Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (01223) 515010**
INTERNATIONAL: **+44 1223 515010**
FAX: **+44 1223 359779**
E-MAIL: **apm@ansa.co.uk**

Training

ANSAwise - Object Technology for Information Systems

Chris Mayers

Abstract

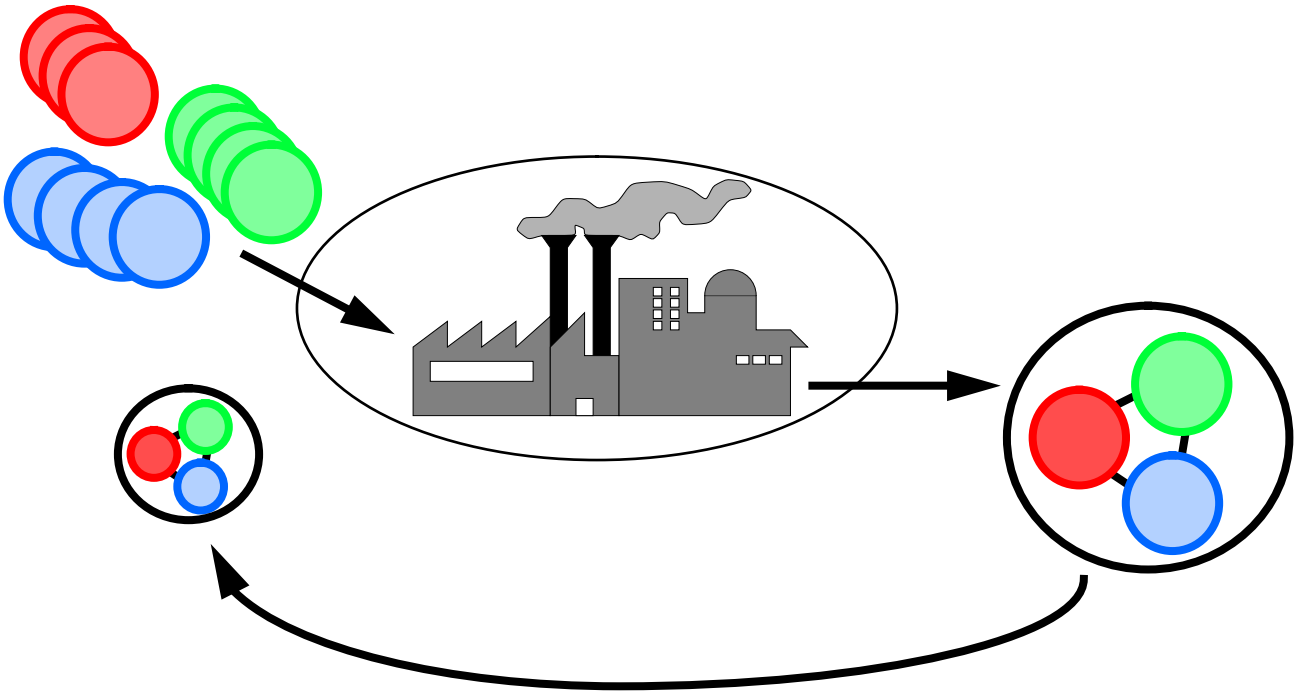
Organizations will be aware of object technology, and have a picture of the benefits it can bring, but may be unaware of the state of the market, and the pitfalls.

This module of the ANSAwise training programme discusses “objects in general”, compares how the ideas of encapsulation, polymorphism, and inheritance are applied in various object technologies, and explains why distributed objects require a specific approach to be successful. It is a variant of APM.1350, aimed at a general IT management audience rather than a software engineering audience.

APM.1478.01 **Approved** 4th May 1995
Briefing Note

Distribution:
Supersedes:
Superseded by:

Object Technology for Information Systems



In this session

- *Explain the business motivation for object technology*
- *Introduce the basic ideas of object technology*
- *Review the issues that arise when using object technology*

Problems with existing software

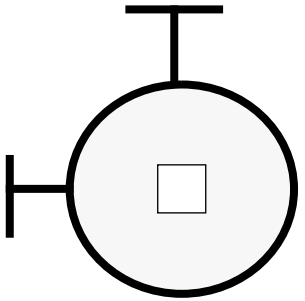
- *Changing software is difficult*
- *Changing software can have unanticipated side-effects*
- *Upgrading software to new versions is difficult*
- *The consequences are*
  - Application backlogs
  - Low productivity and administrative overheads
  - Low quality

Why object technology?

- ***Object technology enables systems to be built out of modular software components***
- ***These systems offer***
  - **Greater productivity through reuse**
  - **Better quality through proven components**
  - **Lower costs through reduced maintenance overheads**
- ***Object technology facilitates the building of distributed systems***

Encapsulation - the key to reuse

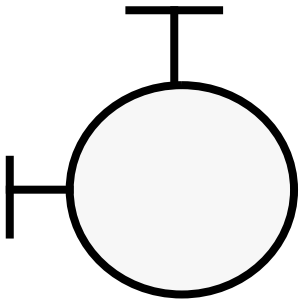
- *Data needs to be encapsulated together with the program...*



- *...forming an object*

Encapsulation for objects

- ***Objects are encapsulated***
  - **every object provides a service via interfaces**
  - **the interface is public; the implementation is private and hidden**
  - **encapsulation forms a boundary; the only access to an object is via its interfaces**



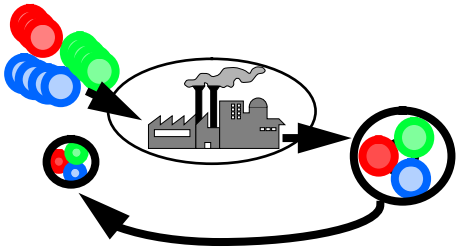
Polymorphism

- *There are more kinds of objects than kinds of operations...*

	Save	Edit	Print	Delete	...
Report					
Spreadsheet					
Diagram					
Graph					
Diary					
Plan					
...					

- *... so each object should handle each kind of 'polymorphic' operation*

Reusable software



- *Writing reusable software is more time-consuming*
- *Reusing software efficiently requires special software tools*
- *Reuse is primarily a strategy issue - a management issue*
- *The best form of reuse is to buy rather than build*

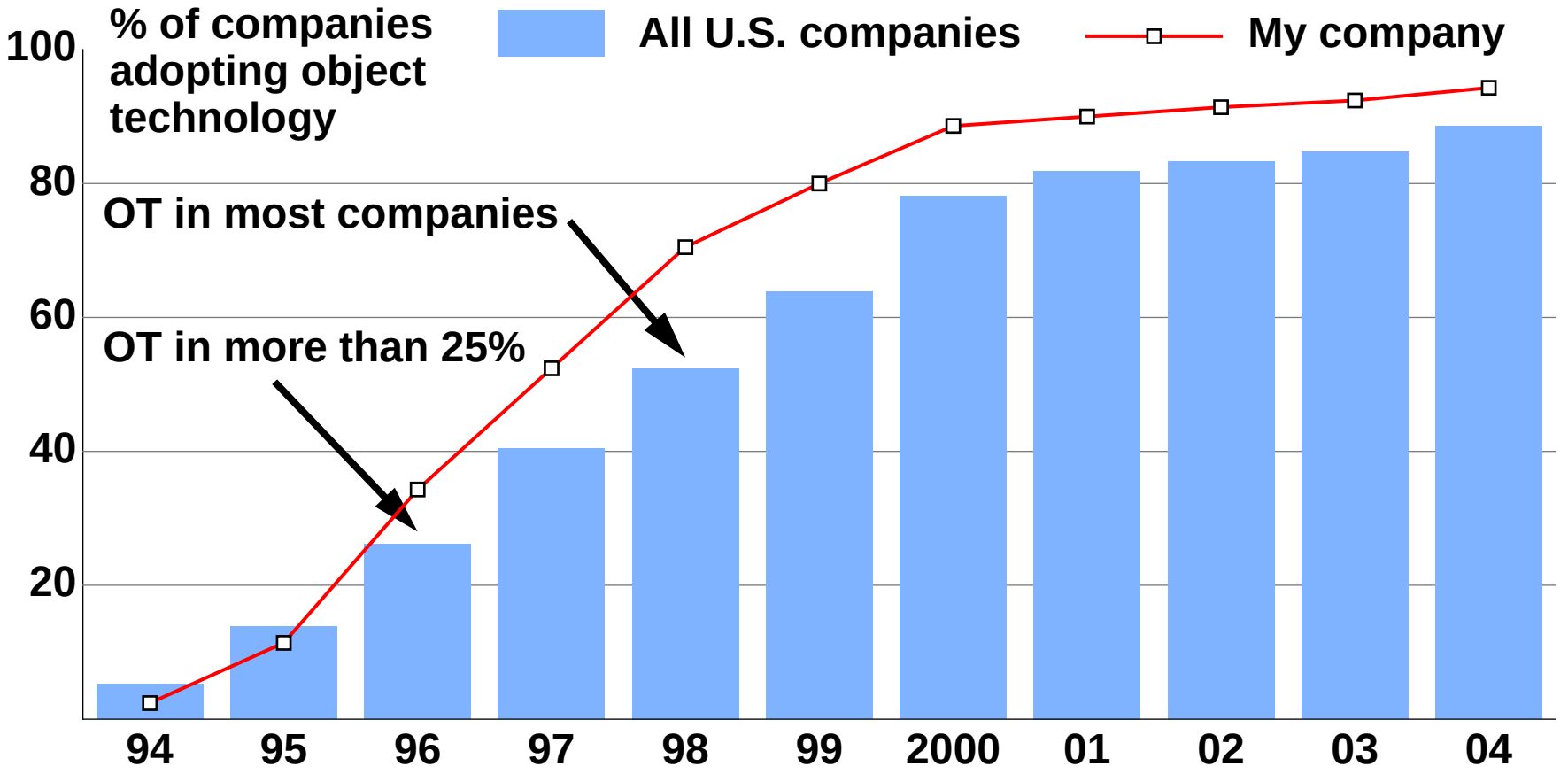
Business Objects and Technology Objects

- ***Business objects are derived from a top-down analysis of your organization's goals***
  - **derived from a strategic enterprise model**
  - **embodied in an operational model**
  - **... with associated business rules**
- ***Business objects might be customers, products, orders...***
- ***Technology objects are acquired bottom-up***
- ***In the future, both business and technology objects may be bought and sold***
- ***supplied by third parties in horizontal and vertical markets***

Influence of object technology

- *Now affecting all areas*
  - operating systems
  - programming languages
  - databases
  - user interfaces
- *Products and standards are emerging in all of these*
  - but at different levels of maturity...
  - ...with patchy support from development methods and tools
  - ...and with important technical differences
- *Not all of these products and standards are 'open'*
  - this poses a challenge to those building systems with object technology

The Object Technology Market



Who is using object technology now?

- ***Software pioneers***
  - **the City**
  - **large organizations**
  - **systems integrators**
  - **software product developers**

Object technology for information systems

- *Even though object technology is immature, it must be considered*
  - when selecting software suppliers
  - when integrating systems
  - when implementing appropriate applications
- *Use object principles when designing systems*
  - even if the systems are implemented with conventional technology

Barriers to object technology

- *Lack of support for large-scale development projects*
- *Shortage of programmers with object skills and experience*
- *Few project managers with object technology project experience*
- *Inadequate metrics for object technology projects*

Learning object technology

- ***Object technology does require a new method of thinking about system design***
  - **a way which is more natural for business analysts and users**
- ***Object programming languages take longer to master than conventional ones***
- ***Object technology may require learning new software development tools as well as new methods***
  - **to avoid becoming overwhelmed, plan the learning process**

Summary

- ***The business benefits for object technology can seem intangible and indirect***
- ***Object technology will be in widespread use within a few years***
- ***Object technology is a good fit with client/server technology***