Quartz

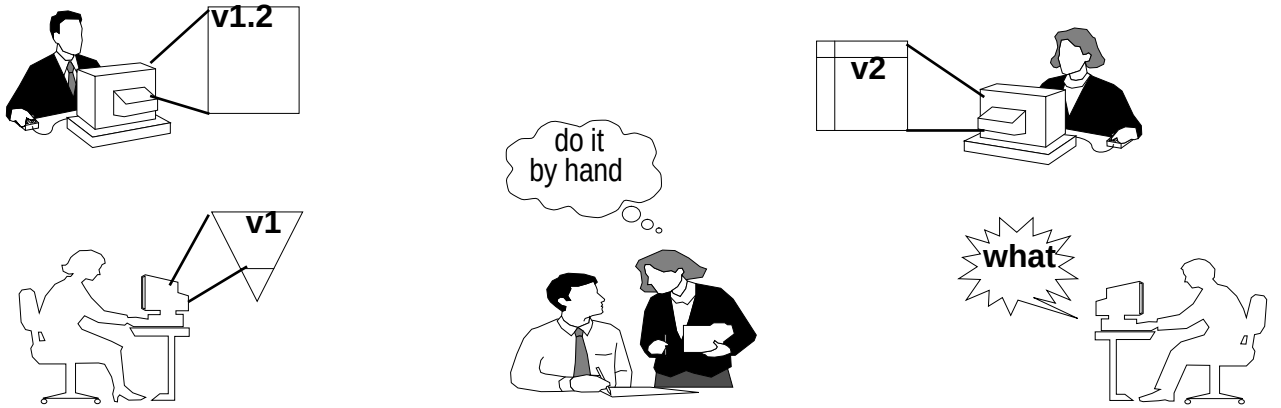
Making CORBA Objects Easy to Reach

Zhixue Wu and Toby Speight

May 1996

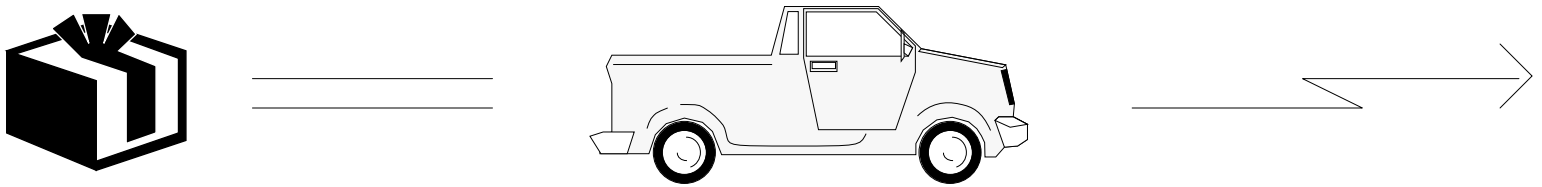
IT Application Situation

- Users may be located far away from the main site
- They are equipped with variety of machines
- They may have only temporary or one-off use of a service
- They require complete consistency
- They require punctual connection



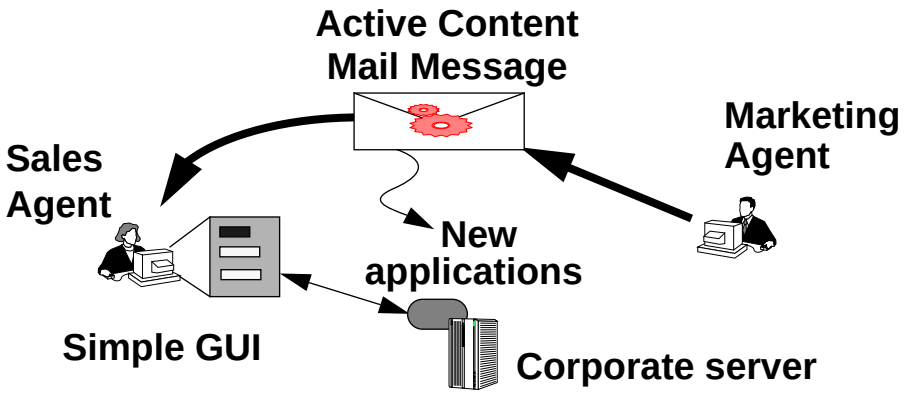
Requirement

- **Problem**
  - Software installation is costly and difficult
  - Each user-interface has a learning curve
  - Software has very low portability
- **Requirement**
  - A method to wrap a service with a familiar interface
  - A simple method to distribute a service



Core Idea of Quartz

- Make CORBA services
  - available widely and easily
  - accessible from desktops and consumer equipment
  - simple to use
- Combining email, WWW and object technology will
  - allow CORBA objects to be accessed from Web
  - use email and WWW for service distribution
  - use Web browser as user interface

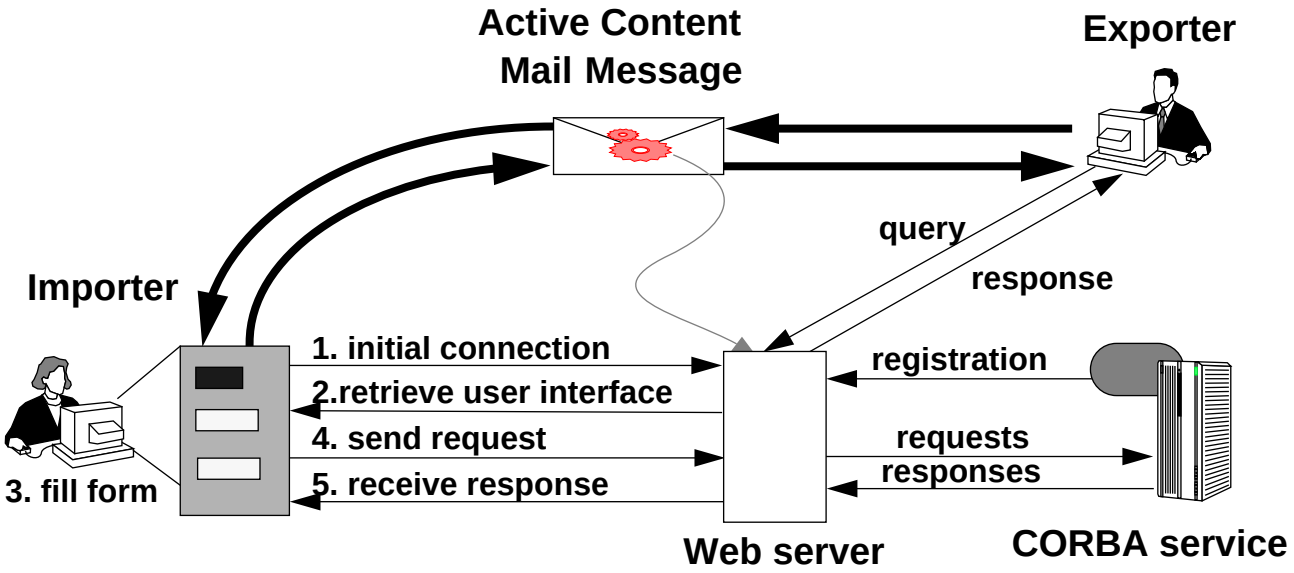


Advantages and Benefits

- **Accessible from different sites and different platforms**
  - **workstations, PCs, desktops**
- **Easy connection and installation**
  - **email, Web infrastructure, Java interpreter**
- **Simple but powerful interface**
  - **Web browser**
- **Easy access to related information**
  - **on-line help, related events**
- **Rapid implementation**
  - **reusable objects, Java, HTML**

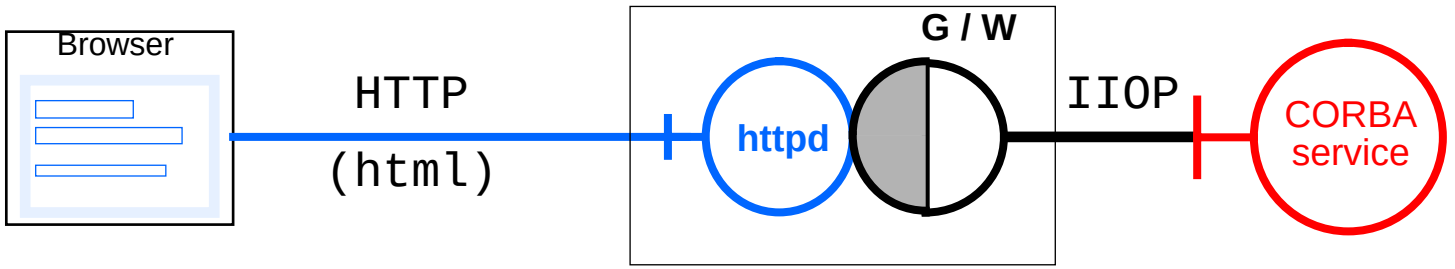
Scenario

- Service development
- Client interface development
- Registration
- Use

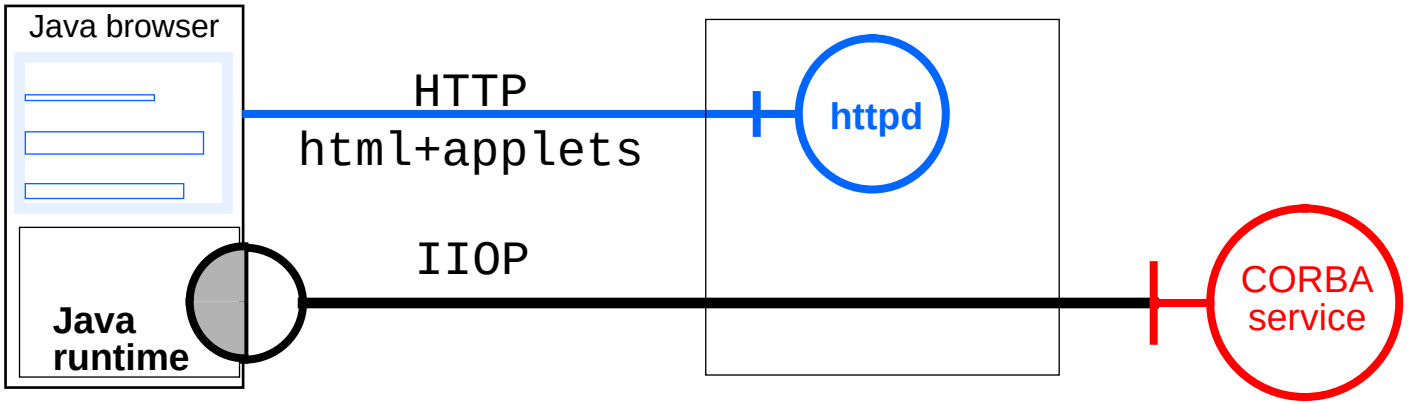


Interoperability between WWW and CORBA

- **ANSAWeb**

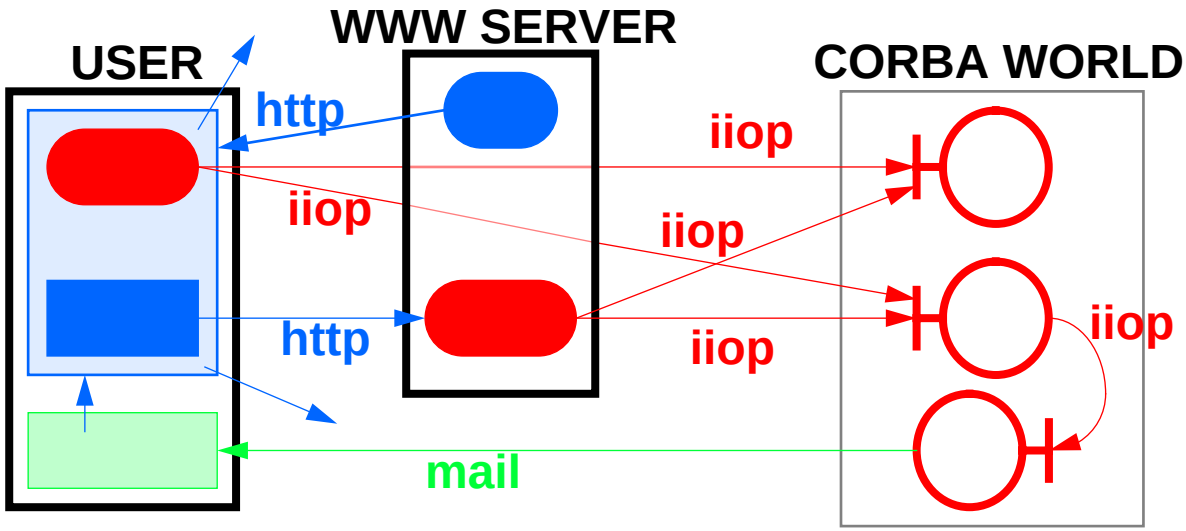


- **Jade**



Web Interface for CORBA Objects

- Use CORBA objects as backbone
- Implement CORBA interfaces as Java applet interfaces
- Use Java as Web programming language
- Use email as a means of notification
- Use Web as execution environment

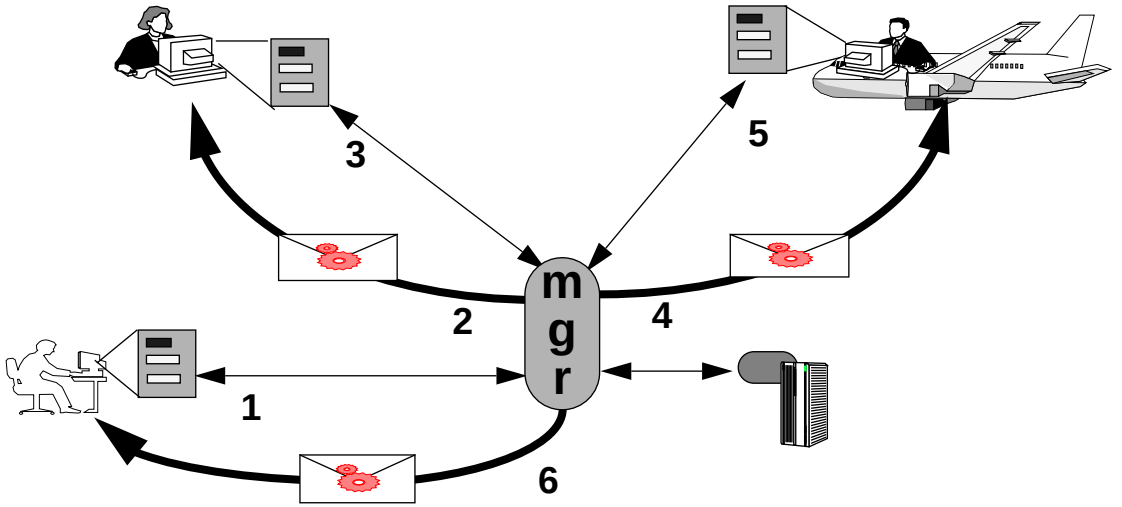


Aim of the Project

- **Demonstration via working examples**
  - **show how to make use of the idea in practical applications**
  - **demonstrate its feasibility, advantages and benefits**
- **Provide reusable components**
  - **implement domain-specific and generic components**
  - **make generic components reusable**
  - **implement new applications by customising**
- **Provide support tools**
  - **make it easier to write Web interfaces**

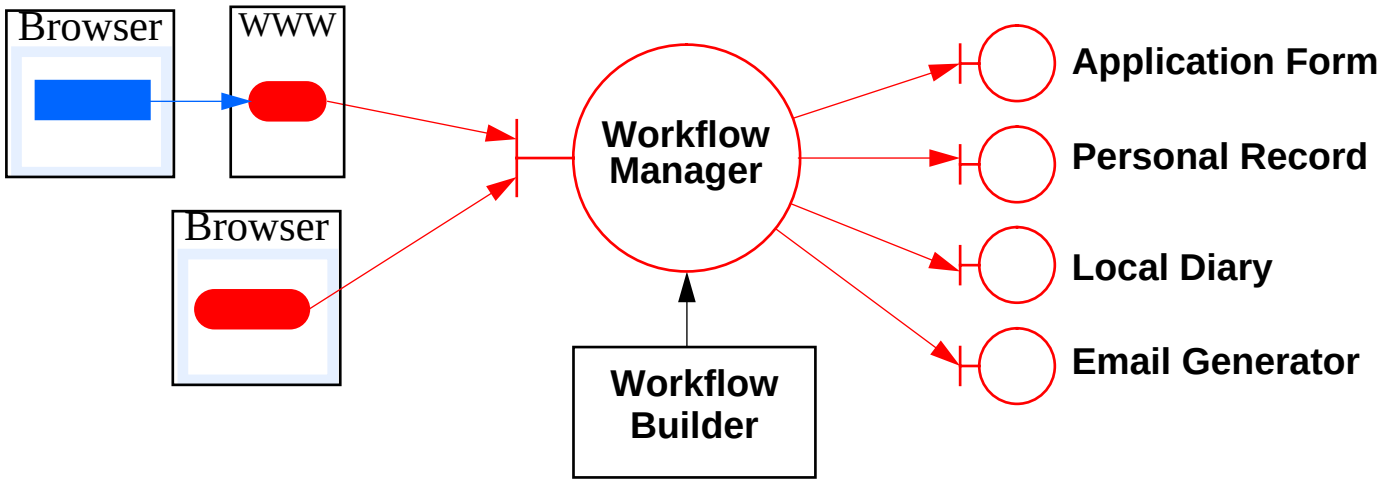
Example Application: Holiday Request

- **Employees**
  - submit request (workstations)
- **Secretary**
  - check available holiday time (PC)
- **Manager**
  - make decision (notebook)



The Architecture

- **Workflow manager**
  - **object invocation based on an event-action table**
- **Workflow builder**
  - **define an event-action table for an application**
- **Application objects**
- **Web clients**



Advantages over Workflow Software

- Distribution transparency
- Platform transparency
- Simple user interface
- Powerful on-line help
- Easy access to related information
- Low cost connection and installation

Project Deliverable

- **Demonstrations**
  - leave request system
  - project proposal and review management
- **Object Library**
  - re-usable components
- **Builder Package**
  - tools to make application implementation easier
- **Architectural Report**
  - architectural features and design decisions
- **Project Report**
  - project principles and design decisions