Introduction: Web Services Are Changing

- Service complexity is increasing
  - from static doc-based resources to complex applications
  - commercial use: from low-cost advertising to critical applications
  - commercial applications demand high QoS
- Common set of application requirements is emerging
  - persistence, concurrency control, fault-tolerance, etc.
  - support for dynamic content
    - responses to client-driven applications
    - system-driven customisations
  - increased use of Web application toolkits
- Manageability is increasing in importance
Management Issues

- Site management
  - supporting increased numbers of users
  - painless addition and removal of services
  - supporting increasingly computationally intensive services
  - providing high QoS
  - scalability

- Service management
  - supporting look and feel changes consistently
  - supporting change to presentation logic driving dynamic content
  - supporting change while maintaining consistency

W3Objects Overview

- Framework to aid in the construction of Web-based applications
  - key goal is to provide an extensible Web architecture
- W3Objects are encapsulated entities
  - interface inheritance provides polymorphism
  - code reuse achieved using behavioural inheritance
- W3Objects are organised and named within contexts
- Referencing mechanisms ensure referential integrity and migration transparency (See WWW5 paper)
- Inter-object communication via remote procedure call (RPC)
W3Objects Site Architecture

Web browser → HTTP → 'extensible' Web server → RPC → W3Objects gateway module → W3Objects

Web Access to W3Objects

HTTP Request
GET w3o/banking/balance?acc=1234
HTTP/1.0

HTTP Response
HTTP 200 OK
<headers>
<body>

Web server → RPC → Nameserver → balance

Nameserver → RPC → W3OServer

W3OServers
W3Objects (services)
Site Management Features

- Scalability through transparent distribution
  - arbitrary allocation of services to machines, transparent to users
- Transparent service migration
  - services may be migrated between processes and hosts
  - referential integrity ensured
- Introduction and removal of services
  - new services added by registering them in the nameserver
- Support for stateful services
  - W3Objects persist across requests
  - session-state can be held in memory or optionally on disc
  - persistence support provided
- Management operations accessible via API or Web interfaces

Comparison with Alternative Techniques

- Common Gateway Interface
  - highly inefficient
  - centralised services
  - poor support for session-based services
- Server APIs
  - performance benefits over CGI
  - poor isolation of faults
  - centralised services
  - poor support for session-based services
Service Management Introduction

• Components of a service
  – clients see a service as a collection of pages
  – some may be held as static components
  – some may be generated dynamically
    • results of a user-initiated computation
    • customised presentation
  – services consists of functional and presentation components
  – services may contain replicated presentation components
  – services may share presentation components

• Management operations
  – changes to static components, e.g., look and feel changes
  – changes to presentation logic
  – addition of new operation interfaces

Manageable W3Object Services

• Strong separation of presentation logic from functional aspects
• A service is logically represented as a single object
• Internally a service object contains view objects
  – a view either represent a complete page or a page component
  – views are either static or dynamic; private or shared
• Presented pages are created by assembling view objects
• Manageability is obtained through inheritance
  – develop application without consideration for Web presentation
  – develop Web interface using view components
  – dynamic views glue the Web interface to the functional interface
Service Management Features

- Web interface can be configured at run-time without outages
- Isolation of commonality
  - shared views are updated once; changes automatically propagated
- Encapsulation
  - entire service can be managed as a single object
- Service evolution
  - views can be created, modified and removed
  - views can be migrated, e.g., created privately then shared
- Accessible management interface
  - all management operations can be accessed via Web interfaces
Manageable Object Internals

Simple Static Views
Scripted Views

- Implemented using W3OScript
  - server-side scripting language based on tcl
  - safe tcl interpreter augmented with W3Object-specific commands
  - implemented using Embedded Tk (ET)
- Can be used to define presentation logic
  - tailored presentations
- Provides glue between Web interface and functional interface
  - user classes can define new W3OScript operations
- W3OScript resources can be configured using Web interfaces

Example: Supporting Metadata

- Resource metadata
  - necessary for improved searching and indexing
  - introduction can be time-consuming and error-prone
  - lots of replicated information
  - updates are a headache
  - standards are not yet well defined
- W3Object support
  - specialised view object
  - provides convenient interface entry interface
  - common components can be shared
  - data held in structured fashion
  - W3OScript defines presentation
Example: Supporting Metadata

Alternative Approaches

• Style sheets can provide consistent look and feel
  – use is encouraged
  – does not help maintaining consistency of replicated data

• Server-side includes (SSI)
  – server-parsed HTML templates may include CGI calls
  – improves flexibility
  – poor performance

• W3Objects
  – pre-parsing where possible to improve performance
  – overheads of RPC can be alleviated with caching
Summary

- Manageability is increasing in importance
- Distributed object technology provides a scaleable platform for Web service provision
- Smart referencing mechanisms support migration transparency
- Separation of presentation and functional logic improves service manageability
- Isolating commonality simplifies management
- Scripted resources simplify presentation logic management
- Web-based interfaces improve management accessibility

W3Objects

http://arjuna.ncl.ac.uk/w3objects/

David Ingham
Research Associate, Arjuna Project
Department of Computing Science, Newcastle University, U.K.

Email: dave.ingham@ncl.ac.uk
URL: http://www.cs.ncl.ac.uk/~dave.ingham/

This work has been partially funded by: