



University of
Southampton

Open Hypermedia

COMP3227 Web Architecture & Hypertext Technologies

Dr Heather Packer – hp3@ecs.soton.ac.uk

What is Open Hypermedia?

Hypermedia is the approach to building **information systems** made up of nodes of various media connected together by a collection of associative links

An **open** system generally implies that:

- There is some interface by which third party programs may access the functionality of the system.
- The system may be accessed from applications on heterogeneous architectures.

Open hypermedia: hypertext features present within whole environment

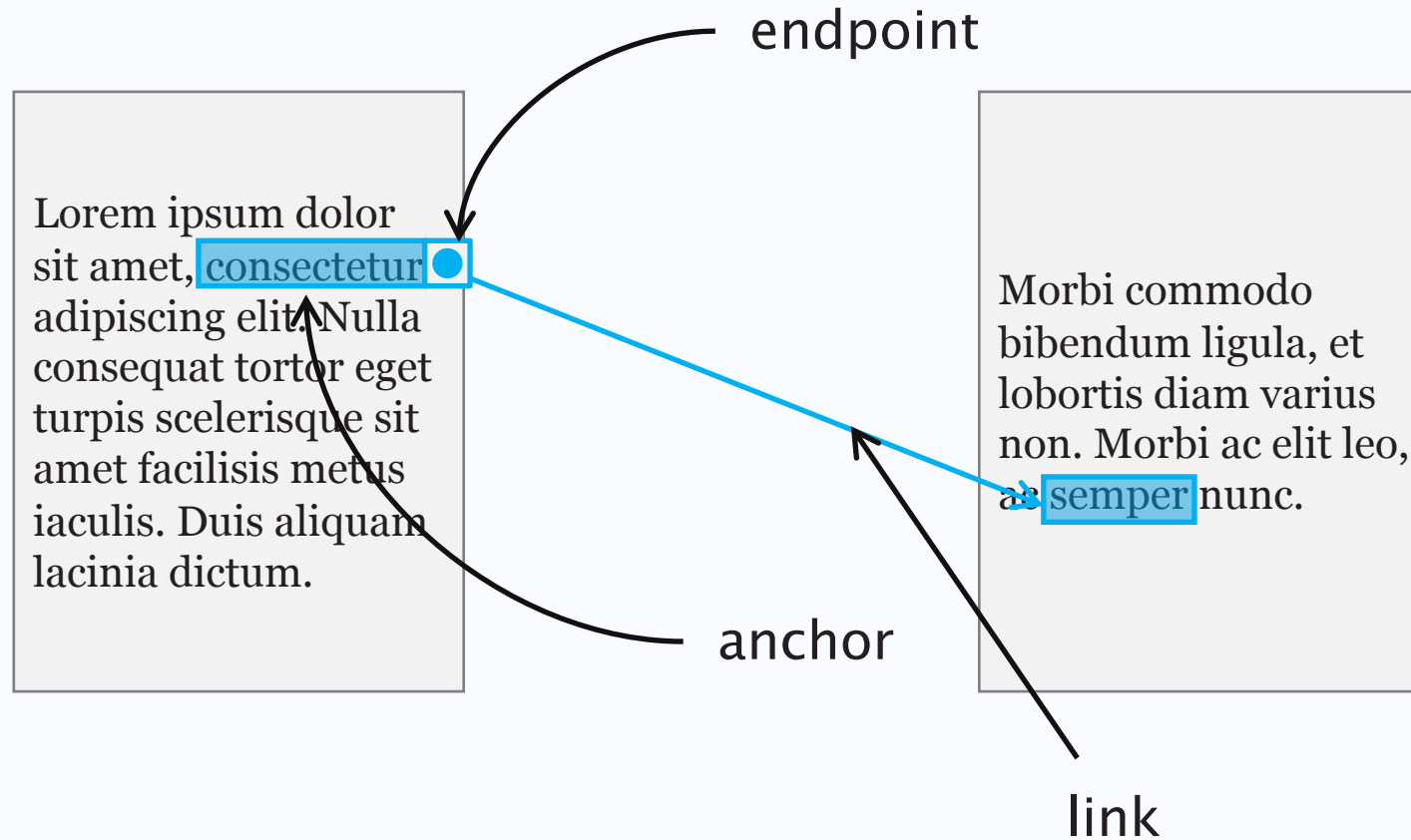
- Links and anchors are kept separately from documents (i.e. first-class links)
- Linkbases and link services

What is Open Hypermedia?

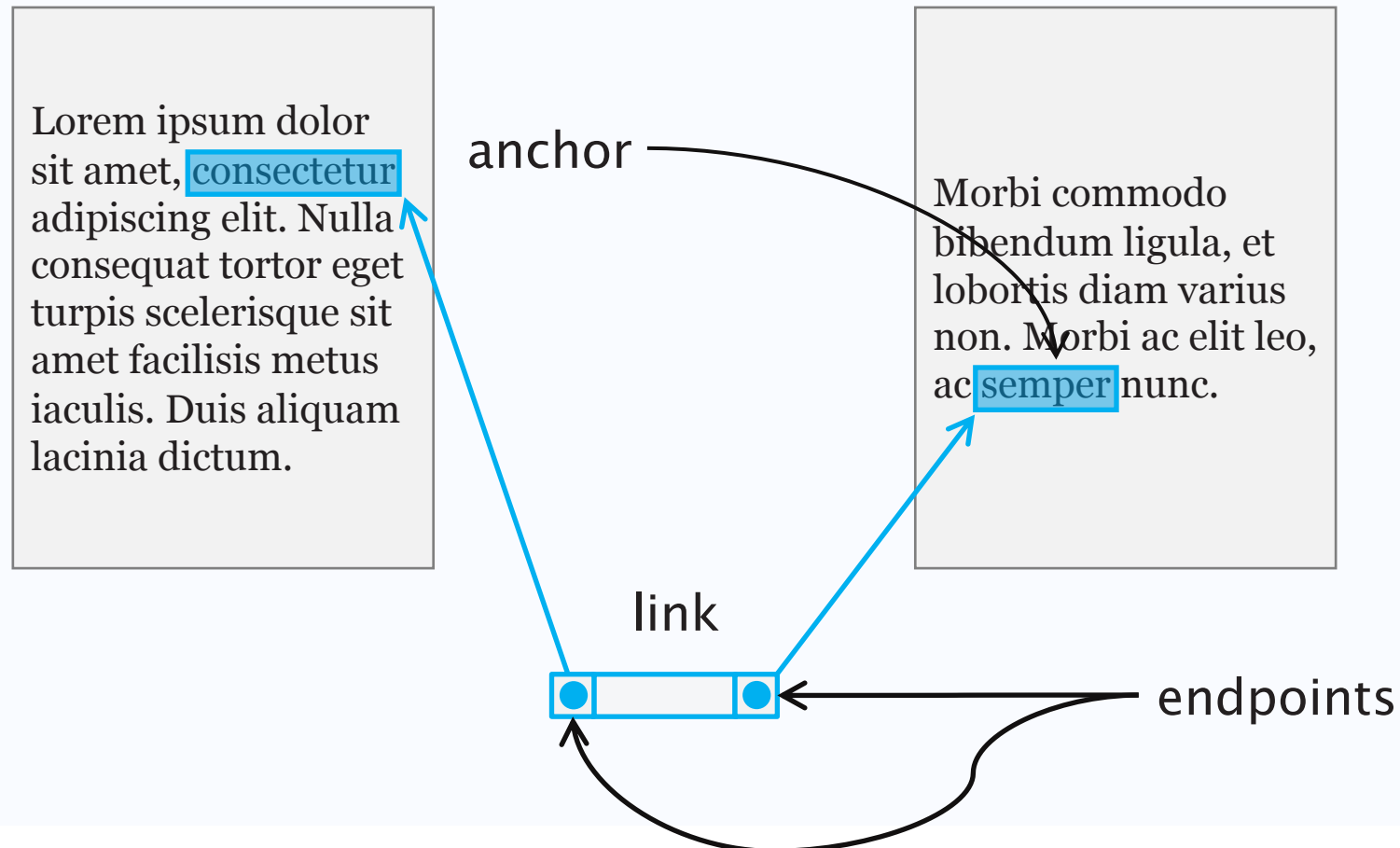
Open Hypermedia Systems are open with respect to:

- Applications
- Data formats (add links to any files)
- Functionality (configurable and extensible)
- Other OHS systems
- Platforms
- Users

Embedded links



First-class links





Dexter Hypertext Reference Model

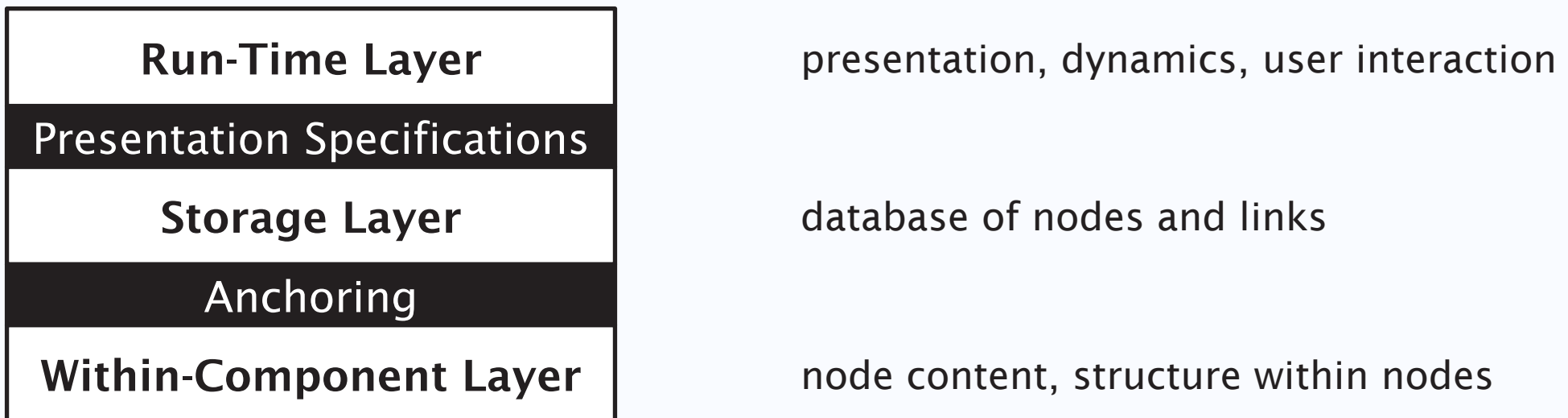
<http://www.flickr.com/photos/hunkdujour/186391965/>

Dexter Hypertext Reference Model (1988-1990)

Formal model of an open hypertext system, not an implementation

Used to compare functionalities of existing systems

Used to design new systems and develop standards for interoperability



Dexter storage layer

Three types of component: atoms, links and composites (sequences of components)

Key concept of a *specifier* – a robust description of a link endpoint

- Component specification, anchor, direction (to, from, bidirectional) and presentation spec
- Links as sequences of specifiers

Atom #3346

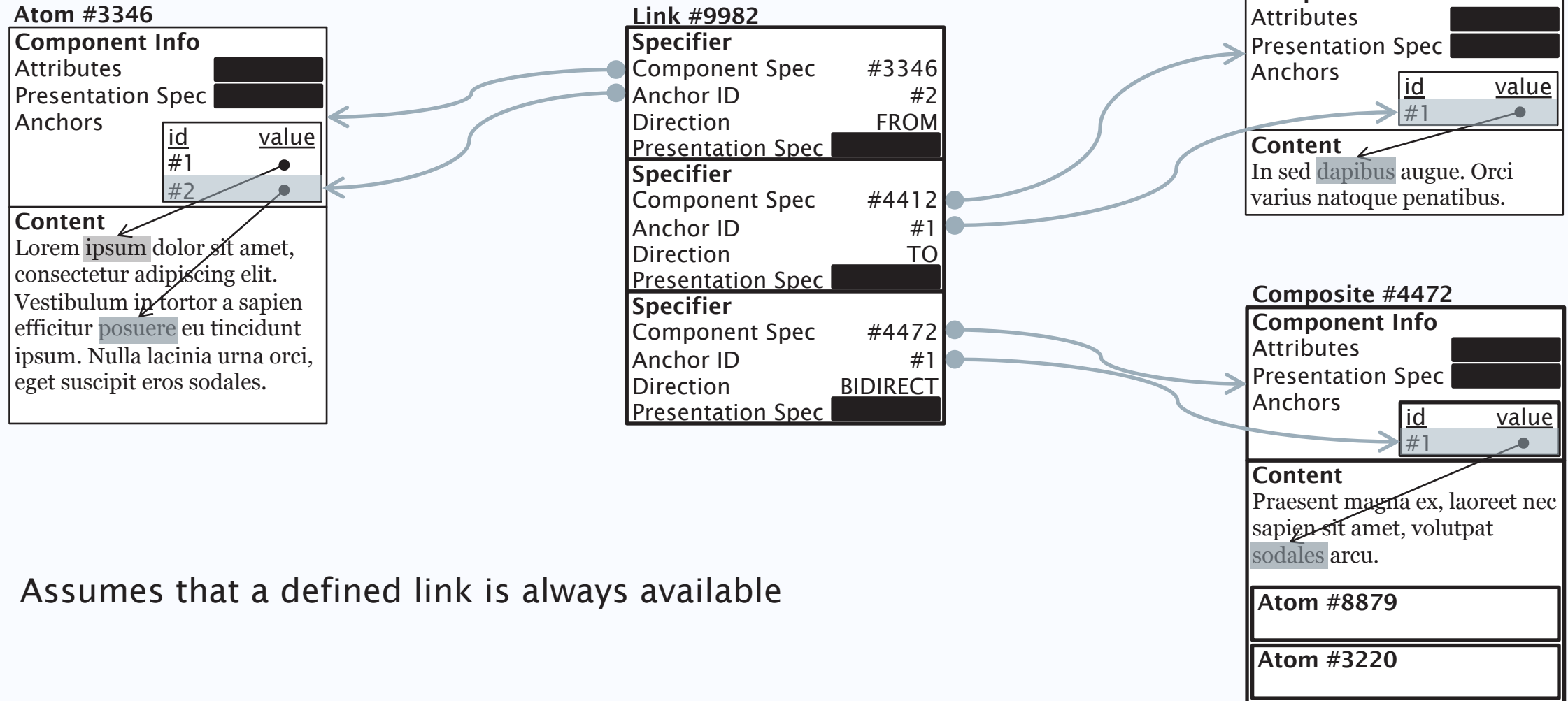
Component Info							
Attributes	████████						
Presentation Spec	████████						
Anchor	<table border="1"> <thead> <tr> <th>id</th> <th>value</th> </tr> </thead> <tbody> <tr> <td>#1</td> <td>●</td> </tr> <tr> <td>#2</td> <td>●</td> </tr> </tbody> </table>	id	value	#1	●	#2	●
id	value						
#1	●						
#2	●						
Content							
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum in tortor a sapien efficitur posuere eu tincidunt ipsum. Nulla lacinia urna orci, eget suscipit eros sodales.							

Link #9981

Specifier	
Component Spec	#3346
Anchor ID	#1
Direction	FROM
Presentation Spec	████████
Specifier	
Component Spec	#4412
Anchor ID	#1
Direction	TO
Presentation Spec	████████

Composite #4472

Component Info					
Attributes	████████				
Presentation Spec	████████				
Anchor	<table border="1"> <thead> <tr> <th>id</th> <th>value</th> </tr> </thead> <tbody> <tr> <td>#1</td> <td>●</td> </tr> </tbody> </table>	id	value	#1	●
id	value				
#1	●				
Content					
Praesent magna ex, laoreet nec sapien sit amet, volutpat sodales arcu.					
Atom #8879					
Atom #3220					



Assumes that a defined link is always available

Hyper-G

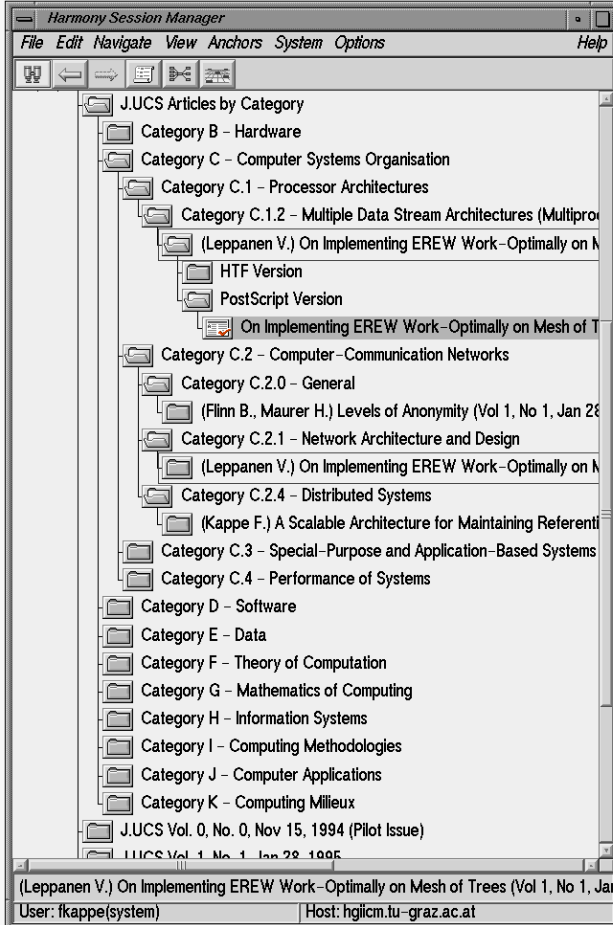
Hyper-G

Developed at the Technical University of Graz, Austria around 1989-1990

Features:

- Client-Server architecture (like the Web)
- Persistent session connections (unlike the Web?)
- First class nodes, links, anchors, composites and collections
- Bi-directional links
- Link integrity

Harmony client



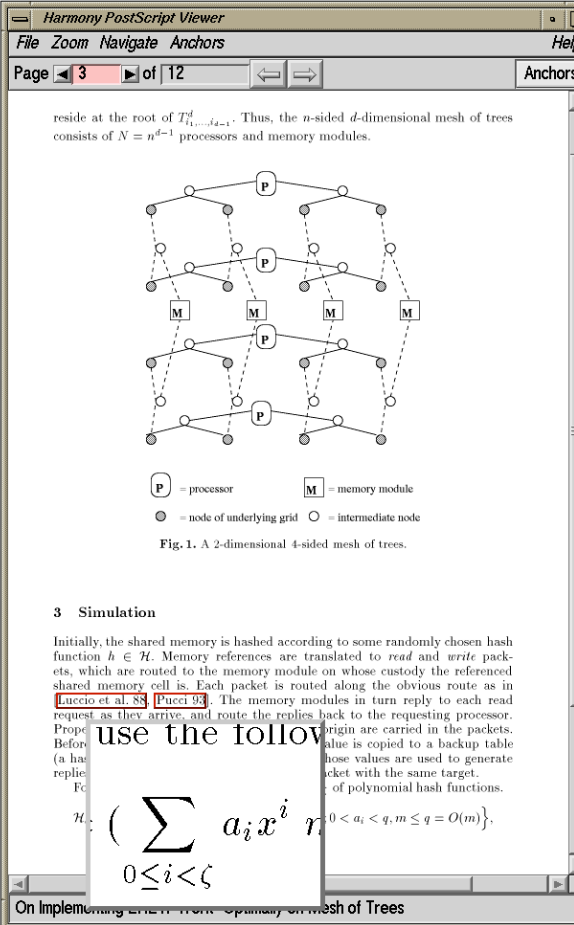
Harmony Session Manager

File Edit Navigate View Anchors System Options Help

- JUCS Articles by Category
 - Category B - Hardware
 - Category C - Computer Systems Organisation
 - Category C.1 - Processor Architectures
 - Category C.1.2 - Multiple Data Stream Architectures (Multipro...
 - (Leppanen V.) On Implementing EREW Work - Optimally on Mesh of T...
 - HTF Version
 - PostScript Version
 - On Implementing EREW Work - Optimally on Mesh of T...
 - Category C.2 - Computer-Communication Networks
 - Category C.2.0 - General
 - (Flinn B., Maurer H.) Levels of Anonymity (Vol 1, No 1, Jan 28...
 - Category C.2.1 - Network Architecture and Design
 - (Leppanen V.) On Implementing EREW Work - Optimally on Mesh of T...
 - Category C.2.4 - Distributed Systems
 - (Kappe F.) A Scalable Architecture for Maintaining Referent...
 - Category C.3 - Special-Purpose and Application-Based Systems
 - Category C.4 - Performance of Systems
 - Category D - Software
 - Category E - Data
 - Category F - Theory of Computation
 - Category G - Mathematics of Computing
 - Category H - Information Systems
 - Category I - Computing Methodologies
 - Category J - Computer Applications
 - Category K - Computing Milieux
 - JUCS Vol. 0, No. 0, Nov 15, 1994 (Pilot Issue)
 - JUCS Vol. 1, No. 1, Jan 28, 1995

(Leppanen V.) On Implementing EREW Work - Optimally on Mesh of Trees (Vol 1, No 1, Jan 28, 1995)

User: fkappe(system) Host: hgjicm.tu-graz.ac.at

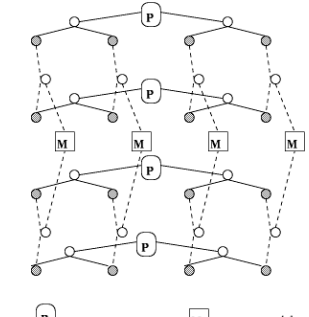


Harmony PostScript Viewer

File Zoom Navigate Anchors Help

Page 3 of 12

reside at the root of $T_{i_1, \dots, i_{d-1}}^d$. Thus, the n -sided d -dimensional mesh of trees consists of $N = n^{d-1}$ processors and memory modules.



P = processor M = memory module
 = node of underlying grid = intermediate node

Fig. 1. A 2-dimensional 4-sided mesh of trees.

3 Simulation

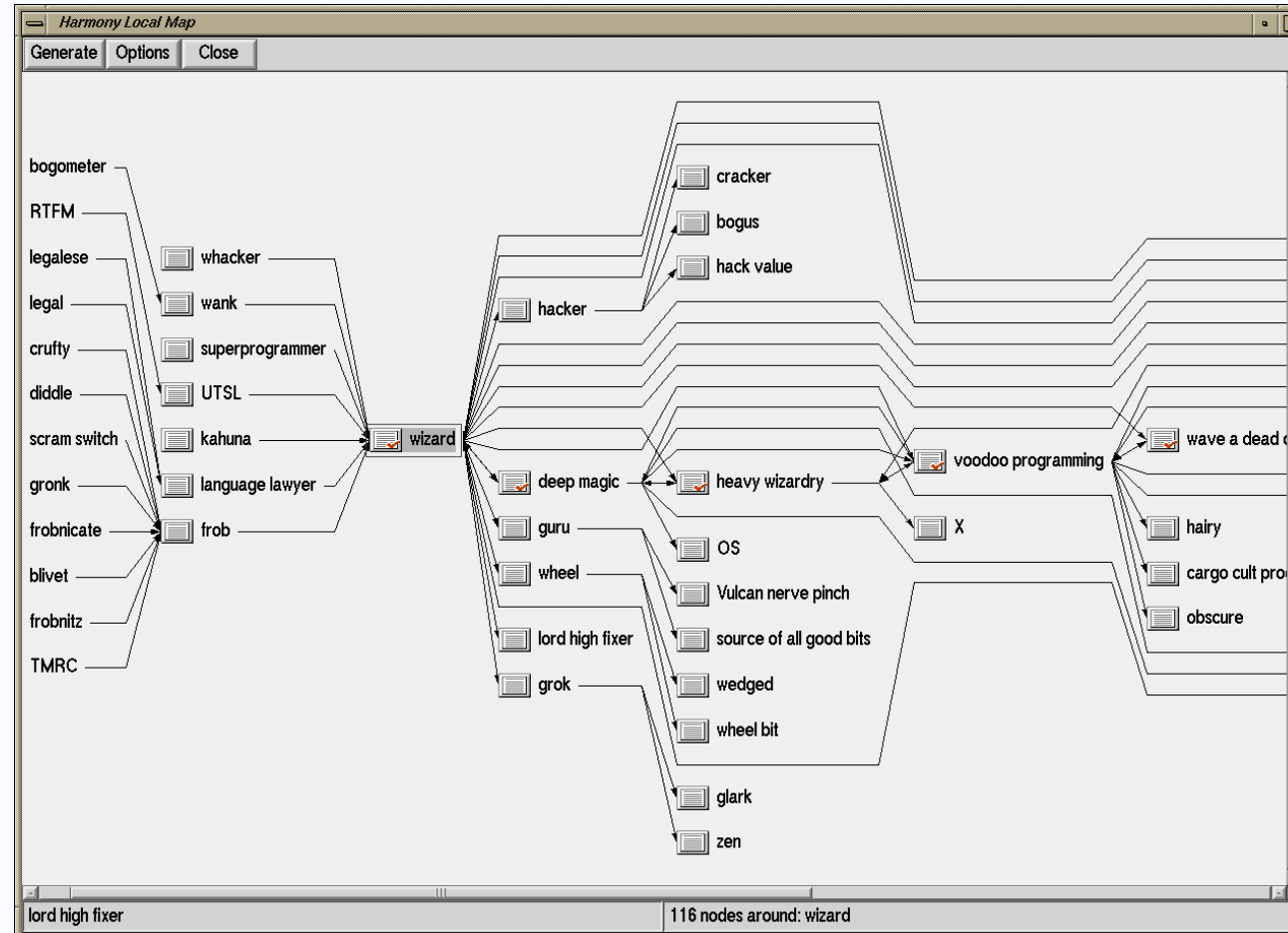
Initially, the shared memory is hashed according to some randomly chosen hash function $h \in \mathcal{H}$. Memory references are translated to *read* and *write* packets, which are routed to the memory module on whose custody the referenced shared memory cell is. Each packet is routed along the obvious route as in [Luccio et al. 88](#), [Pucci 93](#). The memory modules in turn reply to each read request as they arrive, and route the replies back to the requesting processor. Proper origin are carried in the packets. Before these values are used to generate a packet with the same target, a value is copied to a backup table of polynomial hash functions.

use the following

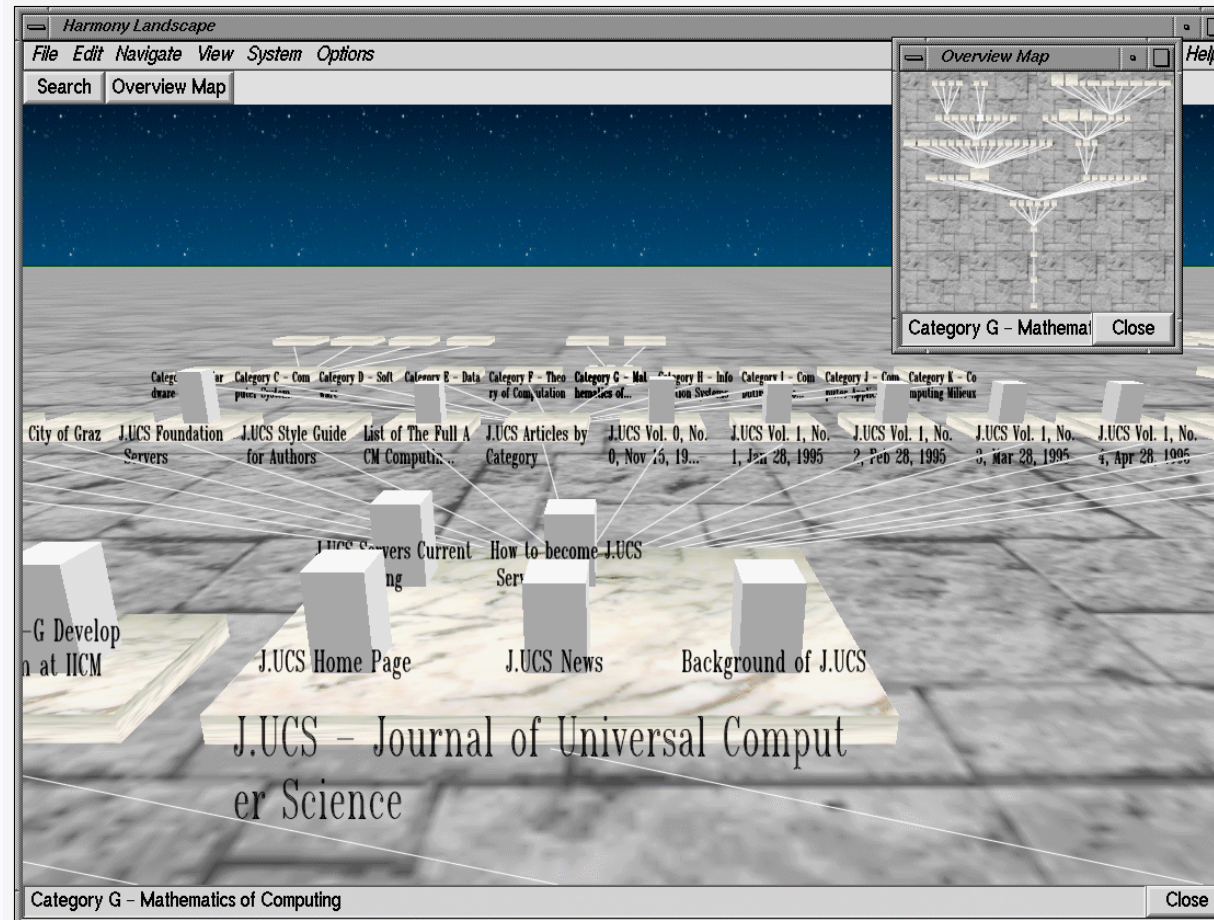
$$\mathcal{H} = \left\{ \left(\sum_{0 \leq i < \zeta} a_i x^i \right) \mid 0 < a_i < q, m \leq q = O(m) \right\}$$

On Implementing EREW Work - Optimally on Mesh of Trees

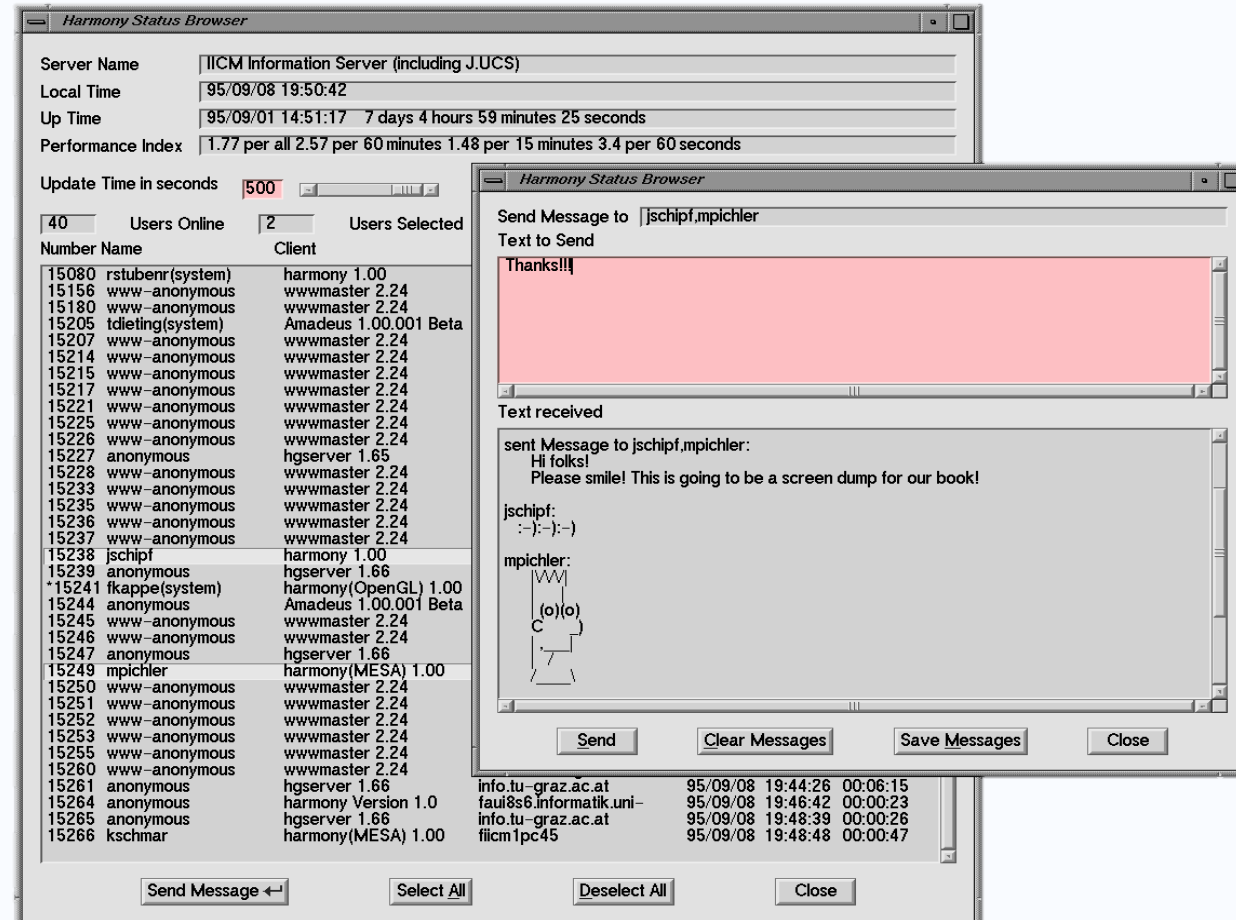
Harmony client: link structure view



Harmony client: landscape view



Harmony client: client-client communications



The screenshot displays the Harmony Status Browser interface. The main window shows server information and a list of users. A secondary window is open for sending a message to a specific user.

Harmony Status Browser - Server Information:

- Server Name: IICM Information Server (including J.UCS)
- Local Time: 95/09/08 19:50:42
- Up Time: 95/09/01 14:51:17 7 days 4 hours 59 minutes 25 seconds
- Performance Index: 1.77 per all 2.57 per 60 minutes 1.48 per 15 minutes 3.4 per 60 seconds
- Update Time in seconds: 500
- Users Online: 40
- Users Selected: 2

Harmony Status Browser - User List:

Number	Name	Client
15080	rstuberr(system)	harmony 1.00
15156	www-anonymous	wwwmaster 2.24
15180	www-anonymous	wwwmaster 2.24
15205	tdieting(system)	Amadeus 1.00.001 Beta
15207	www-anonymous	wwwmaster 2.24
15214	www-anonymous	wwwmaster 2.24
15215	www-anonymous	wwwmaster 2.24
15217	www-anonymous	wwwmaster 2.24
15221	www-anonymous	wwwmaster 2.24
15225	www-anonymous	wwwmaster 2.24
15226	www-anonymous	wwwmaster 2.24
15227	anonymous	hgserver 1.65
15228	www-anonymous	wwwmaster 2.24
15233	www-anonymous	wwwmaster 2.24
15235	www-anonymous	wwwmaster 2.24
15236	www-anonymous	wwwmaster 2.24
15237	www-anonymous	wwwmaster 2.24
15238	jschopf	harmony 1.00
15239	anonymous	hgserver 1.66
*15241	fkappe(system)	harmony(OpenGL) 1.00
15244	anonymous	Amadeus 1.00.001 Beta
15245	www-anonymous	wwwmaster 2.24
15246	www-anonymous	wwwmaster 2.24
15247	anonymous	hgserver 1.66
15249	mpichler	harmony(MESA) 1.00
15250	www-anonymous	wwwmaster 2.24
15251	www-anonymous	wwwmaster 2.24
15252	www-anonymous	wwwmaster 2.24
15253	www-anonymous	wwwmaster 2.24
15255	www-anonymous	wwwmaster 2.24
15260	www-anonymous	wwwmaster 2.24
15261	anonymous	hgserver 1.66
15264	anonymous	harmony Version 1.0
15265	anonymous	hgserver 1.66
15266	kschmar	harmony(MESA) 1.00

Harmony Status Browser - Message Window:

Send Message to: jschopf.mpichler

Text to Send: Thanks!!!

Text received:

```
sent Message to jschopf.mpichler:
Hi folks!
Please smile! This is going to be a screen dump for our book!

jschopf:
:-):-):-)

mpichler:
  W
  (o)(o)
  C
  T
  \
```

Buttons: Send, Clear Messages, Save Messages, Close

Main Window Buttons: Send Message ←, Select All, Deselect All, Close

Log Window (Bottom Right):

Client	Time	Duration
info.tu-graz.ac.at	95/09/08 19:44:26	00:06:15
faii8s6.informatik.uni-	95/09/08 19:46:42	00:00:23
info.tu-graz.ac.at	95/09/08 19:48:39	00:00:26
iicm1pc45	95/09/08 19:48:48	00:00:47

Link integrity

Users should always be able to follow any link that is presented to them

- When documents are moved, edited or deleted, links must be updated
- Compare with the Web: if the destination of a link goes away, the user sees 404 Not Found

Hyper-G divides links into two categories:

- *Core links* relate documents stored on the same server
- *Surface links* relate documents stored on different servers

Changes that affect core links can be processed by a single server

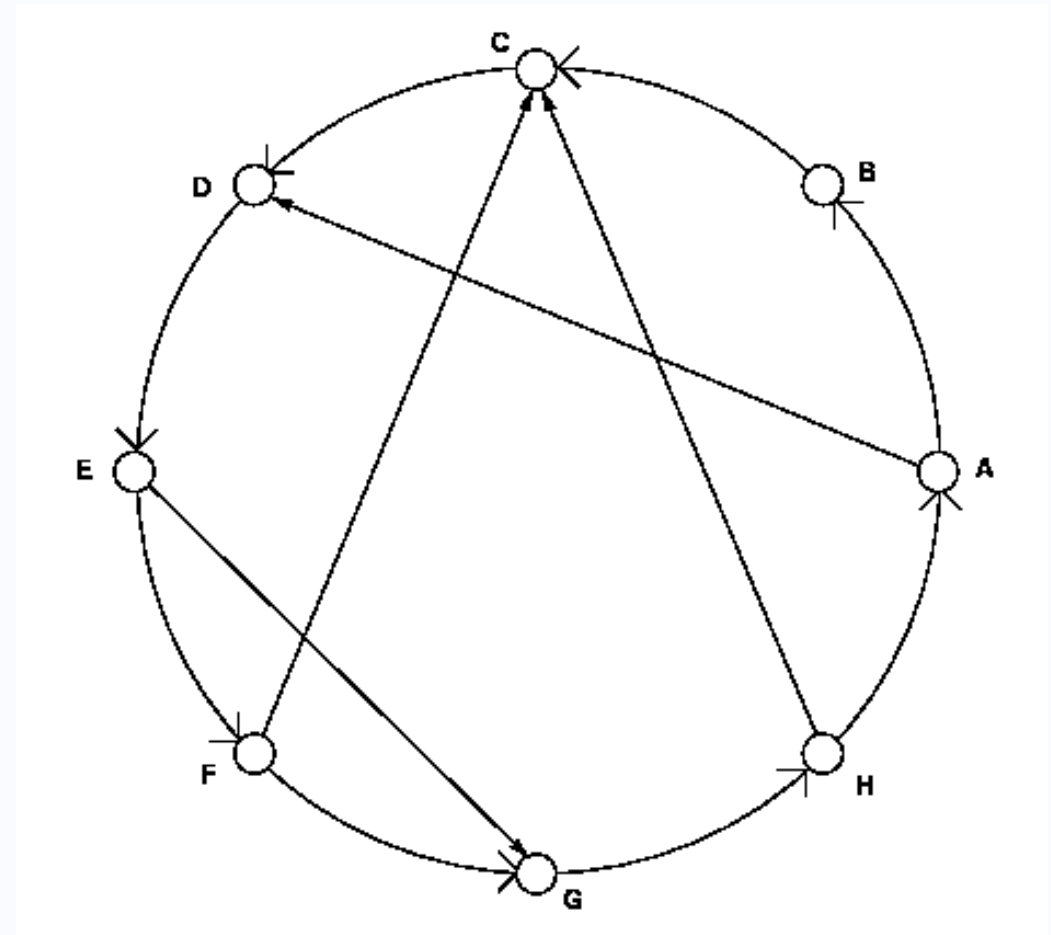
Changes that affect surface links involve multiple servers that need to be notified

P-flood Algorithm

Broadcasting surface link updates to all affected servers is expensive

Probabilistic flooding algorithm used to send surface link updates more efficiently

- Servers arranged in a ring
- Servers send link updates to their immediate successor and randomly to other servers
- Scalable and robust
- Parameterisable – number of additional servers can be altered



Evaluation

Advantages

- Authoring support integrated into browser, and designed into protocols from outset
- Support for collaboration
- Early support for multimedia

Disadvantages

- Flooding algorithm would probably not scale to the Web
(but some similarities with peer-to-peer approaches like distributed hash tables)

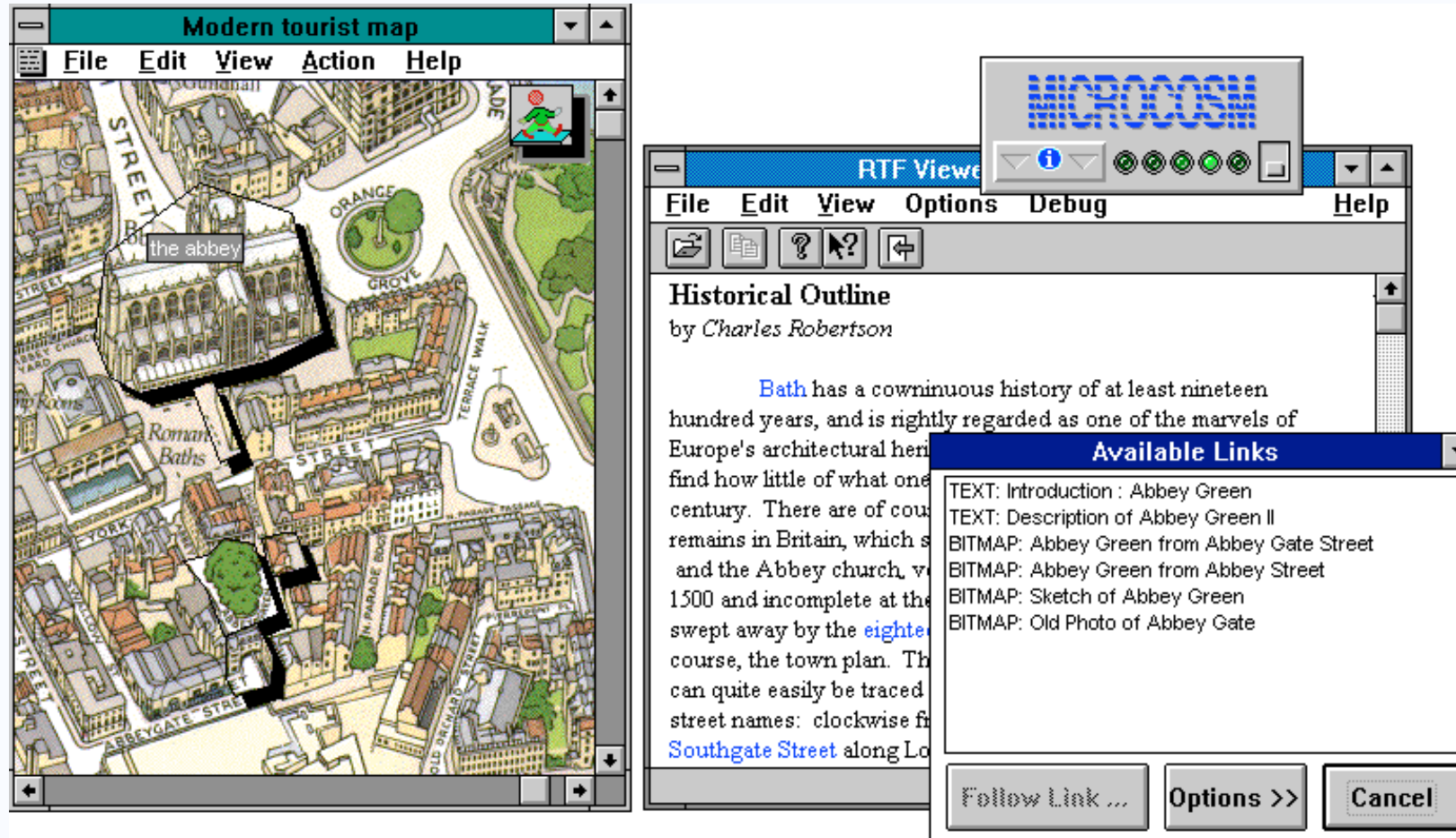
Microcosm

Overview

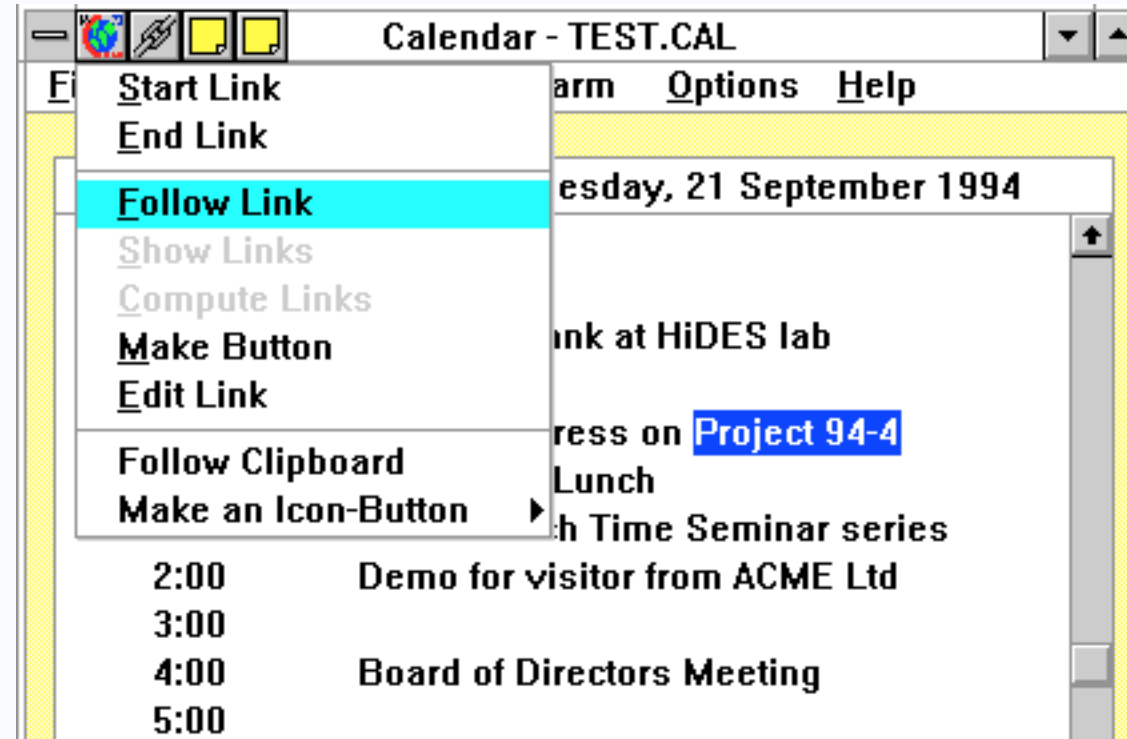
Developed at the University of Southampton around 1989

- Originally designed for use with read-only media (laser discs and later CD-ROMs)
- Originally designed as a desktop-based system, later expanded to a distributed system
- Mountbatten's personal papers were gifted to the University of Southampton Library
 - \approx 250,000 papers
 - \approx 50,000 photographs
- Informed much subsequent work within ECS: DLS, COHSE, etc

Microcosm client

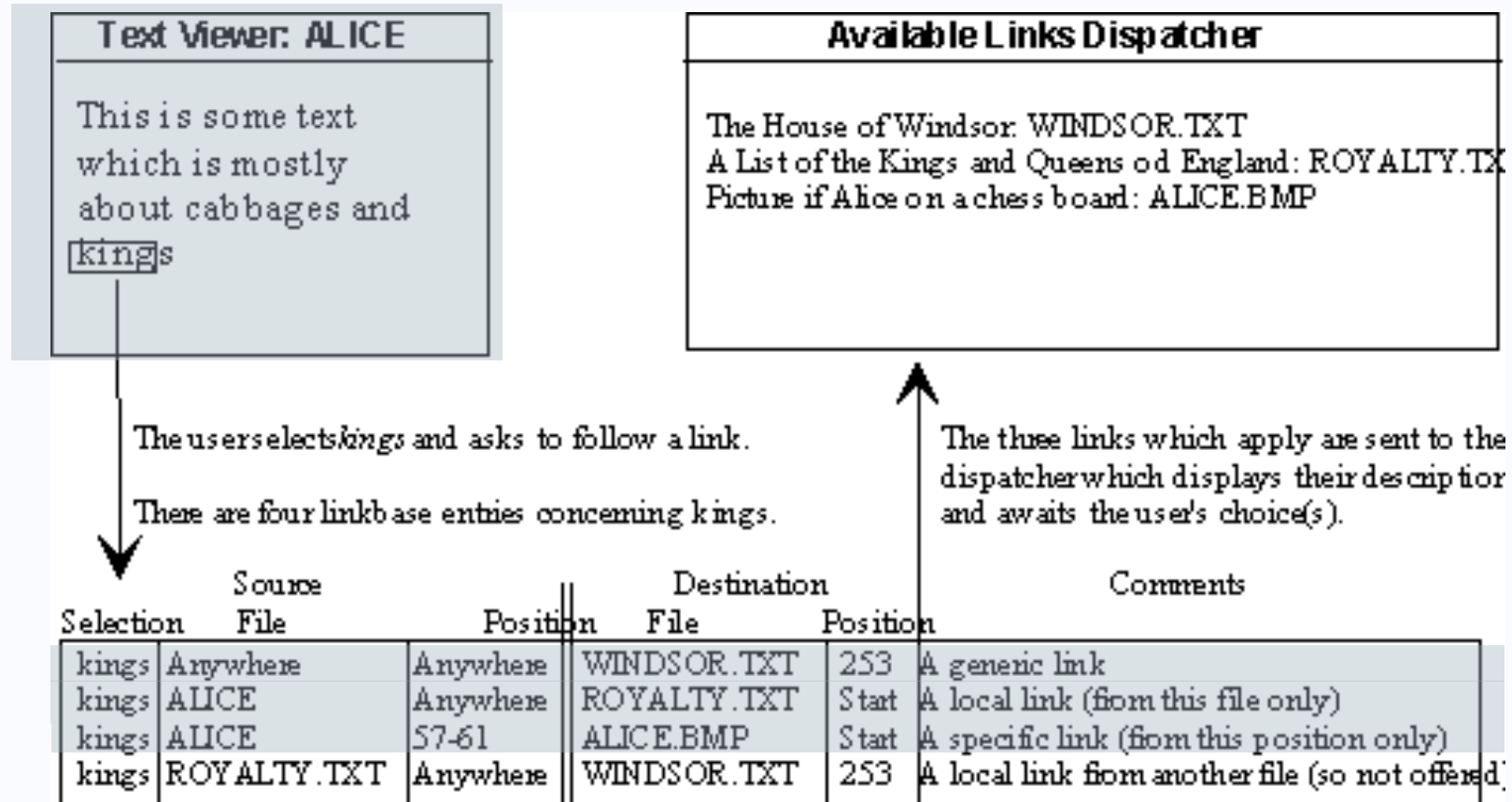


Microcosm universal viewer



Generic, Specific and Local links

Position of source anchor held in the linkbase

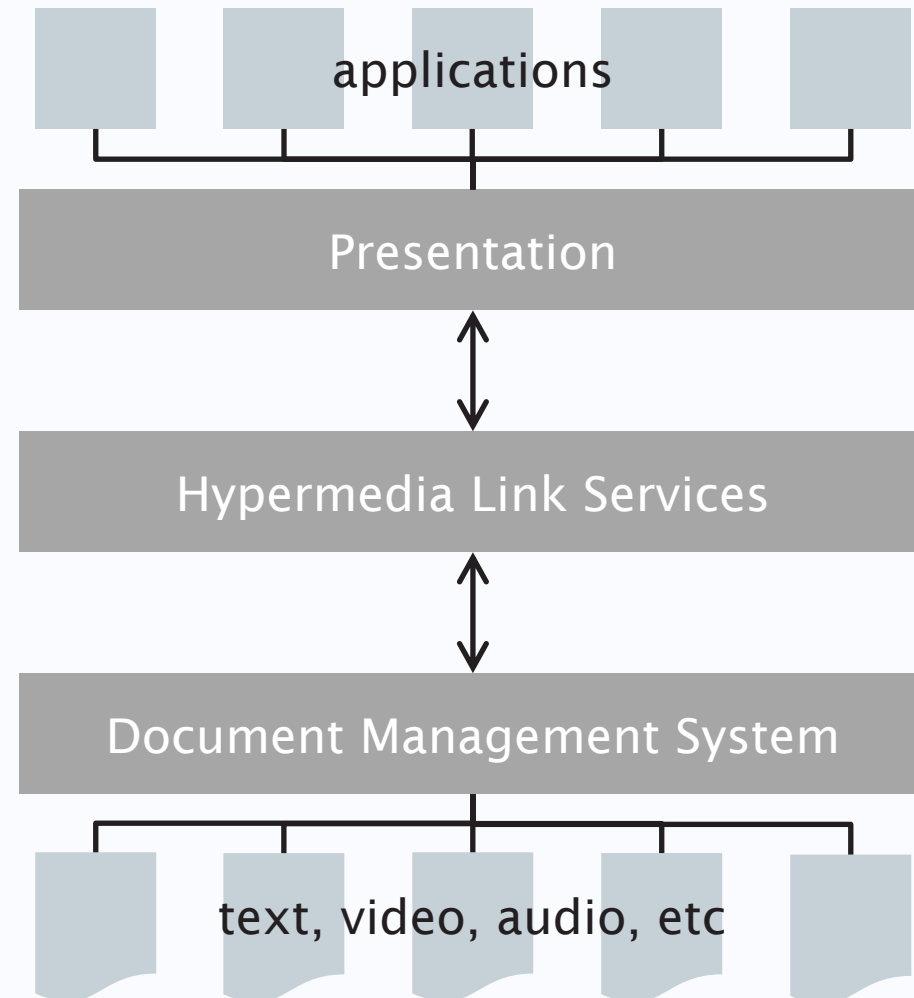


Microcosm architecture

Separation of concerns:

- Document storage and management
- Link storage and management
- Applications and presentation

Link services mediate the other components

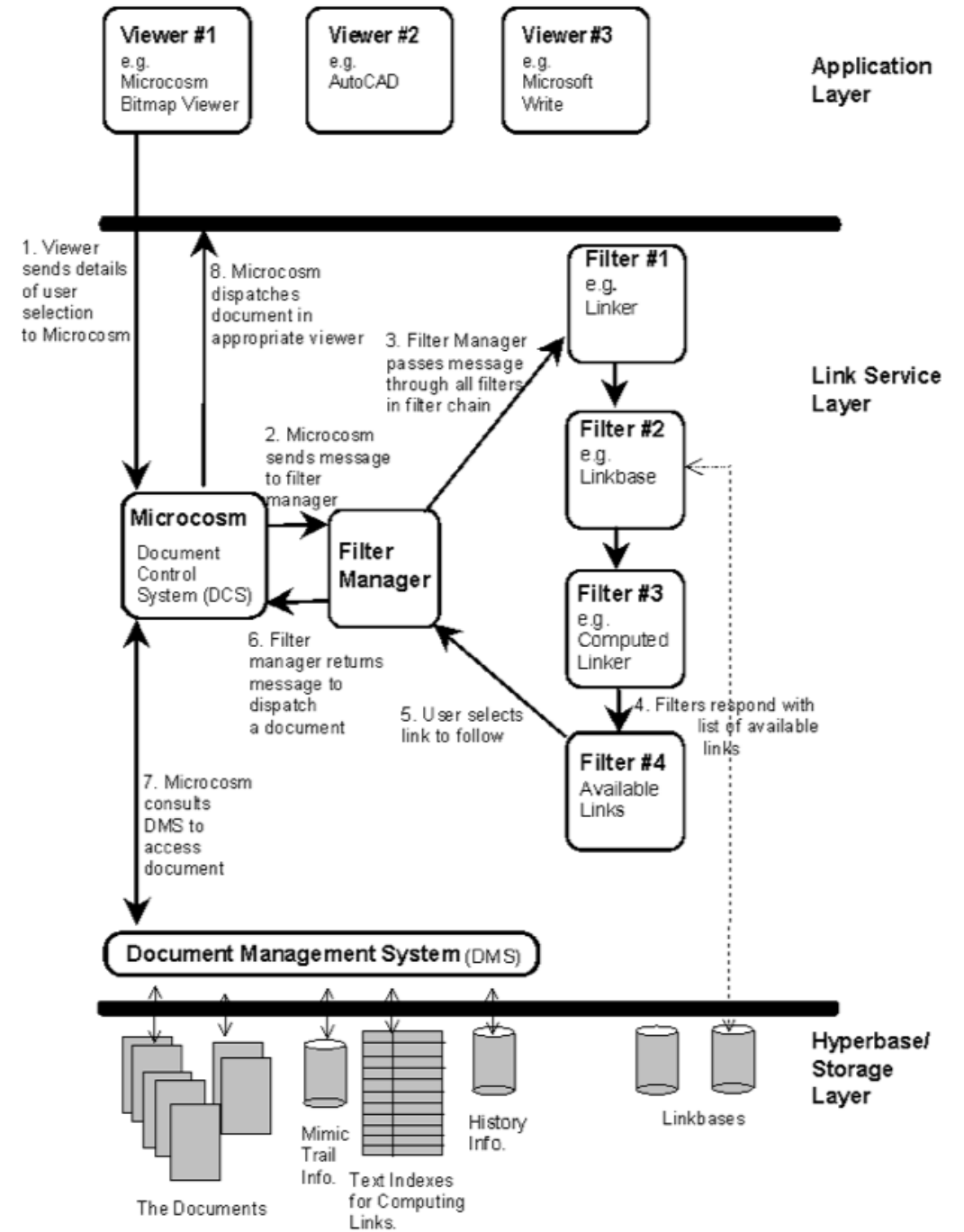


Microcosm architecture

Link service comprises a sequence of *filters*

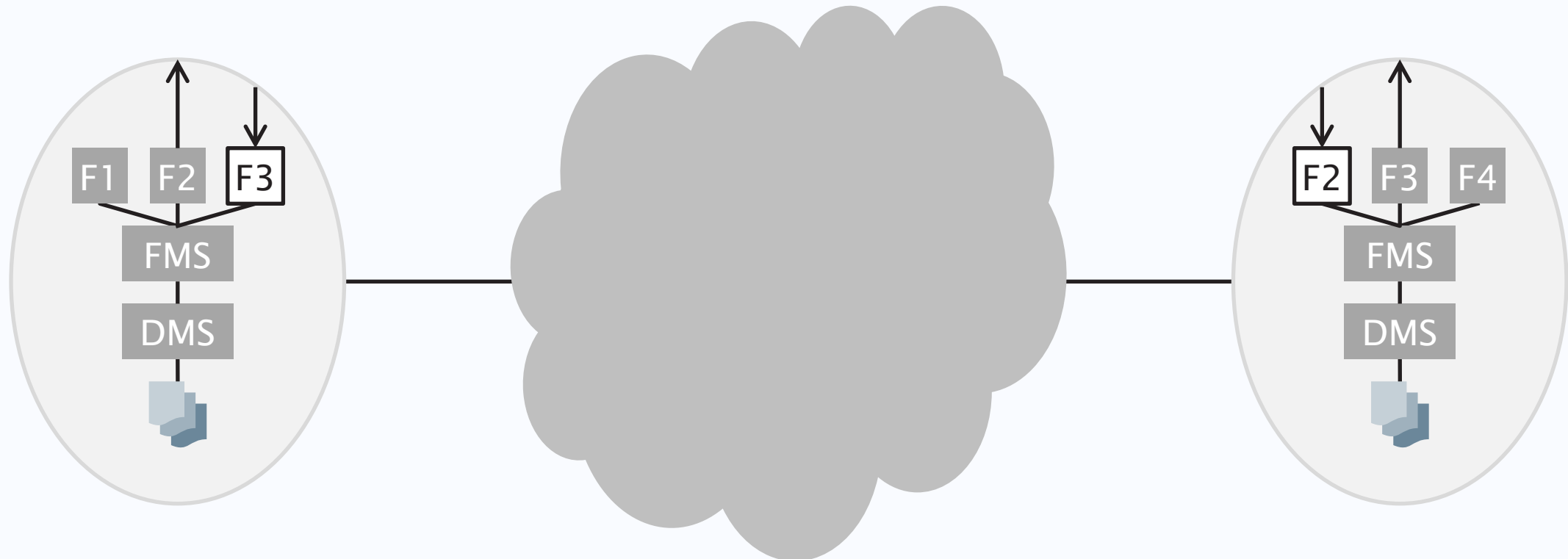
Filters are composable services that generate or manipulate links

Filter manager organises filters into chains



Distribution

Microcosm instances may publish filters to be used by other instances



Evaluation

Advantages

- Rich model of linking
(generic links, n-ary links, etc)
- Flexible document processing
(multiple linkbases)
- Integration with third-party applications

Disadvantages

- Poor scalability
Intended for workgroups to enterprises
- Distribution not intended from the outset
- Arguably, no native document format
- No support for link integrity

Open Hypermedia Protocol

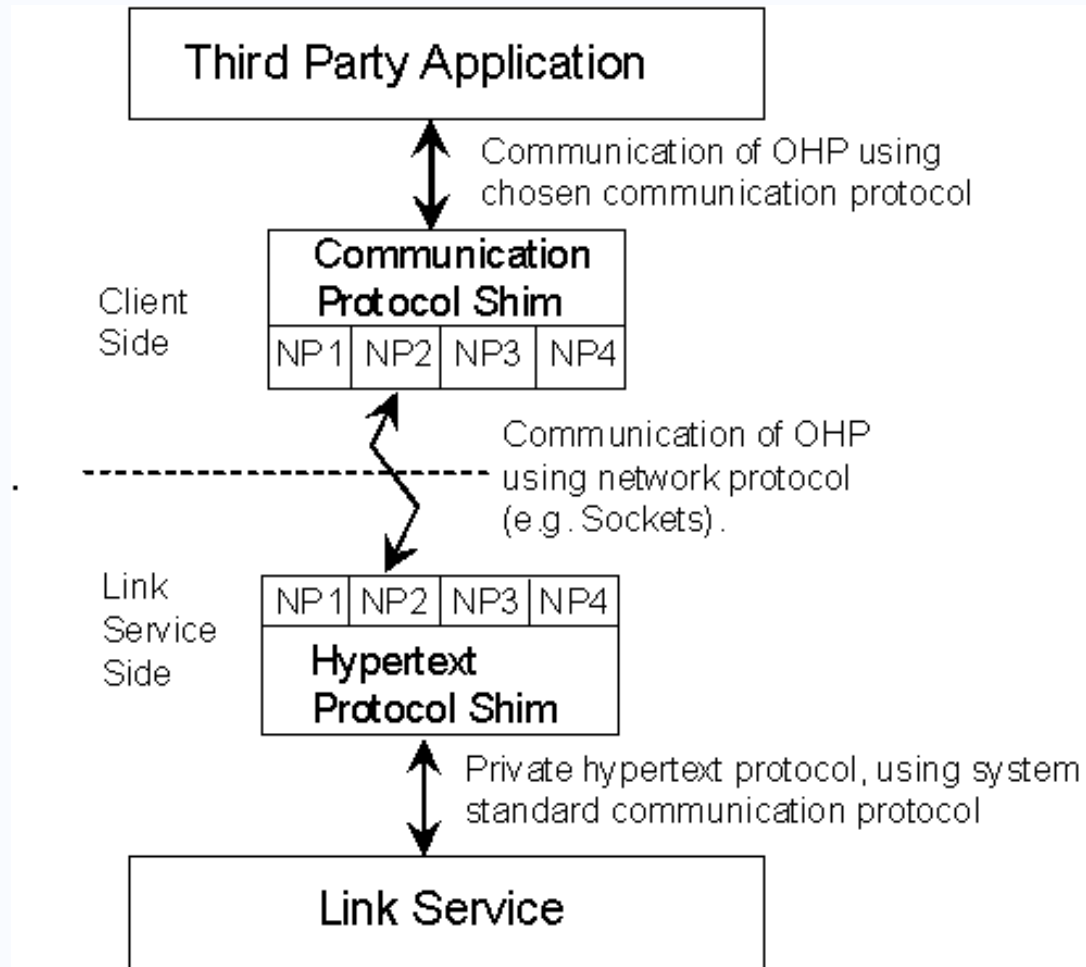
Open Hypermedia Protocol

- Enables client side applications to communicate with server side programs
- Achieved by providing some 'shim' program that translates between OHP and the linkserver's native protocol
- Applications would be able to work with all linkservices.

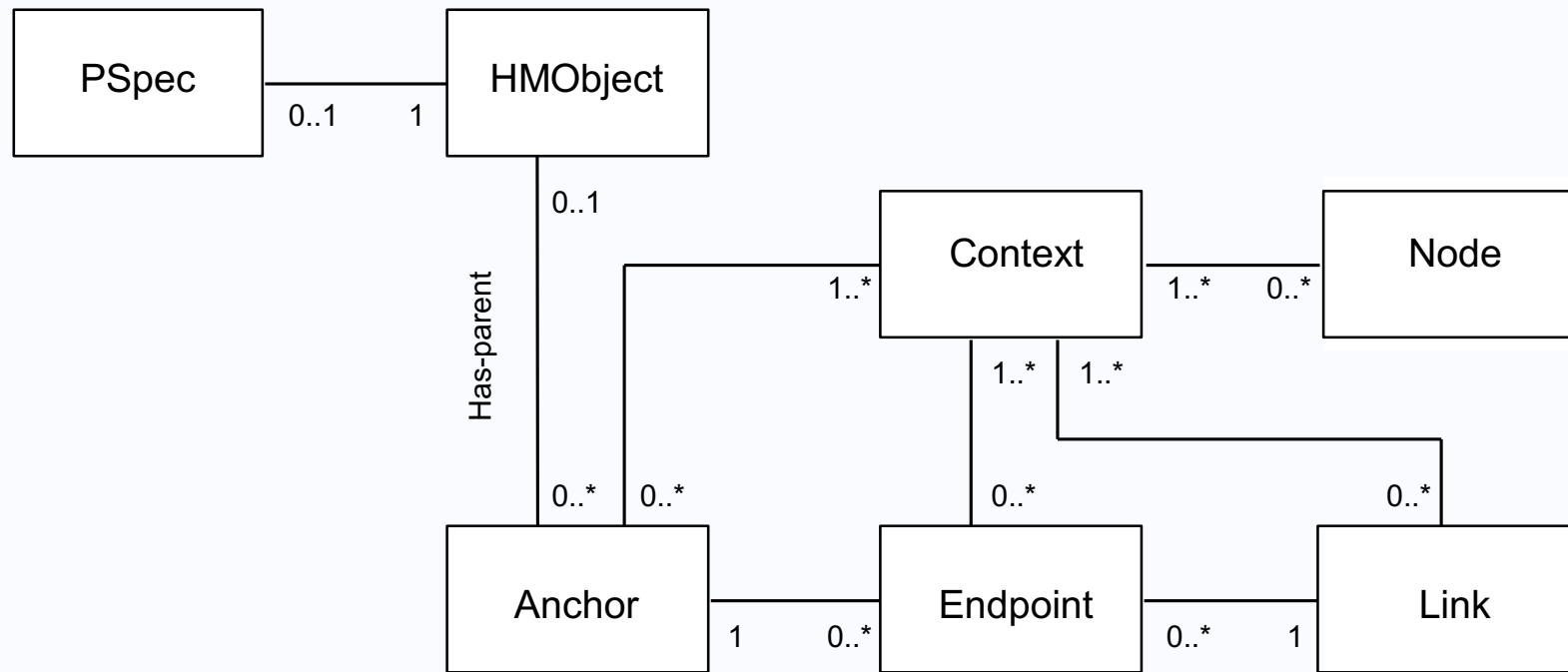
Was not aimed at web scale

OHP formed the basis for later integration efforts (FOHM)

OHP architecture



OHP data model



Location specifiers (LocSpec)

LocSpecs holds information about the position of an anchor

- Byte offset within file
- Occurrence of a given string (i.e. generic link)
- Named location within content (c.f. HTML id attribute)

Generalisation of Dexter's anchor values

But also:

- Can also exist within links
- Function to evaluate to identify anchor position

Evaluation

Advantages

- Used in several demos and prototypes
- Commonly-accepted data model

Disadvantages

- Shim architecture was naïve (expensive to implement)
- High message overhead



Link Integrity

<http://www.flickr.com/photos/andreaswinterer/2460423132/>

Link integrity

The endpoint of a link (source or destination) needs to define:

- a node
- (optionally) a position within a node (locspec)

If an endpoint fails to resolve to the place intended by the author, then it (and the link) is broken

Two types of link integrity failure:

- Dangling link
- Content reference

The dangling link problem

Occurs when an endpoint refers to an invalid node

Commonly the result of a node being moved or deleted

Possible mitigations:

- Responsibility of link creators (current practice on the Web)
- Don't allow links to things that move
- Forward references (i.e. redirects to new location)
- Guaranteed names (PURL servers, DMS)
- The Hyper-G approach (i.e. propagate changes affecting links)
- Link integrity checking agent (Spider)

The content reference problem

Occurs when an endpoint refers to a valid node, but to an incorrect location within that node

Commonly the result of changes/updates to the content of a node that are not reflected in links to that node

Possible mitigations:

- The publishing model (i.e. published resources are read-only, editing creates new resources)
- Manual link editor
- Link service-aware editing tools
- Just-in-time link repairs
- Express specific link positions using queries
- Avoid specific links/anchors
- Versioning (c.f. publishing model)
- Use of diff files (c.f. JIT link repairs)

Who is Responsible, link owner or system?

- Don't bother: it's a social issue.
- Avoid the problem: use declarative link definitions
- Loosely coupled: give the author tools to sort the problem if they want
- Automated link repairs: fix problems as they're encountered
- Tightly coupled: don't let users have this freedom

Evaluating open hypermedia

Advantages

- Applications not responsible for maintaining "foreign" markup
- Tailor linkbases to user needs (contexts)
- Generic links, etc
- Necessary for linking read-only media (e.g. CD-ROM, no permission)

Disadvantages

- Keeping links separately introduces potential consistency issues
- Integration with existing applications can be difficult

Next: Open Hypermedia on the Web