
Intermediaries

New Places for Producing and Manipulating Web Content

Rob Barrett & Paul Maglio
IBM Almaden Research Center

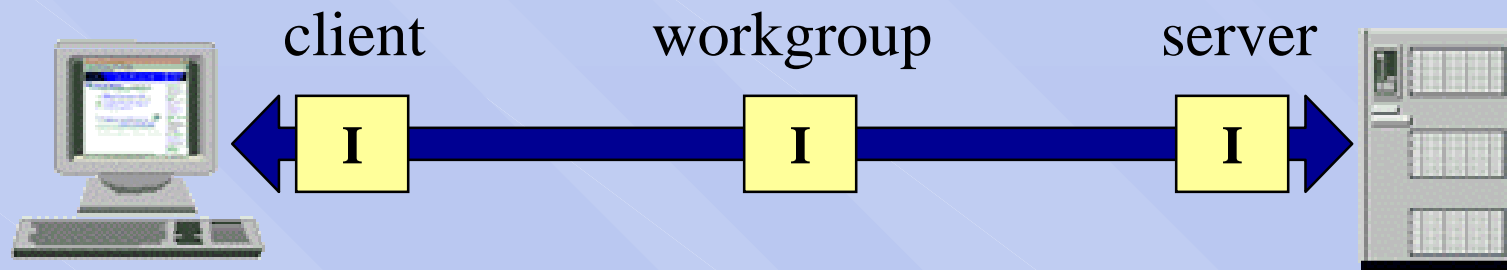
April 1998

Server-Centric Web



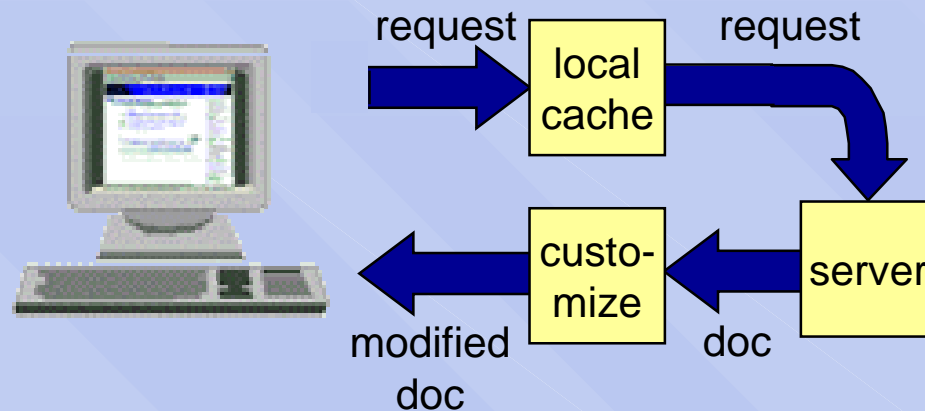
- ◆ server controls all content
- ◆ client and pipe are content-passive
- ◆ applets and plugins notwithstanding

Intermediaries



- ◆ computational elements along data path
- ◆ take advantage of well-defined pipe
- ◆ observe, respond to requests, modify
- ◆ anywhere along path

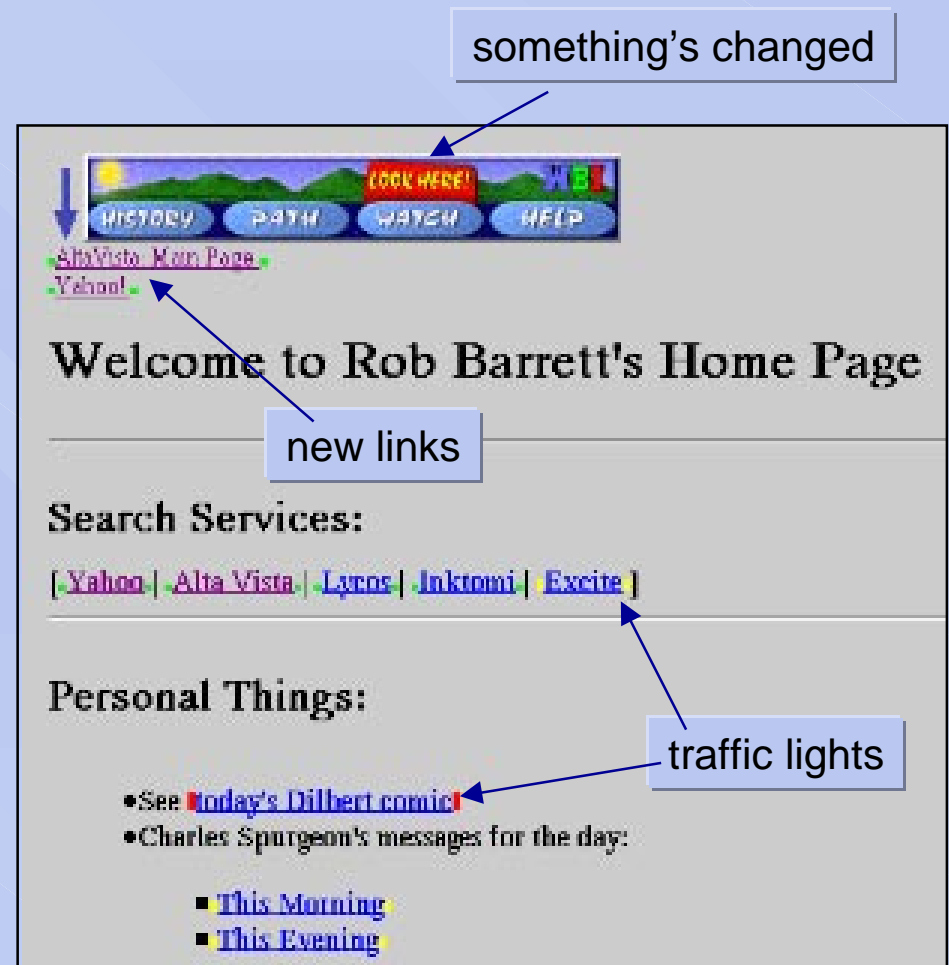
An Example Data Flow



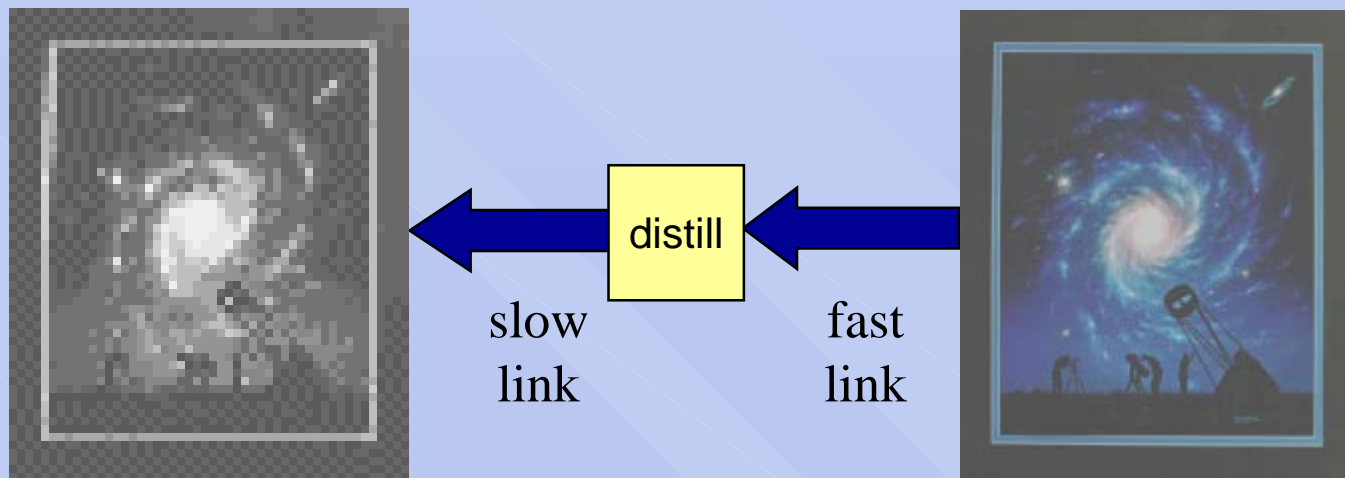
- ◆ browser issues request
- ◆ passes through local cache
- ◆ cache miss, so passed to server
- ◆ resulting document customized and returned

Example: Personalization

- ♦ original motivation (see CHI'97 paper)
- ♦ web changes based on user and history
 - history
 - traffic lights
 - shortcuts
- ♦ download at:
www.cssrv.almaden.ibm.com/wbi/

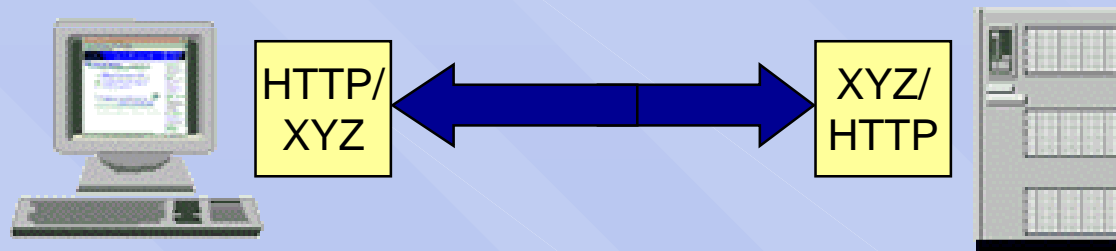


Example: Content Distillation



- ◆ content has various representations (e.g. image quality)
- ◆ transform according to speed, display, etc.
- ◆ see Fox, et al (WWW5 proceedings)

Example: Protocol Extension



- ◆ add protocol features without changing client or server
 - data compression
 - new authentication methods
 - backward compatibility

New Computational Power

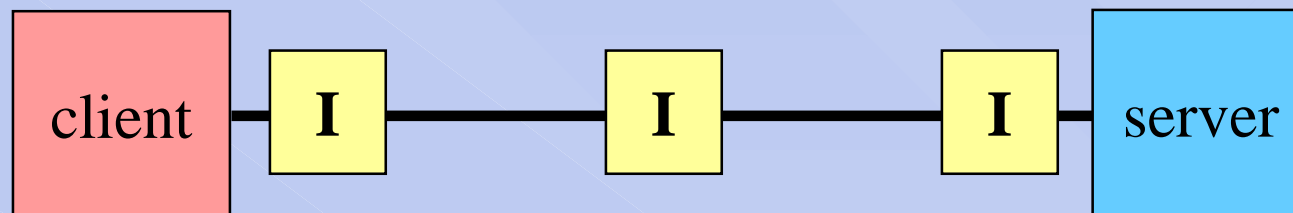
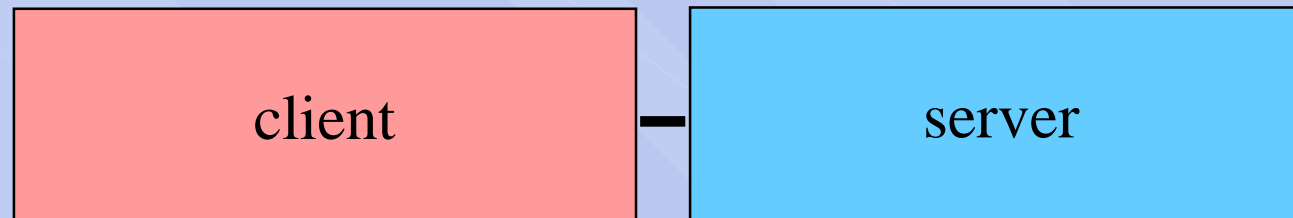
- ◆ key idea
 - expanded content computation
 - server no longer has control of content
- ◆ web gains
 - customizable and extendible at all points
- ◆ content production and manipulation
 - server, client, or workgroup/ISP/firewall

Outline

- ◆ Moving to Intermediaries
- ◆ Architecture
- ◆ Implementation
- ◆ Related Work and Future Work

Progression: Application to Data

monolithic application



Extract Intermediary from Browser

- user-interface
- URL requests
- document display
- applet exec



browser

client-side
intermediary

- pluggable computation
- process requests
- retrieve documents
- customize/personalize

- ◆ move function from browser to intermediary
 - cache, off-line cache
 - network communications
 - bookmarks, history

Add Programmability to Proxy

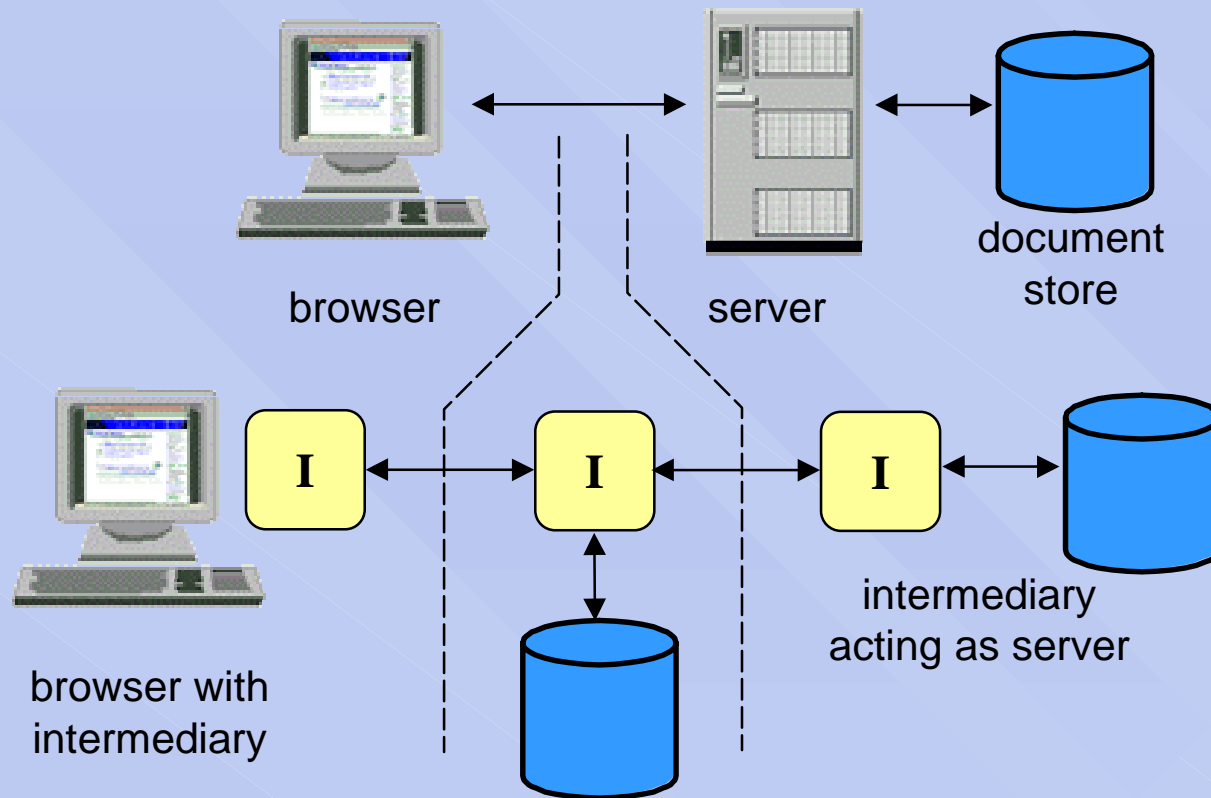
- ◆ normally transparently sends requests/docs
- ◆ sometimes has caching functions
- ◆ make into full-fledged intermediary
 - transparency is a special case
 - add access to workgroup resources
 - screen docs for viruses
 - content distillation
 - wide range of pluggable features

Replace Server w/Intermediary



- ◆ server always returns doc for a request
- ◆ intermediary may
 - respond or forward
 - observe transaction
 - modify request or doc

Drawing the Lines



◆ generalize web model with intermediaries

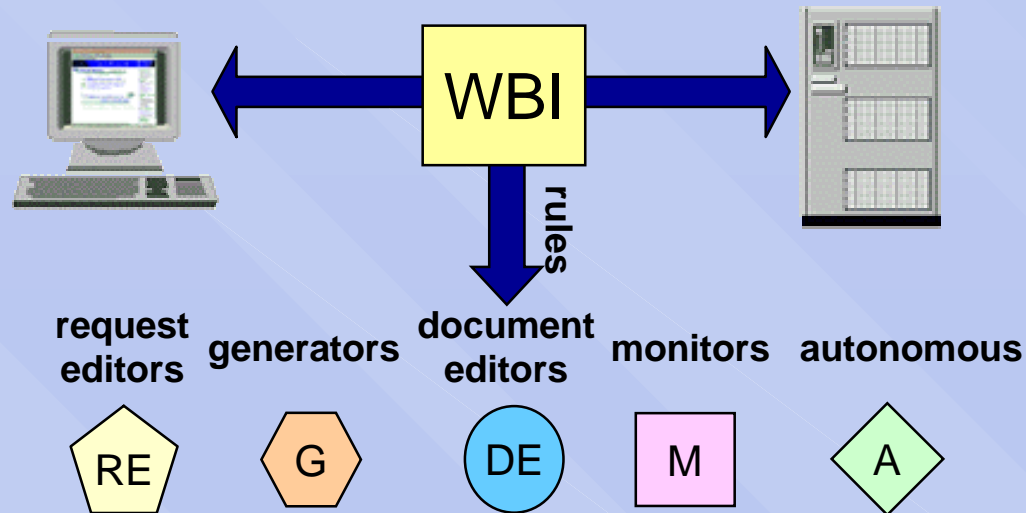
Outline

- ◆ Moving to Intermediaries
- ◆ **Architecture**
- ◆ Implementation
- ◆ Related Work and Future Work

Why One Architecture?

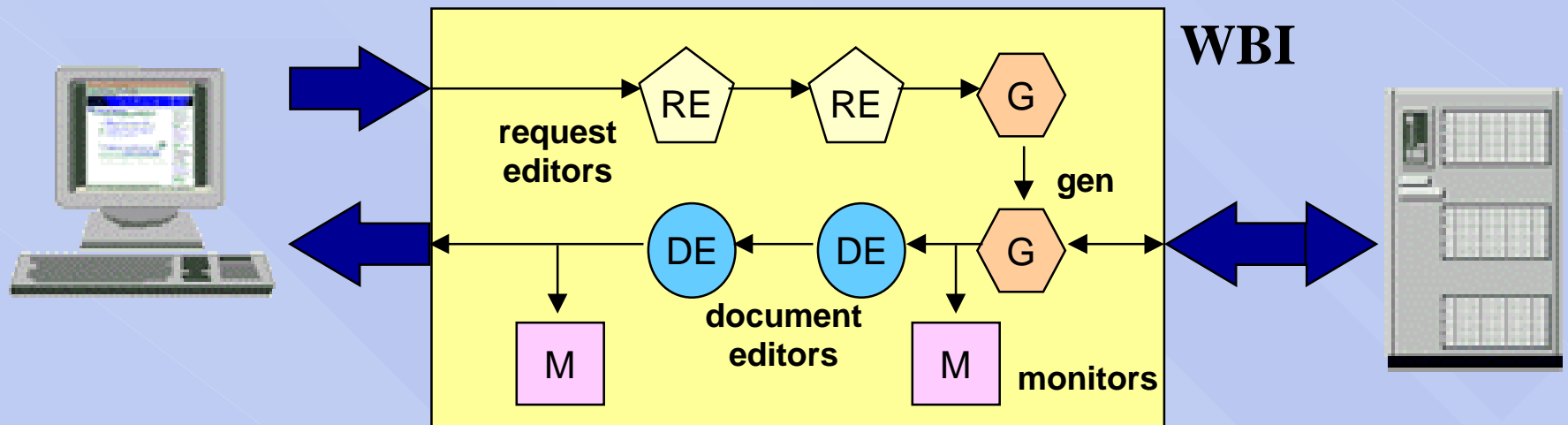
- ◆ building a robust intermediary is tough
 - browser/server protocol perversions
 - rare error conditions
 - interactive multi-threading
- ◆ separate framework from particular int. apps
 - app. developer needs only deal with data
 - uniform interfaces across intermediary platforms

WBI Architecture



- ◆ five basic building blocks (MEGs)
- ◆ WBI app. (“plugin”) built out of MEGs
- ◆ rules control MEG involvement

WBI Data Flow



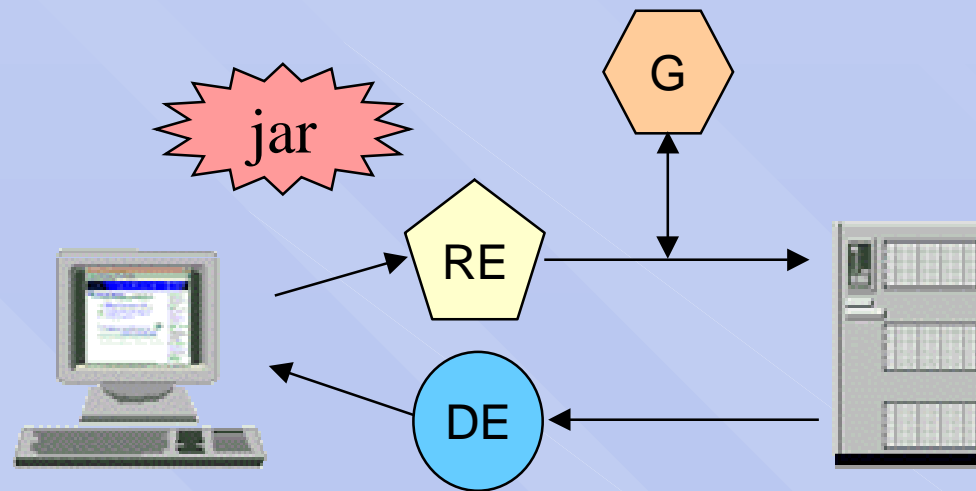
- ◆ dynamic flow by rules and transaction type
- ◆ chaining and rejecting
- ◆ monitor placement

Rules

- ◆ boolean expressions with wildcards
 - rejection allows other selection criteria
- ◆ switch on structured data in req/doc
 - URL, content-type, client addr, protocol, etc.
- ◆ priority
 - chaining (editors)
 - ordering (generators)

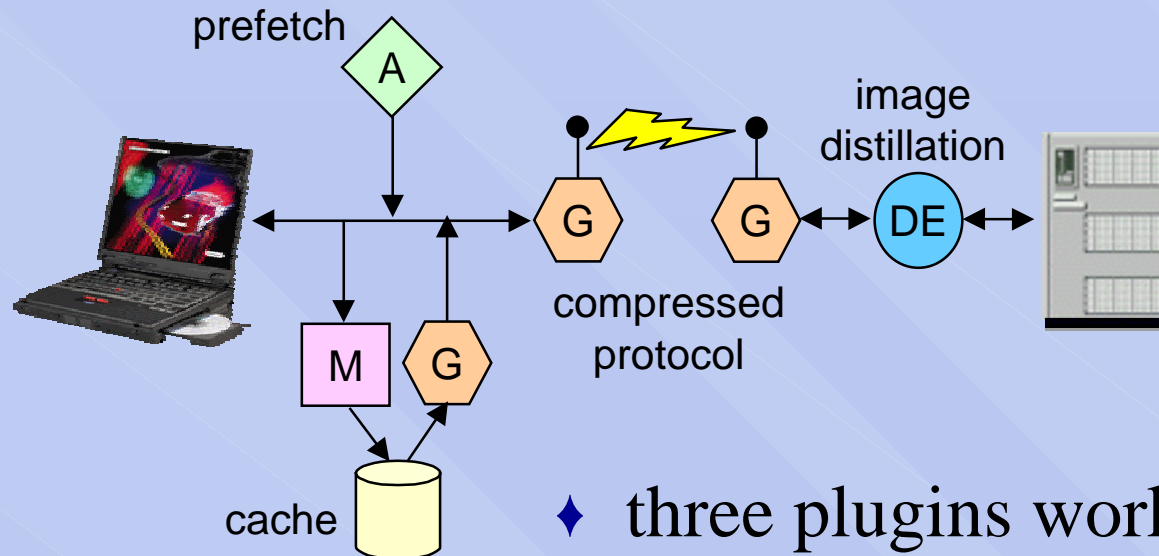
```
host=* .stanford.edu & content-type=text/*
```

Example: Cookie Manager



- ◆ Req Editor: adds cookies from cookie jar
- ◆ Doc Editor: removes cookies and puts in jar
- ◆ Generator: produces HTML for management
- ◆ cookie policies, mult. users, mult. browsers

Example: Wireless Web



- ◆ three plugins work together
 - cache with prefetch
 - compressed protocol extension
 - image distillation

Outline

- ◆ Moving to Intermediaries
- ◆ Architecture
- ◆ **Implementation**
- ◆ Related Work and Future Work

Languages/Protocols

- ◆ Native Win95/NT or Java
 - mix and match base code and plugins
- ◆ HTTP, HTTP-S, FTP, Gopher
- ◆ proxy- or server-type requests
- ◆ usable with any browser or server

Configurations

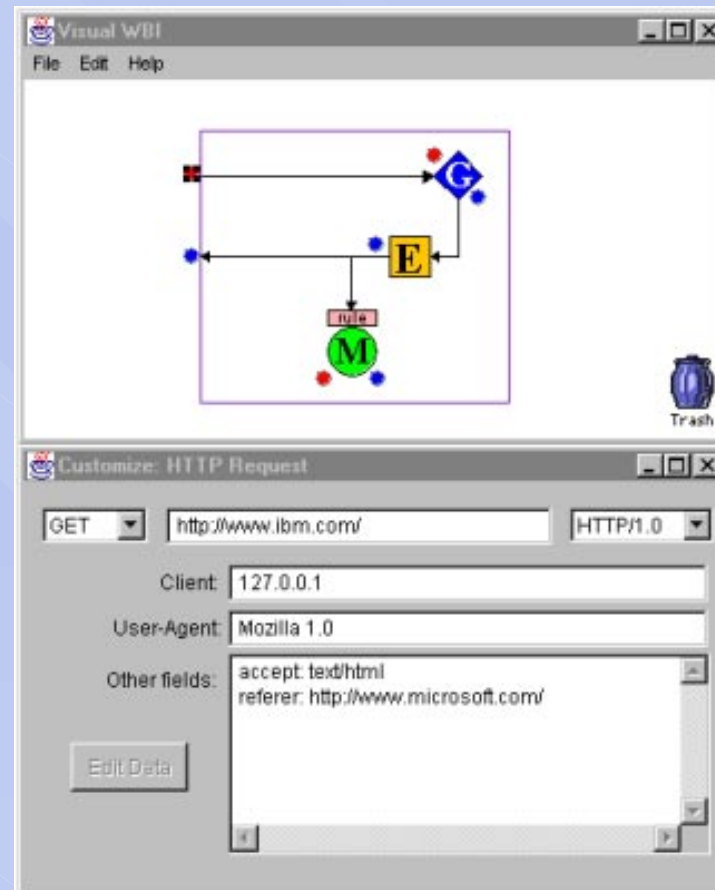
- ◆ client-side
 - install on client machine
 - forwards to Internet or firewall
 - ◆ workgroup-side
 - install on any networked machine
 - users identified by HTTP Proxy Authentication
 - ◆ server-side
 - default forwarding to designated server
 - ◆ same code in each case
-

Component Model

- ◆ Java Beans
 - WBI Plugins, MEGs are beans
- ◆ Properties
 - Plugins: MEGs and plugin-scope data
 - MEGs: rules and configuration info
- ◆ Library
 - static HTML, serve file from disk, add annotations, interpret forms, “doc moved”

Visual Programming/Debugging

- ◆ create MEGs
- ◆ build-up plugin
- ◆ follow req/doc through chain
- ◆ work-in-progress



Performance

- ◆ firewall & server are special cases, so intermediaries are not *fundamentally* slower
- ◆ C++ WBI server is 1.2x slower than MS IIS
- ◆ Java JIT WBI is additional 2.5x slower
- ◆ other functions use more processing
- ◆ care must be exercised!

Outline

- ◆ Moving to Intermediaries
- ◆ Architecture
- ◆ Implementation
- ◆ **Related Work and Future Work**

A Million Ways to Do the Web

- ◆ many programming models for the web
 - CGI, NSAPI, ASP, Oreo, Servlets
 - ◆ compare with Servlets
 - more languages: C, C++, Java
 - more locations: client, server, workgroup
 - more basic operations: MEG vs. servlet
 - more granularity: Plugin & MEG
 - dynamic configuration
 - servlets allow remote code loading
-

Metadata

- ◆ current web is human-readable only
- ◆ difficult to parse, impossible to understand
- ◆ PICS defines web metadata language
 - content description
 - item, price, merchant, terms
- ◆ presence greatly extends intermediary applications -- agents

Additional Streams

- ◆ beyond HTTP
 - mail, news
 - routing, summarization, virtual newsgroups
 - push
 - telephone, pager
 - transfer between modalities
- ◆ sharing data between streams
 - intermediaries communicate via blackboard

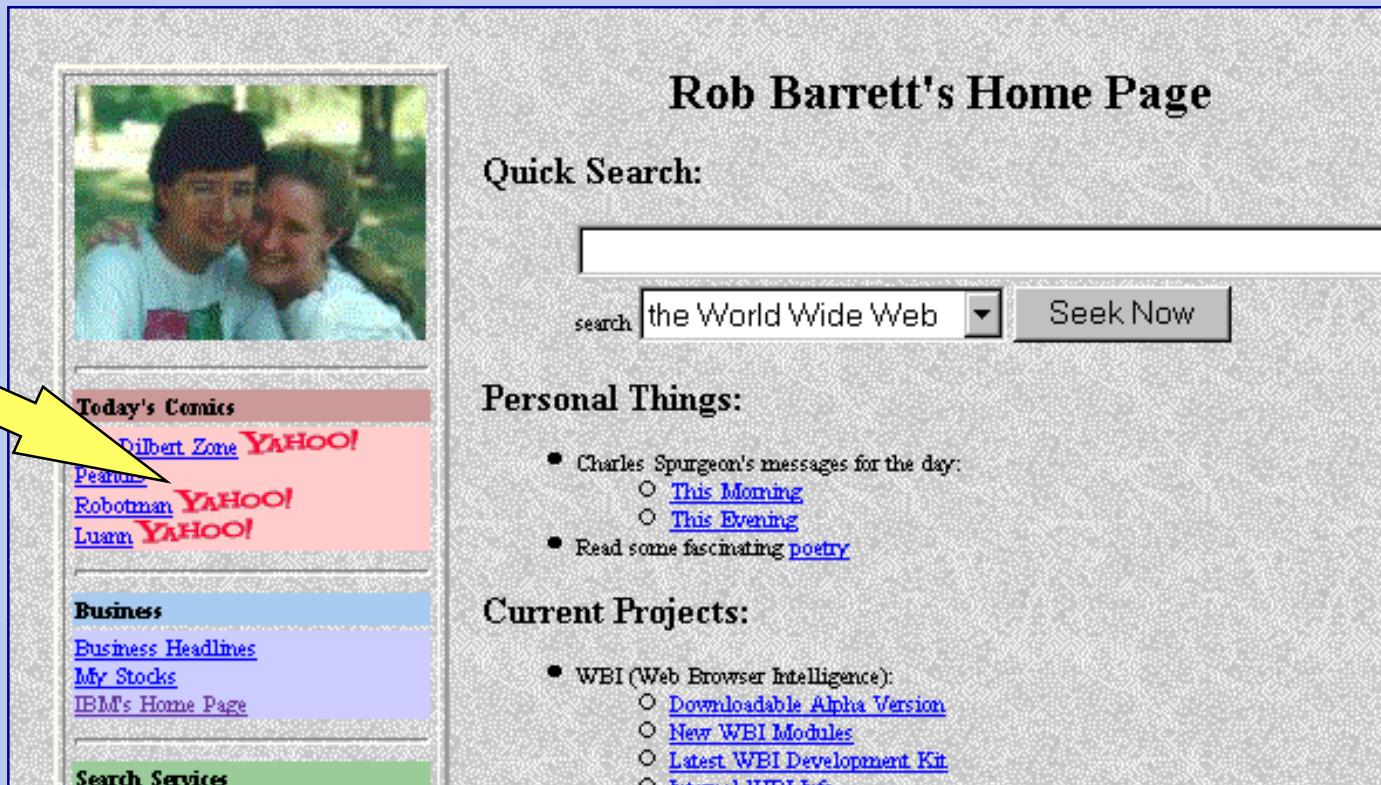
General Programming Model?

- ◆ MEG model can be applied to functional programming
 - RE: modify parameters
 - G: produce result
 - DE: modify result
 - M: observe calculation
 - ◆ would you want to do this??
 - ◆ cf. Stanford DigLib search engine proxies
-

More Examples....

- decorate links to Yahoo!-listed pages

*automatically
add link to Yahoo!*



Rob Barrett's Home Page

Quick Search:

search

Personal Things:

- Charles Spurgeon's messages for the day:
 - [This Morning](#)
 - [This Evening](#)
- Read some fascinating [poetry](#)

Current Projects:

- WBI (Web Browser Intelligence):
 - [Downloadable Alpha Version](#)
 - [New WBI Modules](#)
 - [Latest WBI Development Kit](#)
 - [Extended HTML 5.0](#)

Today's Comics

- [Gilbert Zone](#) **YAHOO!**
- [Pearls](#)
- [Robotman](#) **YAHOO!**
- [Luan](#) **YAHOO!**

Business

- [Business Headlines](#)
- [My Stocks](#)
- [IBM's Home Page](#)

Search Services

Yahoo Advisor

- user: more valuable web
- Yahoo!: more access to valuable info



The screenshot shows the Yahoo! homepage with the following elements:

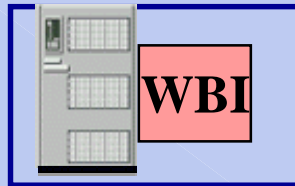
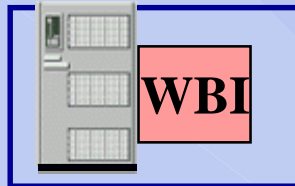
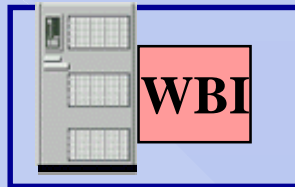
- YAHOO!** logo in red.
- Three circular icons: "My Personalize", "Add URL", and "More Yahoos".
- A red banner for "TAKE BACK THE NET.™" by "The Mining Company™" with the URL "www.miningco.com" and a "Click now!" button.
- Breadcrumbs: [Top](#): [Entertainment](#): [Comics and Animation](#): [Comic Strips](#): [Titles](#)
- A search input field with a "Search" button and a link to "Options".
- Radio buttons for "Search all of Yahoo" (selected) and "Search only in **Titles**".
- A list of comic strips:
 - [Angriest Dog in the World](#) **NEW!** - comic strip by director David Lynch.
 - [Abductee](#) - deals with subjects ranging from aliens to horror to Star Trek.
 - [Adventure with A.R. Bogworthy & Platis, A](#) - Earth is evacuated; everyone now lives in the galactic suburbs. Well...almost everybody. Meet Bogworthy.
 - [Adventures of Aaron](#) - So full of action, it should be a verb. Meet characters and fans, see strips and live a day in Aaron's life.

Client-Side Examples

- ◆ sees all transactions from one client
 - personal history
 - active home page
 - off-line browser
 - information scout
 - form filler-outer
 - web travel scripting



Server-Side Examples



- ◆ sees all transactions for one server
 - usage statistics / user profiling
 - navigation: search, TOC, BOBI
 - site mapping
 - convert document formats
 - discussion groups
 - “look” control

Workgroup Examples

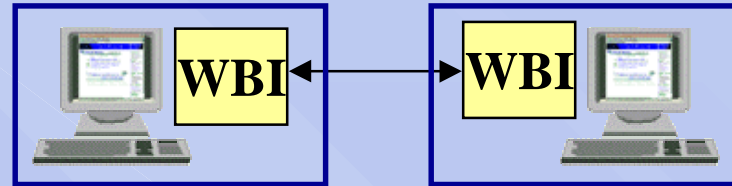
- ◆ sees all transactions for a group of users
 - all client-side applications
 - local access to server-side apps
 - community page rating
 - shared experience
 - custom data distillation



Multi-Platform Examples

◆ client-client

- active e-mail
- aglet docking



◆ server-client

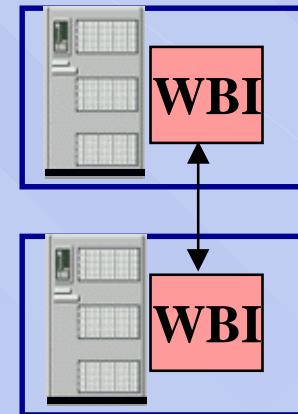
- user/vendor matching
- collaboration database

◆ workgroup-client

- proprietary protocols
- single for-fee gateway

◆ server-server

- index sharing
- virtual server



Conclusion

“So how does this affect me?”

How does this affect me?

- ◆ think in terms of intermediaries
 - they are everywhere
 - computational, human, physical
- ◆ check out the WBI demo and info
 - <http://www.cssrv.almaden.ibm.com/wbi/>
- ◆ develop your own WBI plugins!
 - barrett@almaden.ibm.com

