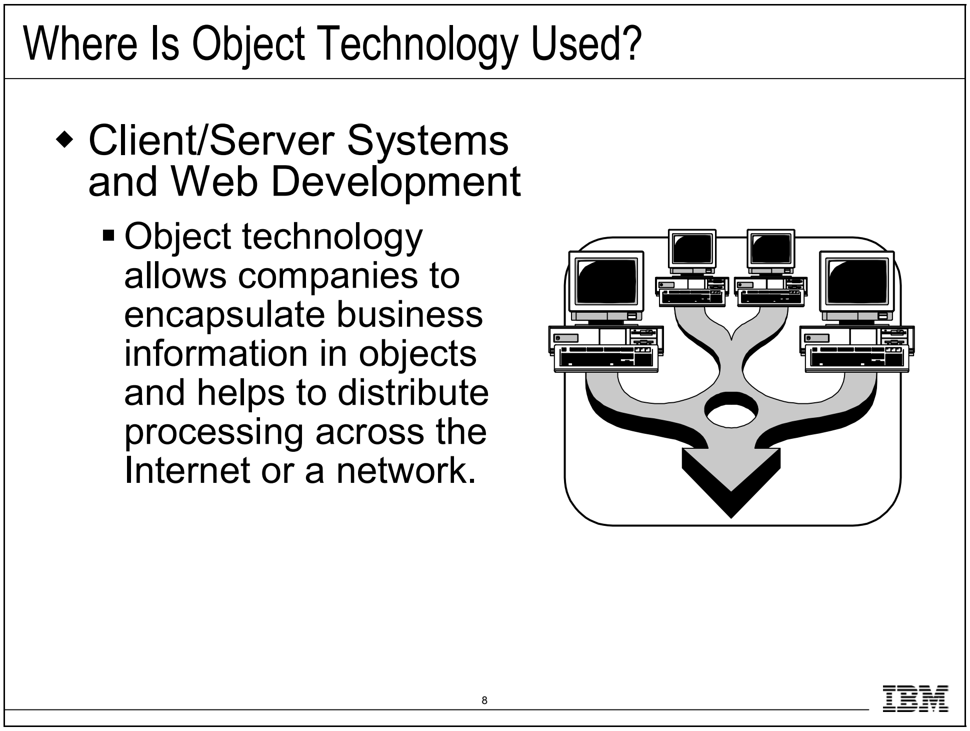


Where Are We?

- What is object technology?
- ☆ Where is object technology used today?

7

Where Is Object Technology Used?

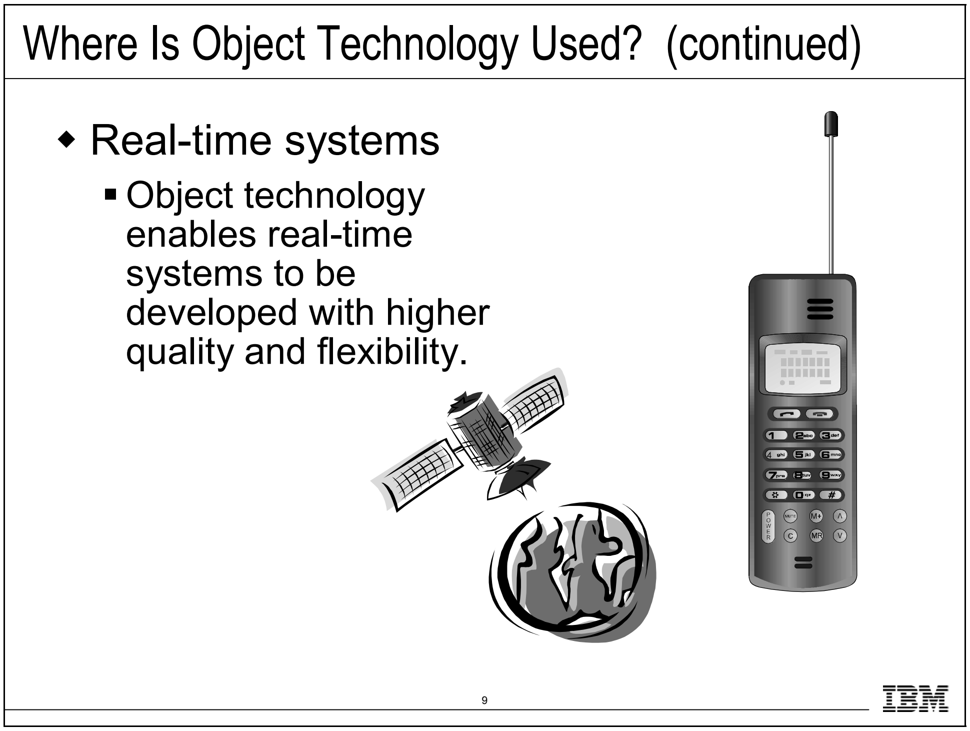


Where Is Object Technology Used?

- Client/Server Systems and Web Development
  - Object technology allows companies to encapsulate business information in objects and helps to distribute processing across the Internet or a network.

Object technology allows companies to encapsulate business information in objects and helps to distribute processing across the Internet or a network. Enterprise development environments such as Sun's J2EE and Microsoft's .Net technologies use objects as the basis for their technology.

Where Is Object Technology Used? (continued)



Where Is Object Technology Used? (continued)

- Real-time systems
  - Object technology enables real-time systems to be developed with higher quality and flexibility.

9

Object technology is used in the following industries and software applications:

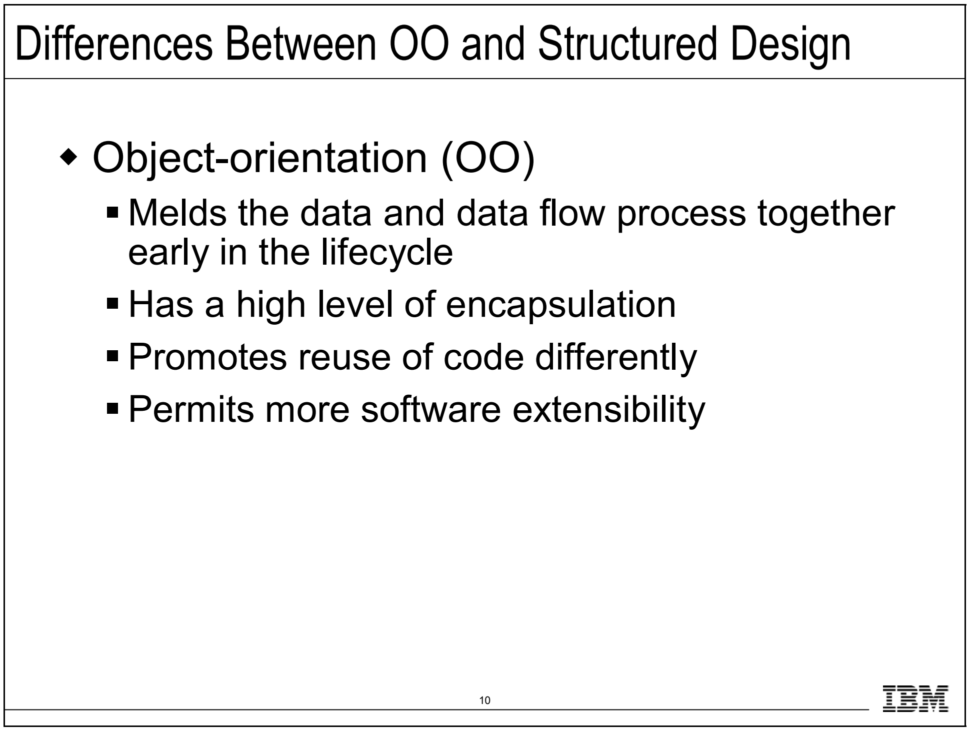
Telecommunications – circuit switching systems, wireless systems, transmission systems, satellite systems, fault management, call/connection control, and protocol development.

Data communications – LAN hub switches, multi-protocol routers, packet switching & frame relay, ATM switching systems, connection control, node management, fault management, and protocol development.

Defense and aerospace – command and control systems, missile control systems, defense simulators, cockpit control systems, air traffic control systems, target tracking, distributed computer modeling, man machine interface control, communication modes and control, redundancy management.

Industrial control – office equipment, factory control systems, multi-processor high speed printers, automotive systems, PWB and IC water fabrication, device control, system management, distributed process control, machine position control, bandwidth allocation.

Differences Between OO and Structured Design



Differences Between OO and Structured Design

- Object-orientation (OO)
  - Melds the data and data flow process together early in the lifecycle
  - Has a high level of encapsulation
  - Promotes reuse of code differently
  - Permits more software extensibility

10

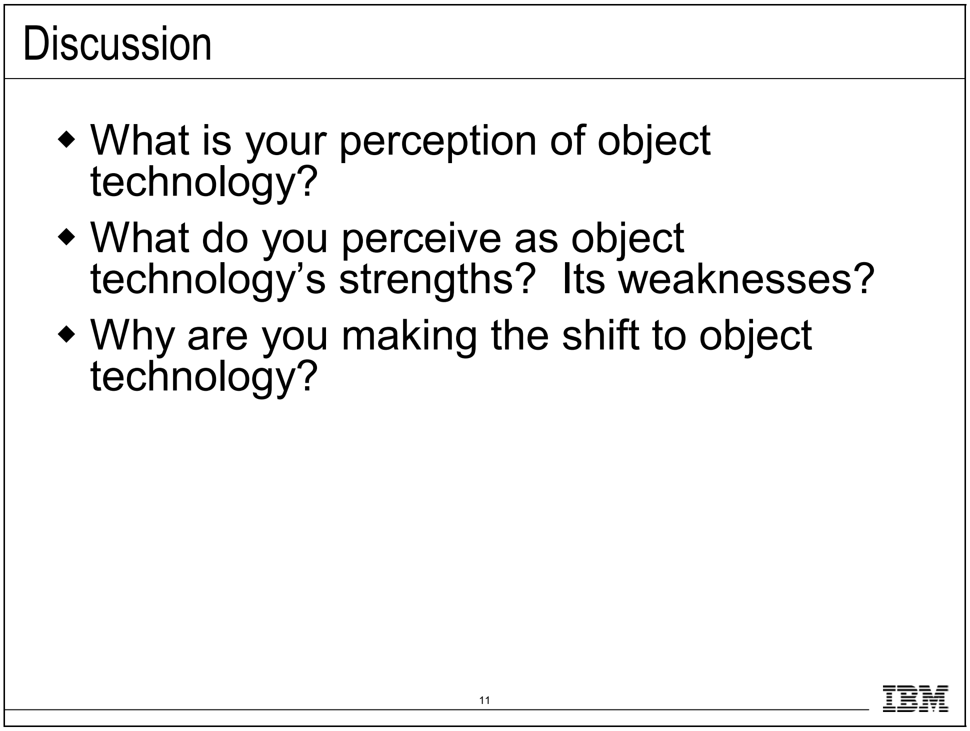
In the world of structured design, there has always been an uneasy relationship between the data model in an entity-relationship diagram and the dataflow diagram. The dataflow process and the data meet in some places, but in not others, they miss each other altogether. Object-orientation (OO) melds the data and dataflow process together early in the lifecycle. In object-orientation, concern yourselves with defining the static and dynamic views of the system (Jones, p.65).

Object-orientation has a very high level of encapsulation associated with it. Data, operations, and entire classes can be encapsulated. Structured programming relies upon data structures, sophisticated algorithms, and elaborate relationships between procedure and data (Jones, p.65).

Object-orientation promotes reuse of code at the class level rather than at the level of the individual subroutine (Jones, p. 66).

The goal of extensible software is to share a solution that most closely fits the problem. By doing this, you can ensure that a small requirements change doesn't require major modifications to your system. Because object orientation is built using classes that are abstractions of actual business objects, OO techniques come closer to permitting software extensibility than structured design.

Discussion



Discussion

- What is your perception of object technology?
- What do you perceive as object technology's strengths? Its weaknesses?
- Why are you making the shift to object technology?

11