

6 Enterprise computing and Corba

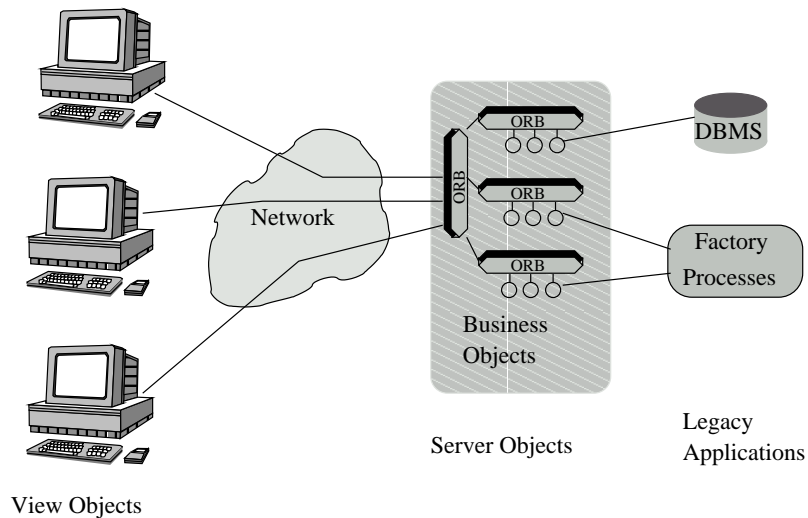
Main Points

- What are computers used for?
- Three tier models
- CORBA

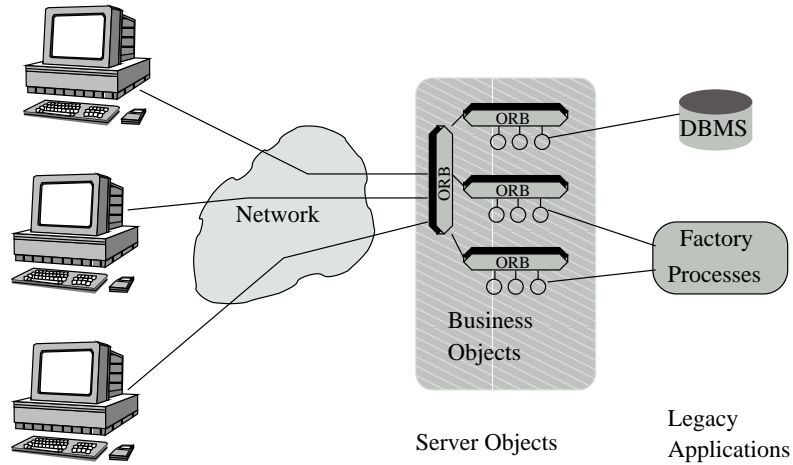
6.1 Computers in Business

- The hard job in commercial computing is not writing word processors...
- ... but to integrate the various business activities - selling, buying, managing, coordinating production.
- The key asset in business computing is the data that the business has built up over the years, generally held in a database management system.
- Business computing therefore has to integrate the front end activities carried out by people with the data held in the databases.
- Changes in the data and requests from the front end must be interpreted according to the *business logic*.

6.2 Three Tier Models and the Web



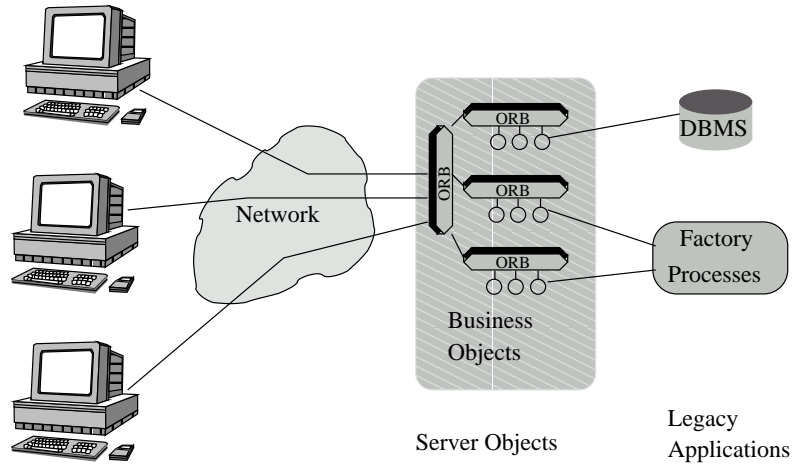
6.2.1 The view objects - Javascript, XML and friends



View Objects

- The front end is the user interface, which will provide various *views*, depending on function.
- Functions include views in the warehouse of the stock control system, business overviews for decision support systems, Point Of Sale views for the tills.
- These can be a mixture of xml, html, javascript and java in a web browser.
- Or they can be fully blown applications talking directly back to the server objects

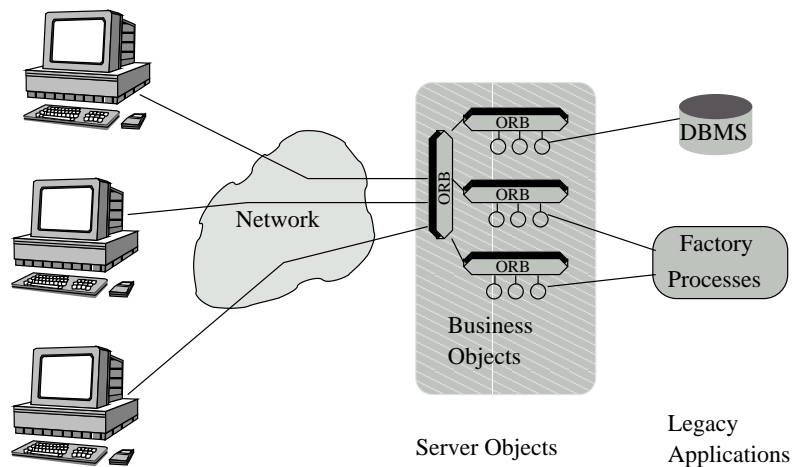
6.2.2 Server objects and middleware



View Objects

- The server objects encapsulate the *business logic* - how the business uses its data.
- This is subject to tweaking and change as the business evolves - ie a high maintenance activity.
- Object and particularly component based approaches are most useful here.
- The CORBA object standards intend to provide *business objects* to support the implementation of business logic in a component framework.

6.2.3 Legacy systems and software wrappers



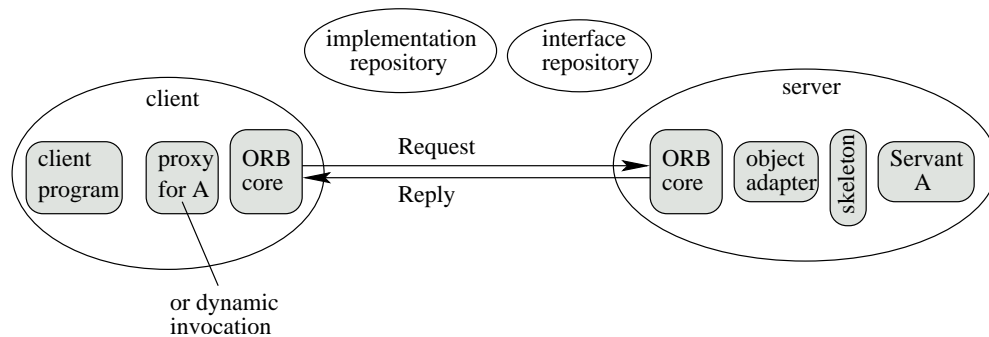
View Objects

- Most companies have a substantial investment in their existing software. . .
- . . . and this software already does its job
- To integrate this *legacy* software into new systems requires software wrappers that can speak to the new systems.

6.3 CORBA

- CORBA is a middleware design to allow application programs to communicate with each other irrespective of their programming language.
- Standardised by the Object Management Group (OMG), an industrial consortium of well over 100 companies.
- Many different implementations, which can all interoperate.

6.3.1 The CORBA Object Model



client program calls a method in the remote *servant* program A.

proxy for A marshalls the arguments in invocation requests and unmarshalls replies, and is generated from the IDL. Available at compile time.

ORB core implements the standardised communication systems, and provides services for converting remote references to strings etc.

Object Adapter bridges between the ORB and the target language, providing dispatch for method invocations, remote references and activation control

skeletons are in the target language and do the unmarshalling of arguments and marshalling of results.

Servant for A does the actual work.

Implementation repository activates registered servers on demand and locates running servers, using the object adapter name.

Interface repository provides information about registered IDL interfaces, so as to enable dynamic invocation of reflection in Java.

6.3.2 The Interface Definition Language - IDL

Modules CORBA IDL provides modules which function in a similar fashion to Java packages, providing scope control mechanisms for names.

Interfaces Interfaces are similar to Java *interfaces*, providing collections of methods offered by an object. IDL supports *inheritance*

Attributes can be declared in IDL - the compiler generates accessor methods (get/set) automatically.

Methods Method descriptions are similar to Java, except that parameters can be tagged as

in value passed into the method

out value returned from the method, in addition to the return value of the call

inout value passed in and returned.

Exception Methods can raise exceptions, which may return values

Primitive types Such as byte, short, int, long, char, double

Constructed Types sequence (variable length list), array, struct (collections of values of different types), enumerated (set of named integer values) unions (choice of values, depending on discriminator enumerated type tag)

Remote Object references Objects are not passed by value. Instead, references to objects can be returned and passed around. These references are interpreted by the local communications modules (the Object Request Broker ORB).

6.3.3 IDL Example

```
// From file Person.idl

struct Person {
    string name;
    string place;
    long year;
};

interface PersonList {
    readonly attribute string listname;
    void addPerson(in Person p);
    void getPerson(in string name, out Person p);
    long number();
}
```

6.3.4 Language Bindings

- In normal system languages - c, c++, java - these are quite straight forward, as long as the programmer is aware of the semantics of the IDL.
- But some of the semantics need special support.
- Consider `void getPerson(in string name, out Person p);`
- The Java equivalent is

```
void getPerson(String name, PersonHolder p);
```

where *PersonHolder* contains an *instance* of the returned *out* value of the *Person* argument.

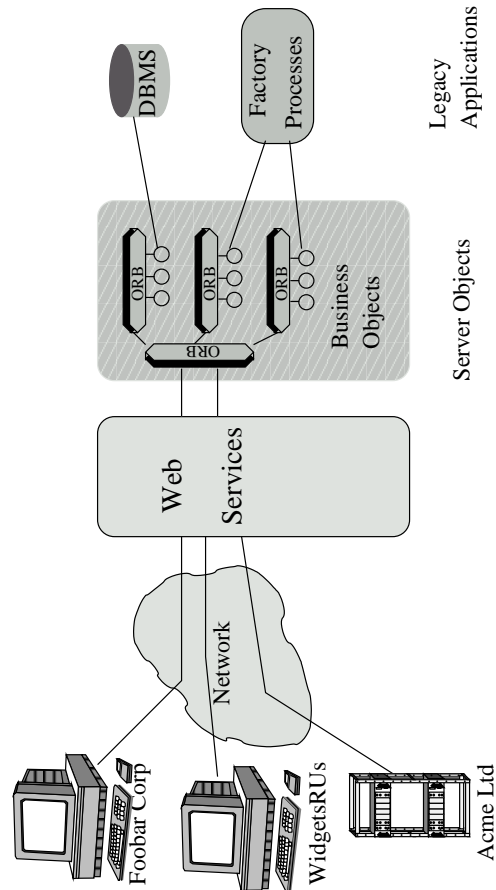
6.3.5 CORBA Services

- CORBA includes specifications for a number of services:
 - Naming Service** uses a *naming context* to lookup a name. The naming contexts can be linked, so that a name actually points to another naming context, allowing hierarchical names.
 - Event and Notification Service** provides for event management, including pattern matching for notification.
 - Security Service** provides for management of authentication and access control, audit trails etc.
 - Trading Service** allows services to be located by description, rather than name.
 - Transaction and Concurrency Control Service** implements transactional mechanisms to provide concurrency control and ACID semantics to operations
 - Persistent Object Service** allows objects to be stored in passive form when not required and activated on demand.

6.3.6 Transport Issues - IIOP

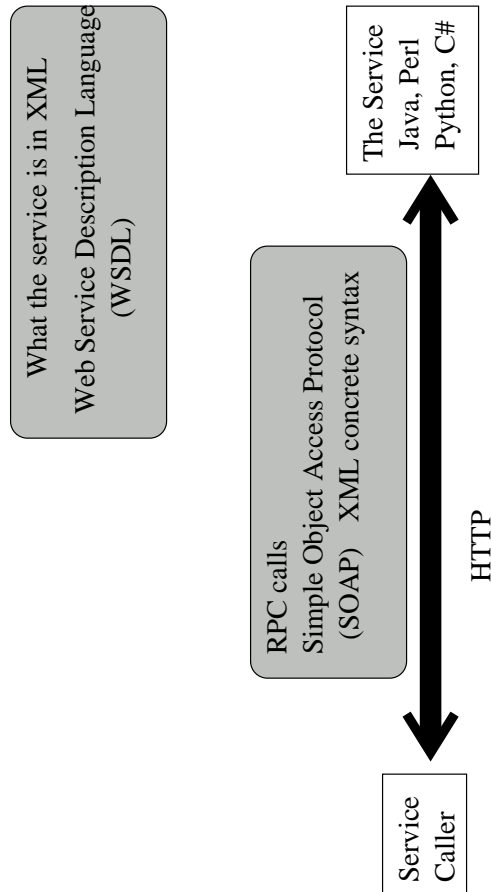
- There is a standard protocol for use between ORBs - the General Inter-ORB Protocol, GIOP.
- The implementation almost universally used is the Internet Inter-ORB Protocol, IIOP.
- GIOP specifies the concrete syntax for data placed into the byte stream, the Common Data Representation or CDR.
- IIOP specifies the layout of messages, and the standard layout for remote object references.

6.4 Web Services - Business to Business



- HTML/javascript and friends are good for rendering interfaces to people...
- ... but how should business computers talk to each other?
- CORBA has never bridged the business to business gap for various reasons.
- Can XML based solutions bridge the gap?

6.4.1 SOAP, WSDL and XML



- *Web Services* are business to business RPC systems based on XML and HTTP
- Messages are described using an XML variant called SOAP.
- The concrete syntax is thus ascii tags and content.
- The interface definition language is Web Services Description Language (WSDL)
- The IDL is thus (very verbose) XML
- More detailed notes can be found at Vladimiro's Internet Technologies notes

6.5 Summary

- The majority of programmers in the commercial world create bespoke solutions to manage business processes.
- Three tier models of business solutions provide flexibility yet control the complexity of the business logic.
- CORBA provides a middleware framework in which to implement the business logic.