

**DISTRIBUTED COMPUTING MEETS THE
INFORMATION SUPERHIGHWAY**

A J Herbert

Rapporteur: John Dobson



Distributed Computing meets the Information Superhighway

Andrew Herbert

ANSA / APM

APM.1556.03

Approved

External Paper

25th August 1995



From Global Network to Global Operating System

- **Observation:**
 - The World Wide Web is creating a “Uniform” information space and a platform for electronic commerce
- **Proposition 1:**
 - Distributed systems technology can overcome engineering weaknesses in the current Web
- **Proposition 2:**
 - Distributed systems technology can extend the capabilities of the current Web
- **The next frontier:**
 - Active content and semantics-driven information processing

APM.1556.03

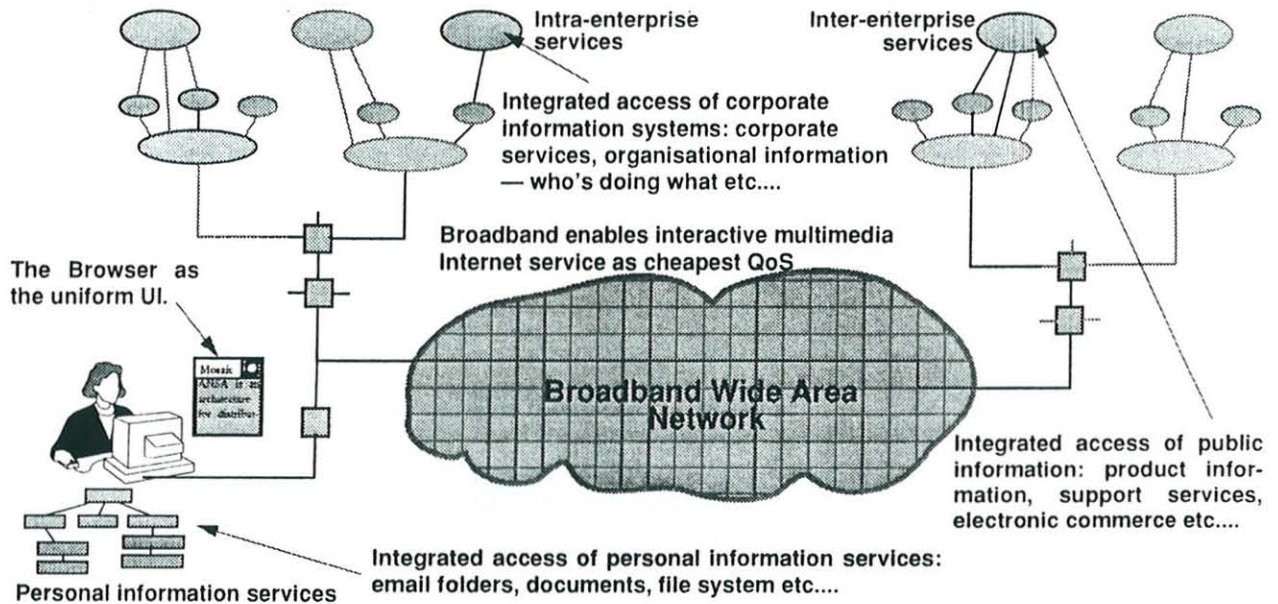
Approved

External Paper

25th August 1995



Creating a uniform information space



Technology requirements for the uniform information space

Well understood

- Presentation (= Browsers + HTML, postscript etc.)
- Creation (= Authoring tools)
- Efficient protocols, interactive multimedia
- Extensible front ends (making new functionality available)
- Extensible back ends (transactional services)
- Dependability (availability, security, integrity)
- Navigation tools (finding the information)
- Administration tools (managing the services)

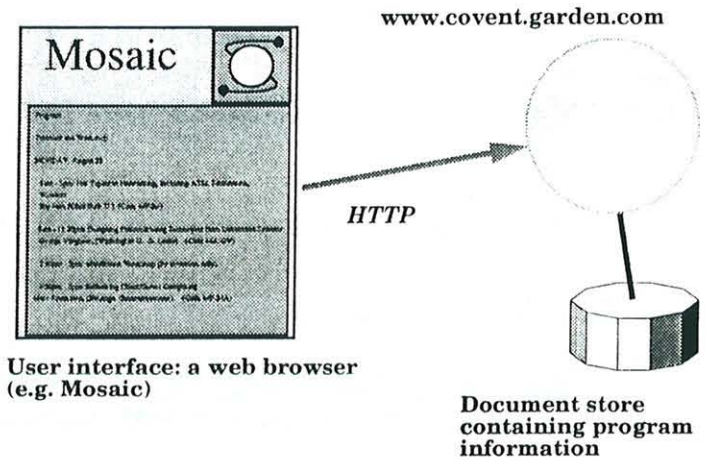


Poorly understood



An example "transactional" application

- Scenario: booking theatre tickets for multiple events.
- Requirements
 - Browsing program information
 - Choice of seats & dates
 - On-line selection of preferences
 - Confirmation of booking by server
 - Ability to change booking
 - Single payment on confirmation of booking by client

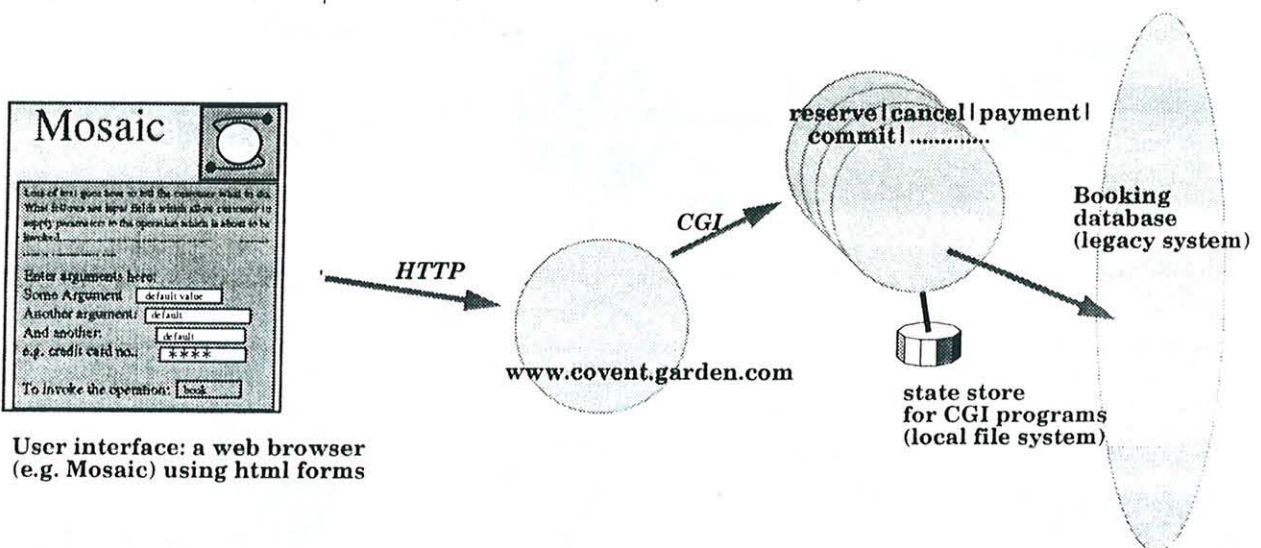


User interface: a web browser (e.g. Mosaic)

Document store containing program information



How it's done today.....



User interface: a web browser (e.g. Mosaic) using html forms

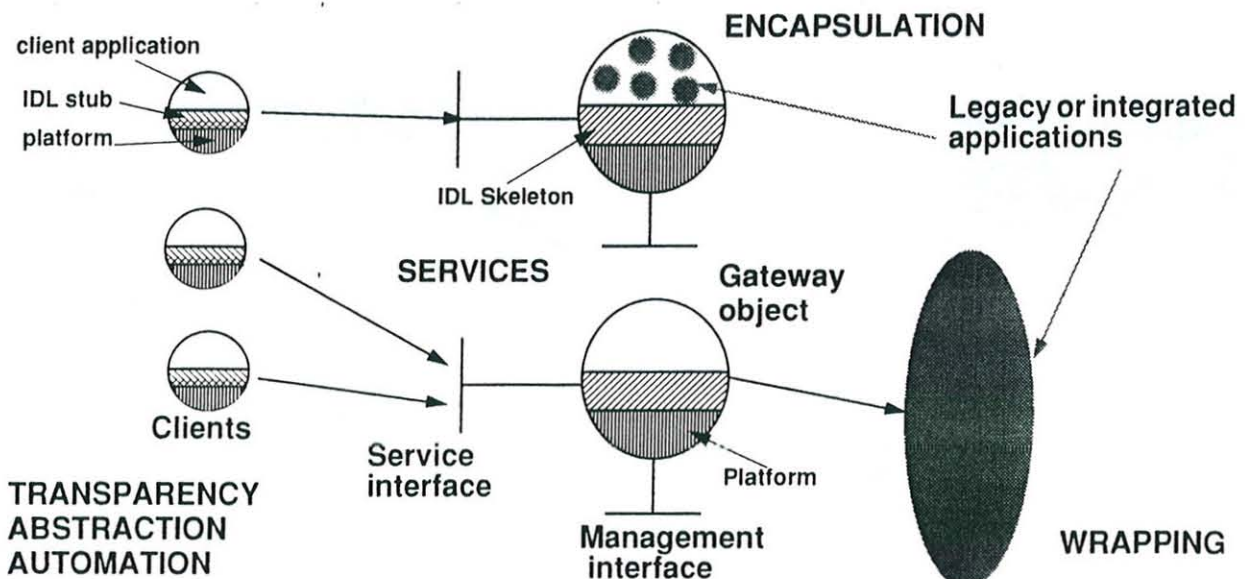


A Hacker's Paradise

- The CGI allows HTTP servers to communicate with other programs
 - It is hard (error prone) to write the code to unmarshal the parameters to the CGI programs (no tool support)
 - CGI Program is "forked" for each interaction, no "session" structure
- CGI driven by HTML forms
 - Care is needed to make sure the HTML form and corresponding CGI program are consistent (both in numbers and "types" of parameters).
 - For most browser, customisation of the user interface is limited by what can be displayed in browsers using HTML forms technology — agent technology (e.g. Java) allows greater customisation (see demonstration).



The Solution: Distributed Object Technology



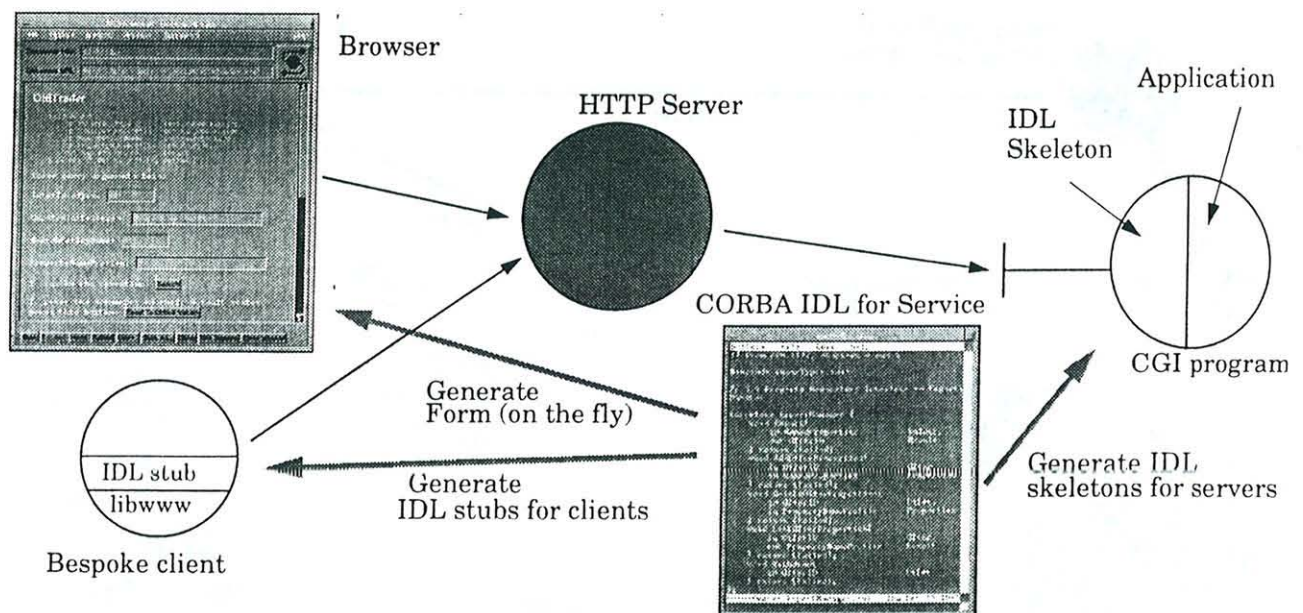


Benefits of Object Technology

- Access services through defined (in IDL) interfaces
 - stub compilers can generate IDL stubs and skeletons which abstracts the programmer from the underlying protocols and API.
 - calling a remote service a procedure call; the underlying protocols and the internals of the platform are hidden
- We can slide in infrastructure services transparently
 - transactions, authentication, replication, migration
- We can apply object management functions
 - everything is an object
 - life cycle, event handling, repository, property, query, trading
- We can substitute alternative protocols and data formats



ANSWeb Phase 1 - A Demonstrator





E.G. HTML Form Generation

```
interface Echo{
    string Echo(in string Src);
    void Sink(in string Src);
    string Source(in long Length);
    string Reverse(in string Src);
};
```

Stub Compiler generates this HTML form

```
<head>
<TITLE>Input for Echo</TITLE>
</head>
<BODY><H1>Input for Echo</H1>
<HR>

<H2> Operation Echo</H2>
<FORM METHOD="POST" ACTION="http://socrates.ansa.co.uk:8080/cgi-bin/Echo">
<P>Enter arguments here:<P>
<INPUT NAME="Operation" TYPE=hidden VALUE="Echo_Echo">
<FORM string Src: <INPUT SIZE=10 NAME="Src"> <P>
To invoke Echo_Echo: <INPUT TYPE="submit" VALUE="Echo_Echo"><P>
</FORM>
<HR>

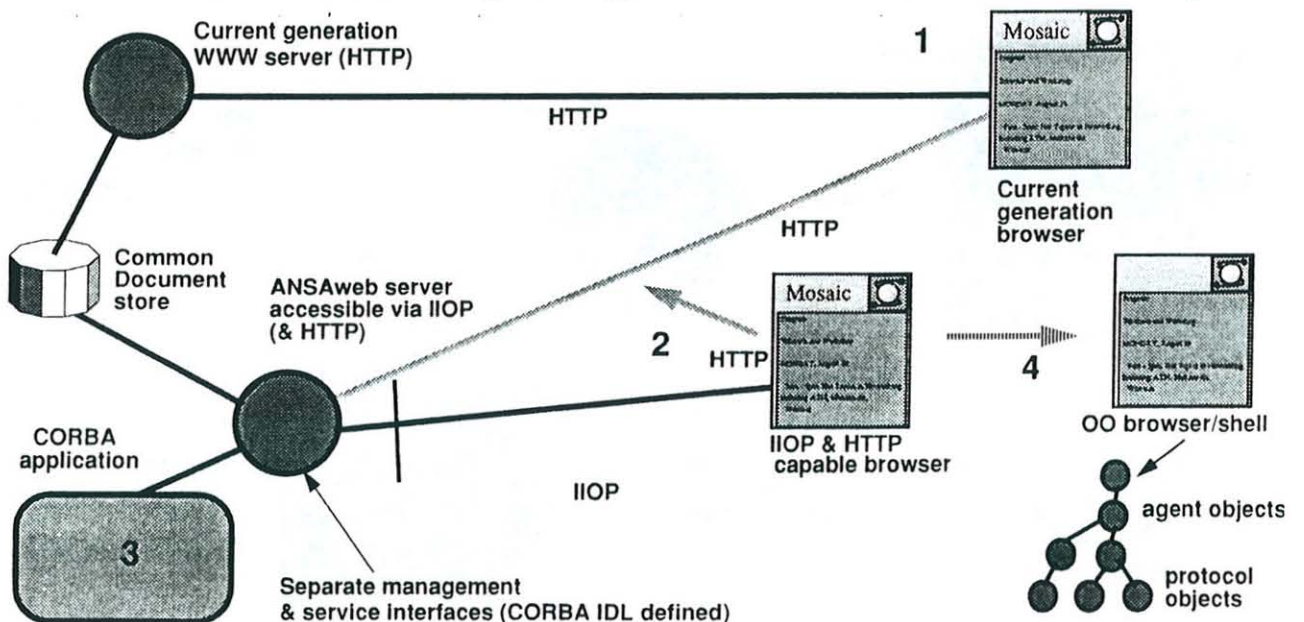
<H2> Operation Sink</H2>
<FORM METHOD="POST" ACTION="http://socrates.ansa.co.uk:8080/cgi-bin/Echo">
<P>Enter arguments here:<P>
<INPUT NAME="Operation" TYPE=hidden VALUE="Echo_Sink">
<FORM string Src: <INPUT SIZE=10 NAME="Src"> <P>
To invoke Echo_Sink: <INPUT TYPE="submit" VALUE="Echo_Sink"><P>
</FORM>
<HR>

<H2> Operation Source</H2>
<FORM METHOD="POST" ACTION="http://socrates.ansa.co.uk:8080/cgi-bin/Echo">
<P>Enter arguments here:<P>
<INPUT NAME="Operation" TYPE=hidden VALUE="Echo_Source">
<FORM long Length: <INPUT SIZE=10 NAME="Length"> <P>
To invoke Echo_Source: <INPUT TYPE="submit" VALUE="Echo_Source"><P>
</FORM>
<HR>

<H2> Operation Reverse</H2>
<FORM METHOD="POST" ACTION="http://socrates.ansa.co.uk:8080/cgi-bin/Echo">
<P>Enter arguments here:<P>
<INPUT NAME="Operation" TYPE=hidden VALUE="Echo_Reverse">
<FORM string Src: <INPUT SIZE=10 NAME="Src"> <P>
To invoke Echo_Reverse: <INPUT TYPE="submit" VALUE="Echo_Reverse"><P>
</FORM>
</BODY>
```

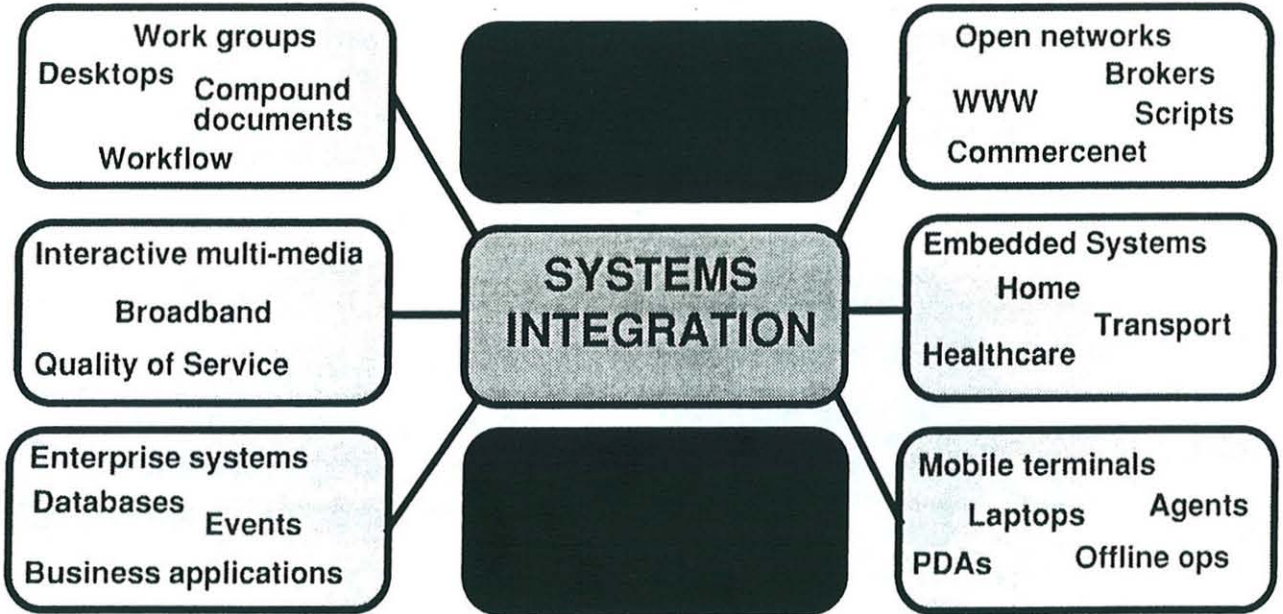


ANSWeb Phase 2: Migrating the web to Distributed Objects

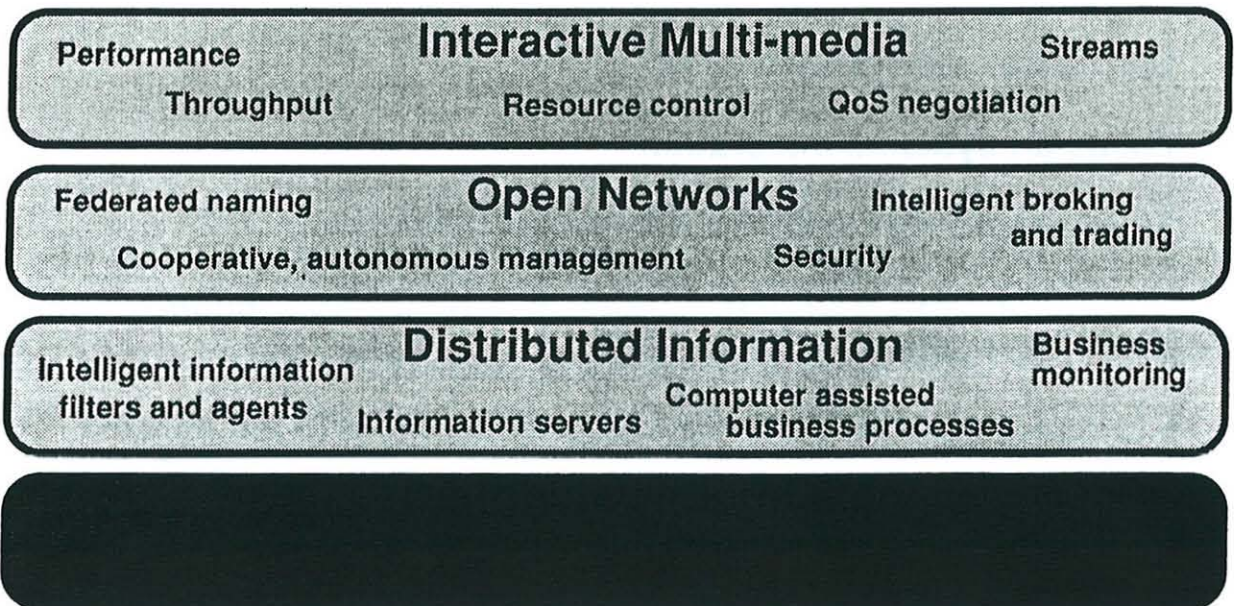




The Future of Software - Systems Integration

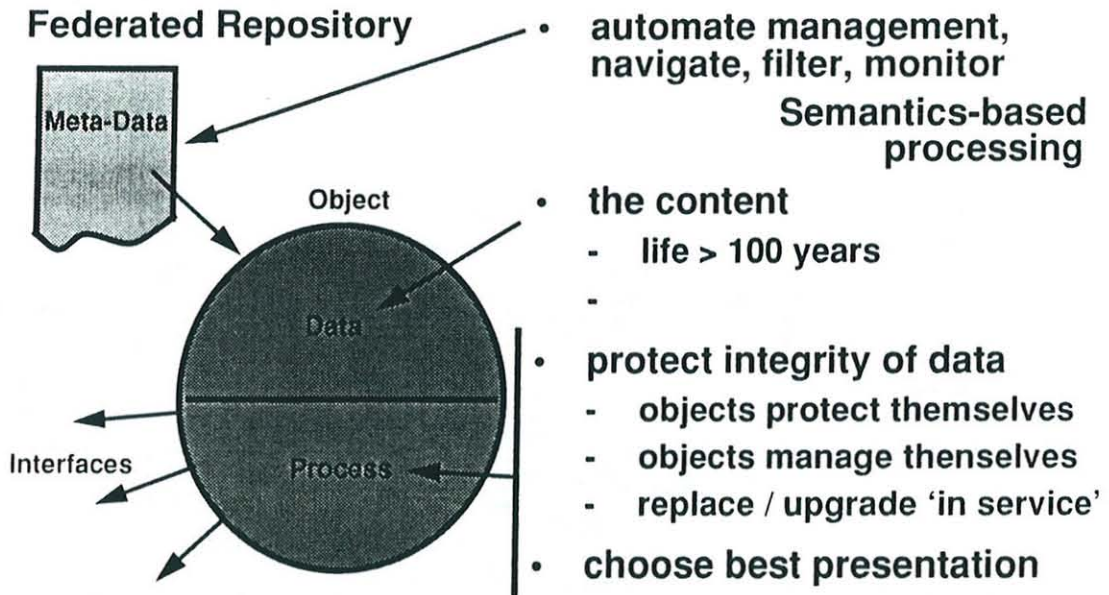


New Requirements





Active Content



Conclusions

- Distributed object interconnect
- Focus on systems integration
- Small distributed systems within vast networks
- Federations of autonomous systems rather than a global distributed operating system
- Interactive multimedia bring QoS management to the Internet
- Active content is the applications paradigm
- The contenders: OMG with CORBA, MicroSoft with OLE 2.0 & (D)COM

DISCUSSION

Rapporteur: John Dobson

Lecture Two

Mr Dobson asked whether some of the management aspects of the world-wide web could benefit from the work on intelligence in telecommunications networks (IN) and telecommunications management networks (TMN). Dr Herbert replied that it could, and the fact that work on IN and TMN is being more and more influenced by distributed object technology simply reinforced the theme of his lecture.

Dr Lesk observed that traditional distributed systems required high availability from their components. This was not observed in the web, where things typically did not work all the time. Dr Herbert replied that the distributed operating system community claimed that it could transparently mask such things as local unavailability. But in fact users often knew how to compensate for system reliability problems, so that distributed operating system designers should learn how to utilise user recovery mechanisms. This would require the ability to load protocols dynamically and to employ active agents. Professor Randell asked whether the designers of the web ignored well-known techniques and partial solutions coming from distributed operating systems. Dr Herbert replied that the main strategy ignored by the web designers was that of splitting a layer in a structure into two halves (a top half and a bottom half) and slotting something in between. Professor Martin observed that in the telecommunications community it was clear that management and service provision were becoming different disciplines, resident in different enterprises. By contrast, Dr Herbert's approach seemed to intertwingle service provision and management. Dr Herbert replied that objects will increasingly be able to manage themselves. The effect of this on market structure would remain to be seen.