6th International World-Wide Web Conference
Santa Clara, CA, USA, April 5-11, 1997

Object-oriented Web Servers and Data Modelling Workshop

W3Objects

A Distributed Object-Oriented Web Server

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/

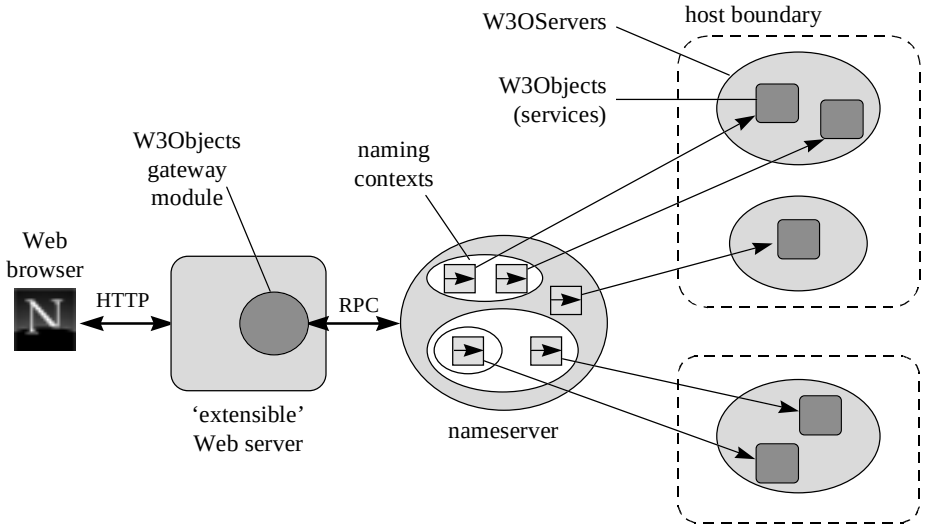
1

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)

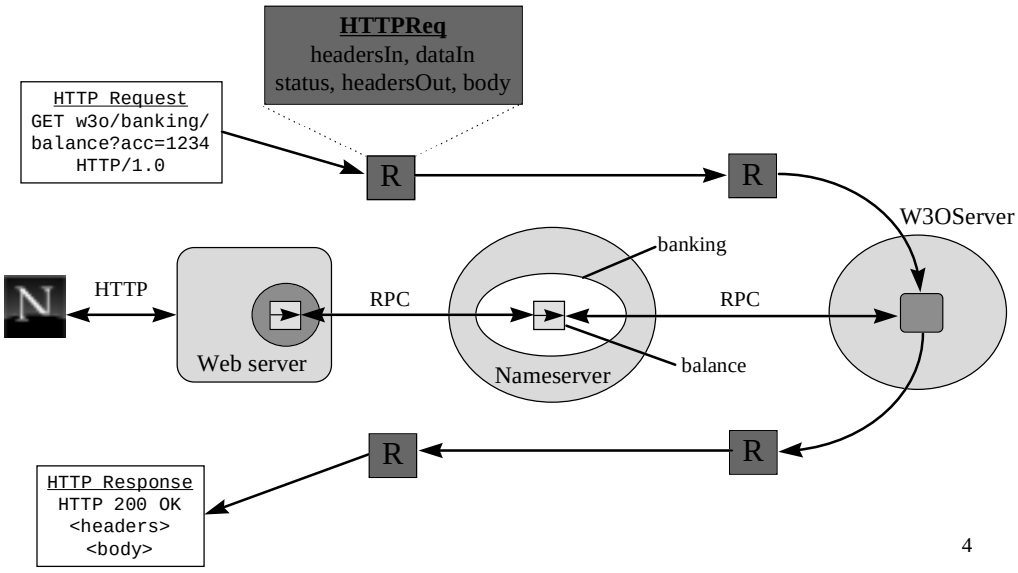
2

W3Objects Site Architecture



3

Web Access to W3Objects



4

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

5

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

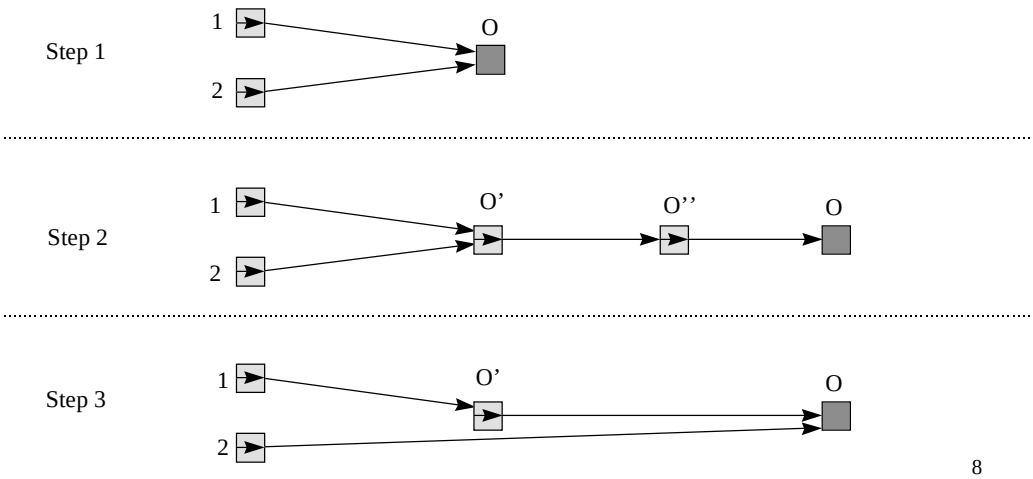
6

W3Objects Referencing Model

- Aims
  - to provide referential integrity
  - to provide migration transparency
  - to provide flexible mechanisms to support differing object requirements
- Integration of referencing techniques
  - all W3Objects support forward referencing with chain short-cutting
  - cheapest guarantee of referential integrity
  - per reference customisation for fault-tolerance or performance
- Implementation built using Shadows
- Deployed within W3Objects servers
  - guarantees integrity of links between W3Objects

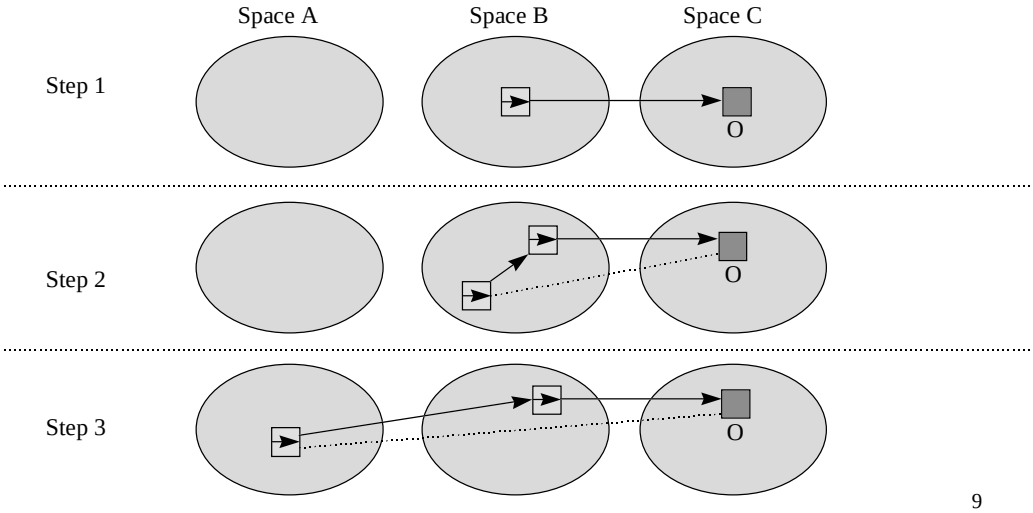
7

Object Migration & Short-cutting



8

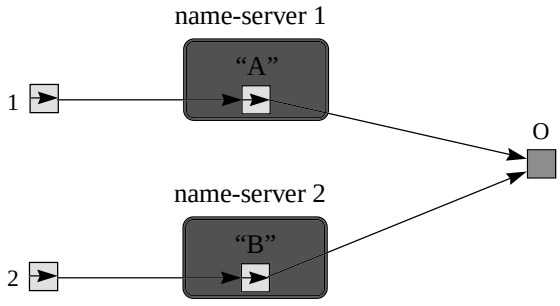
Obtaining References



9

Name-servers

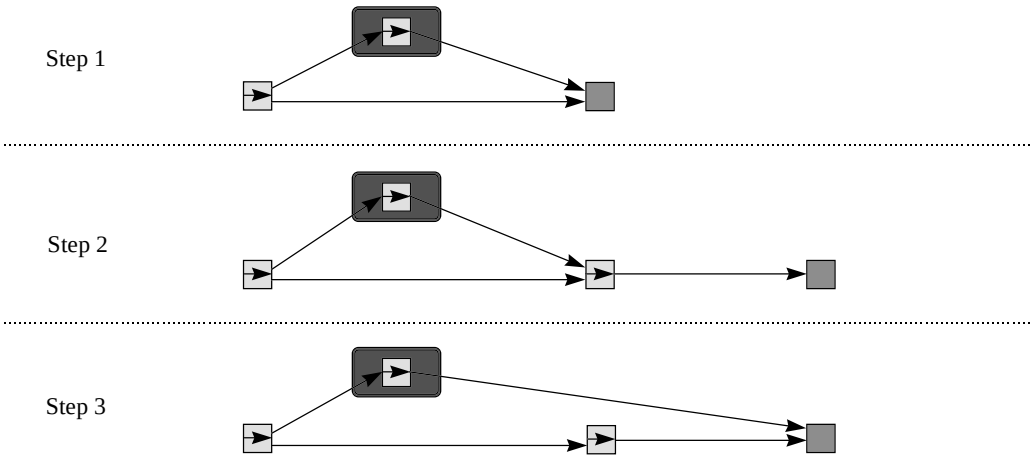
maintaining bindings between names and objects



10

Name-servers & Call-back

providing alternative paths to objects



11

Referencing Model Summary

- Forward referencing
  - default mechanisms for all W3Objects
  - achieves referential integrity
  - provides migration transparency
  - chain-shortcutting reduces points of failure
- Name-servers
  - used to provide name to object bindings
  - used to provide alternative paths to an object
- Callback
  - used in conjunction with name-servers to eliminate common paths
  - used with references to minimise invocation overhead

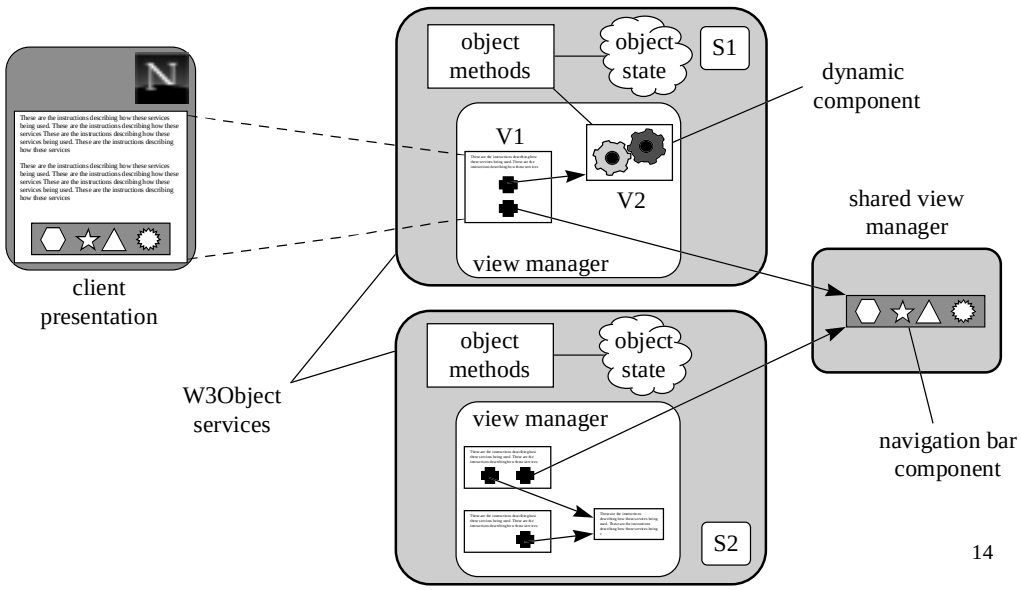
12

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

13

Views



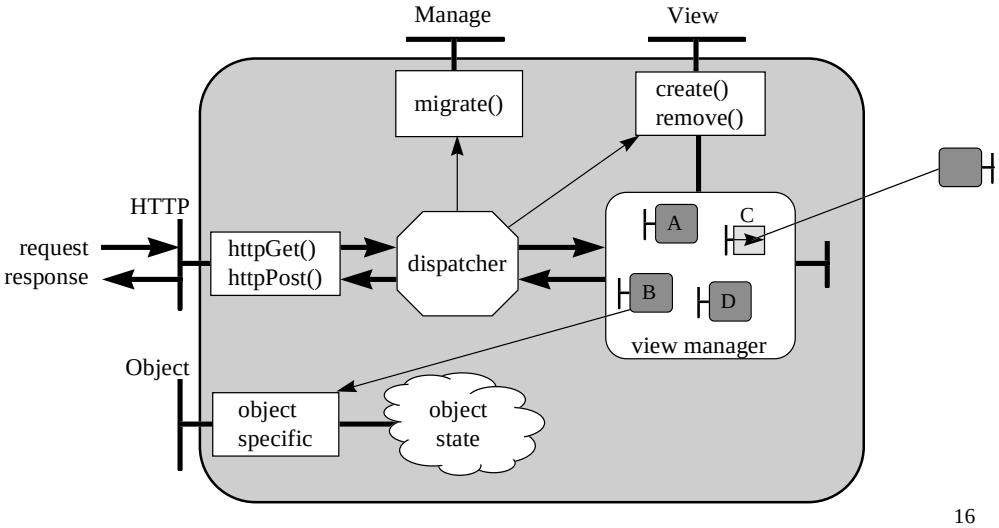
14

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

15

Manageable Object Internals



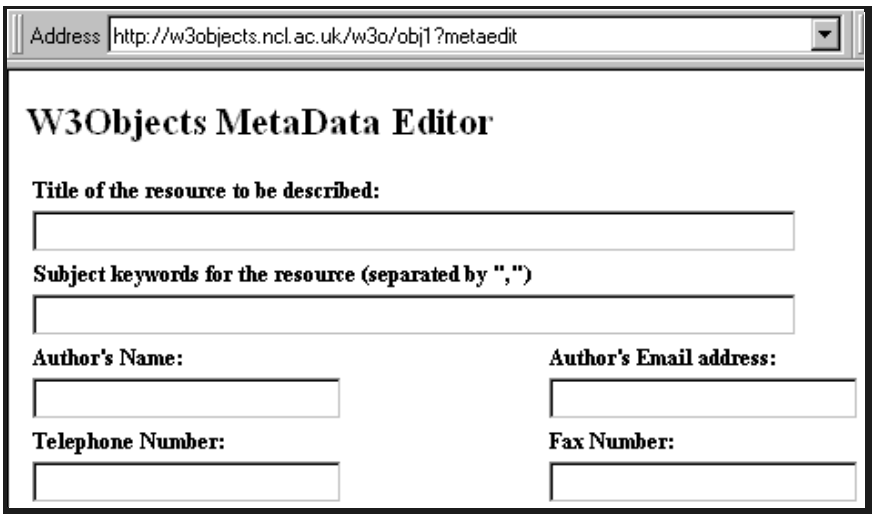
16

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

17

Example: Supporting Metadata



18

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 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

19

Data Modelling: *Object Wrapping*

- Web access to legacy systems requires gateways
  - protocol translations
  - interface translation
- Object-oriented model is suitable for legacy system gateways
  - strong separation of interface & implementation
  - object wrapping allows legacy resources to be treated as native
- Wrapping using W3Objects provides Web gateway
- Distributed objects offer increased flexibility
- Scripted presentation logic simplifies management

20

Summary

- W3Objects provides a Web-to-object gateway
- Distributed object technology provides a scaleable platform for Web service provision
- Smart referencing mechanisms support referential integrity and migration transparency
- Support for building Web services
  - persistence, concurrency control, etc.
  - support for session-based applications
- Manageable services
  - strong separation of presentation and functional logic
  - scripted resources

21

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:

HEWLETT PACKARD LABORATORIES GPT EPSRC

22