Objects and the World Wide Web

Nigel Edwards, Owen Rees

Abstract

Information services are being deployed in the internet now. However, there are a number of technical problems which need to be solved to take full advantage of this global network for deploying commercial information services. Two of these problems are explored in more detail: naming and extensibility.

Within the internet, the World Wide Web has highlighted a number of problems that involve naming issues. This presentation explores some of these problems and looks at concepts and technology which is being developed to solve them.

Being able to extend the functionality of basic services is crucial for promoting market differentiation and innovation. This presentation describes how distributed object technology, particularly CORBA, can be applied to facilitate extensibility. Also described, is work in progress on using script technology to extend clients and server functionality dynamically.
Objects and the World Wide Web

Nigel Edwards

Owen Rees
Objects and the World Wide Web

Objective: “Develop the concepts and infrastructure needed to manage information services in the global network”

- Review of technical challenges
- The problems in the Web today
- The solutions being developed
- Applying CORBA technology to extend the functionality of the World Wide Web (WWW)
- Dynamically extending the functionality of WWW
Business using a broadband network

Commercial information services

Support services: availability, reliability, security, reconfiguration, remuneration, migration, load-balancing auditing............

Platform with well-defined QoS guarantees & support for time-critical applications: interactive multimedia etc.

Broadband Wide Area Network

Local ATM network
Technical Challenges

- Resource control
- Dependability
- Interactive Multi-media
- Ubiquitous connectivity
- Security
- Remuneration
Technical Challenges (continued)

- Trading & naming
- Agents & brokers
- Fast service development & deployment
- Multi-party, federated collaboration

Latest issue of CACM?
1993 issues of CACM?
I want an ATM switch like...
I want a pizza
Growth in I.S. deployment and use in WWW

- There is strong pressure to use this technology for commercial purposes.

- The technical challenges identified earlier need to be solved e.g:
  - URLs (<http://www.ansa.co.uk/>), URNs, URCs, URIs

- It is not just about retrieving documents

- Conclusion: an excellent opportunity to test and develop the components for the Global Management Engine.
Problems in the World Wide Web

• Content
  - large quantity, variable quality
  - discovery services
  - enquiry services

• Management of information resources
  - consistency — evolving resources
  - quality control

• Management of delivery services
  - server overload
  - unpredictable latency — timeout

• Growth rate makes these worse
Solution space

- Solutions native to WWW and Internet
  - extend HTTP, HTML
  - compatibility with existing clients and servers
- Solutions that enable WWW/CORBA interworking
  - compatible concepts
  - protocol translation
- Solutions that exploit WWW/CORBA interworking
  - CORBA-based components as part of solution
Active groups

• W3C — the World Wide Web Consortium
  - team based at MIT
  - common code library
  - reference implementations
  - active in standards work

• Internet Engineering Task Force working groups
  - HTTP — standard for protocol, enhanced versions
  - HTML — standard for mark-up language, enhanced version
  - URI — standard for URL, naming and meta-information
  - IIIR — FTP archives, Z39.50, integrated infrastructure vision
  - HTTPSEC — security in HTTP (proposed)
Solutions

• Replication to solve load problem
  - “mirror sites” in use today
  - another instance of same resource
  - but how is consistency maintained

• Versions of a resource
  - administrative technique — e.g. ANSA internal/external WWW servers
  - copy of a resource, but different resource

• Cost is in more administration — ad-hoc

• Interworking to access discovery services, management services
  - between Internet protocols today; resource at both http://..., wais://...
  - between Internet and CORBA; more of the same or different?

• Naming model is foundation
URC usage — URN to URL translation

**URN**: //dns:ansa.co.uk:1234

**URC**:

instances of the resource

- <URL:http://www.ansa.co.uk/ansa.html>
- <URL:http://www.hp.com/ansa.html>
- <URL:http://www.bellcore.com/ansa.html>

**Title**: Welcome to ANSA

**Abstract**: ANSA is an architecture for distributed computer systems ...

**Author**: APM Ltd.<apm@ansa.co.uk>

**URL**: http://www.ansa.co.uk/ansa.html

**Content-Type**: text/html

**URL**: http://www.hp.com/ansa.html

**Content-Type**: text/html

**URL**: http://www.bellcore.com/ansa.html

**Content-Type**: text/html
The fundamental questions

- What is a resource
  - what resources do we want to name — naming domain
- What is a name
  - what symbols (i.e. syntax) — naming convention
- What makes that a name for that resource, whose point of view
  - contexts, binding
- How do I tell you your name for a resource, given that I have my name
  - naming network, name resolution
- What can I/you do with the name once I/you have it
- Answers/definitions must be acceptable to Internet community
Trading and URCs — Discovery and Enquiry

- Similarities and differences between trading and URC service
- Meta-information
  - technical — locations, formats (type)
  - content (information) — keywords, subject matter
  - enterprise — access rights, cost
- Discovery vs Enquiry
  - Discovery — what resources about topic, online as PS/text/HTML, free
  - Enquiry — APM.1282.00.01: about what, where, format, price
- Naming model is fundamental
  - in response to discovery question
  - as part of enquiry question
Information Services Framework

Objective: “Develop the concepts and infrastructure needed to manage information services in the global network”

• Review of technical challenges
• The problems in the Web today
• The solutions being developed
• Applying CORBA technology to extend the functionality of the World Wide Web (WWW)
• Dynamically extending the functionality of WWW

Motivation: promoting market differentiation and innovation is crucial for commercial exploitation
A “commercial” 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
Extending the functionality of WWW using CGI Programs

User interface: a web browser (e.g. Mosaic) using html forms
Notes for previous slide (not for presentation)

- Currently server/browser interaction uses HTTP (HyperText Transfer Protocol) — an RPC protocol.
- The CGI (Common Gateway Interface) protocol running over (unix) pipes allows HTTP servers to communicate with other programs.
- This allows service providers to extend the functionality of their web servers (in ANSA we used it to provide a web interface to the trader).
- It is hard (error prone) to write the code to unmarshal the parameters to the CGI programs.
- A CGI Program gets “forked” each time it is invoked, so state has to be stored externally.
- CGI programs are driven by HTML (HyperText Mark-up Language) forms technology. HTML forms are active documents: browsers marshal the parameters provided by the user and (using HTTP) passes them to the HTTP server. In turn, the HTTP server passes them to the CGI program.
- Care is needed to make sure the HTML form and corresponding CGI program are consistent (both in numbers and “types” of parameters).
- Customisation is limited by what can be displayed in browsers using HTML forms technology — there is no notion of a custom client

Many of these problems can be solved by using IDL technology
CORBA IDL and the web

Browser

HTTP Server

IDL for Service

CGI program

Client stub
libwww

Bespoke client

Generate Form (on the fly)

Generate client stubs

Generate server stubs
Current status

• Based on Sun’s public domain CORBA IDL compiler front end
• Implementation complete, documentation in progress
• Some limitations
  - CORBA Type “Any” not supported
  - Stateless servers
  - No threads
  - “Interfaces” are URLs so no location transparency or relocation
• Size of sources (not including .hh, .h and .idl files):
  - Back-end 5800 lines of C++
  - Stub library 2300 lines of C
  - Example and test applications 1500 lines of C
Lessons learned

• We have been able to produce a programming environment for WWW which is close to CORBA by using stub compiler technology

• Easy to write a back-end once the basic concepts used by the compiler are mastered

• Decision to generate C code caused some minor difficulties where C and CORBA are not well aligned (C is more mature and more widely accepted in the Web community than C++)
  - C cannot return arrays from operations
  - There is no first class entity in C analogous to CORBA’s sequences

• To make debugging easy the “on the wire” format for everything is ascii: this hurts performance, but performance is not a priority at present.
Extending the HTTP protocol

A Fully extensible HTTP Server.
Instead of using CGI, new functionality appears as HTTP extension methods

HTTP server
methods = GET|HEAD|PUT|POST
| DELETE|LINK|UNLINK
| extension-method

HTTP

Client stub
libwww

HTTP

Enter arguments here:
Some Argument
Another argument:
And another:
e.g. credit card no.:
To invoke the operation: book
Notes for previous slide (not for presentation)

- HTTP is designed to be a fully extensible RPC protocol: servers are required to support a set of base methods and are free to implement extension methods.
- To the best of our knowledge, no current server supports the concept of extension methods — hence the need for CGI
- Requirements
  - Mapping of HTTP methods (GET|HEAD|PUT|POST|DELETE|LINK|UNLINK) to (CORBA) IDL
  - Some redesign and re-implementation of current (CERN) HTTP server
  - Modifications to stub compiler back-end
- Benefits
  - Easier to write new applications
  - Extending the functionality of a Web server and implementing standard HTTP methods (like PUT and POST) will be exactly like extending the functionality of a CORBA server (note could still use CGI if you want to)
  - Improved performance — do not have to fork a new process to run a CGI program
  - Can choose between stateless and stateful interaction between client and web server

This is static extensibility the server has to be (re)compiled to add new methods
Dynamic Extensibility using Script Technology

Script dynamically added to server
To provide new HTTP extension method

Client side script processing engine
Bespoke client program
Migrates as a script to client

HTTP server

Mosaic

Lots of text goes here to tell the customer what to do.
What follows are input fields which allow customer to:
supply parameters to the operation which is about to be
invoked

Enter arguments here:
Some Argument
Another argument: default
And another: default
e.g. credit card no.: default

To invoke the operation: book

HTTP

Script processing engine
(Safe-Tcl)
Notes for previous slide (not for presentation)

- We assume everybody has a browser; in general not everybody is going to have the bespoke client application — this means that it needs to migrate to the customer.
- All interaction will bootstrap off browsers which are ubiquitous; each client will also have a script processing engine.
- There are big security issues: who is allowed to extend the server; who does a client trust to send it a script?
- Current status: dynamically extensible server has been implemented, now experimenting with client side migration.
- Future work: how does this relate to the ANSA (and CORBA) computational models?
- Longer term objective: the bespoke client should be able to deliver real time media. Returning to our earlier scenario, this would allow the customer to view trailers of the film or video clips of the performance as part of the process of deciding whether or not to book seats.
Summary

• Reviewed the basic challenges and problems in the World Wide Web.
• Extensibility is vital in a commercial WWW environment to promote market differentiation and innovation.
• CORBA (IDL) technology can be used to provide WWW extensibility within a CORBA like programming environment.
• Scripting engines can be used to provide dynamic extensibility.