UNIVERSITY OF
Southampton

Open Hypermedia

COMP3220 Web Infrastructure

Dr Nicholas Gibbins – nmg@ecs.soton.ac.uk

What is Open Hypermedia?

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

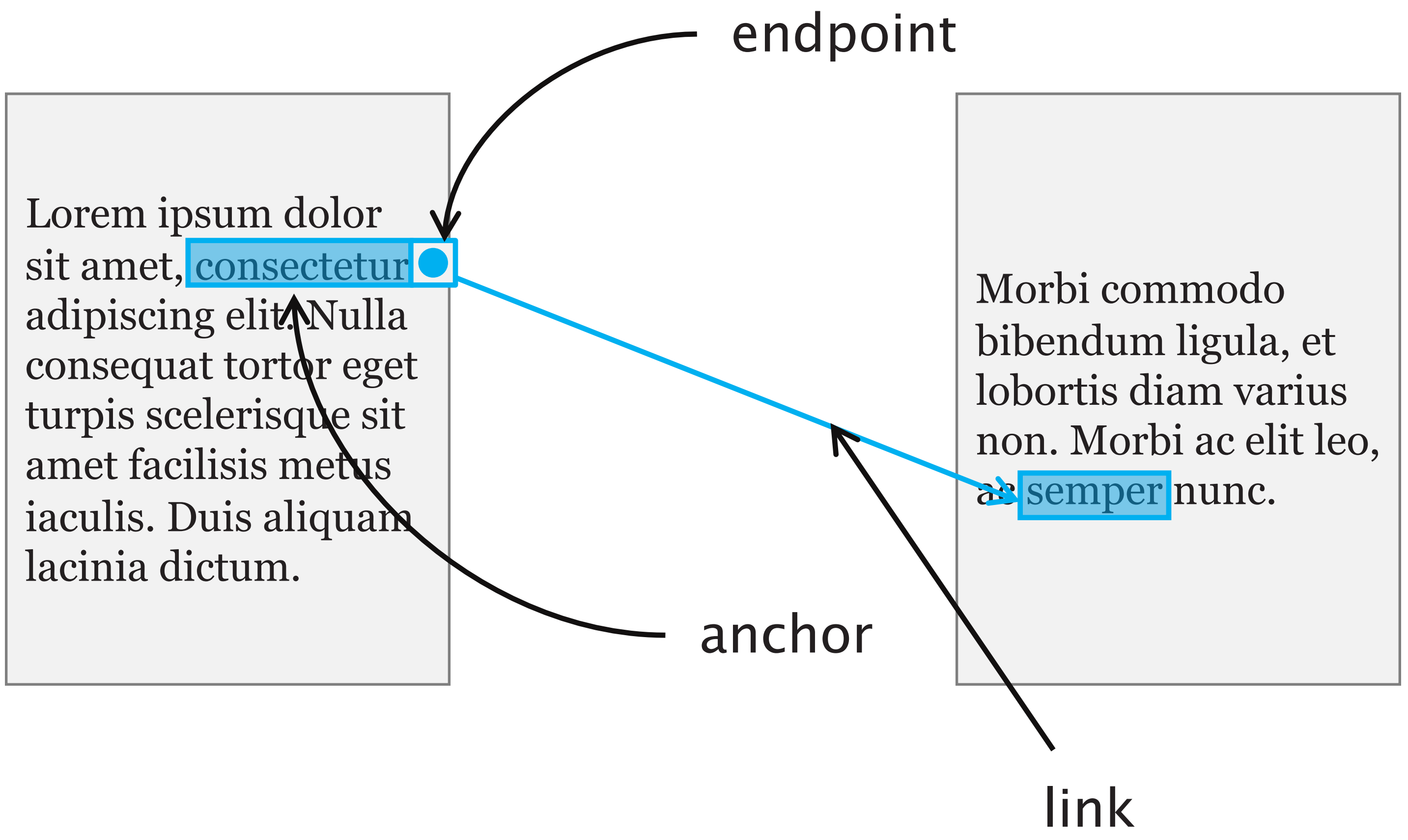
- Links and anchors are kept separately from documents (i.e. first-class links)
- Linkbases and link services

What is Open Hypermedia?

Open Hypermedia Systems are open with respect to:

- Applications
- Data formats
- Functionality (configurable and extensible)
- Other OHS systems
- Platforms
- Users

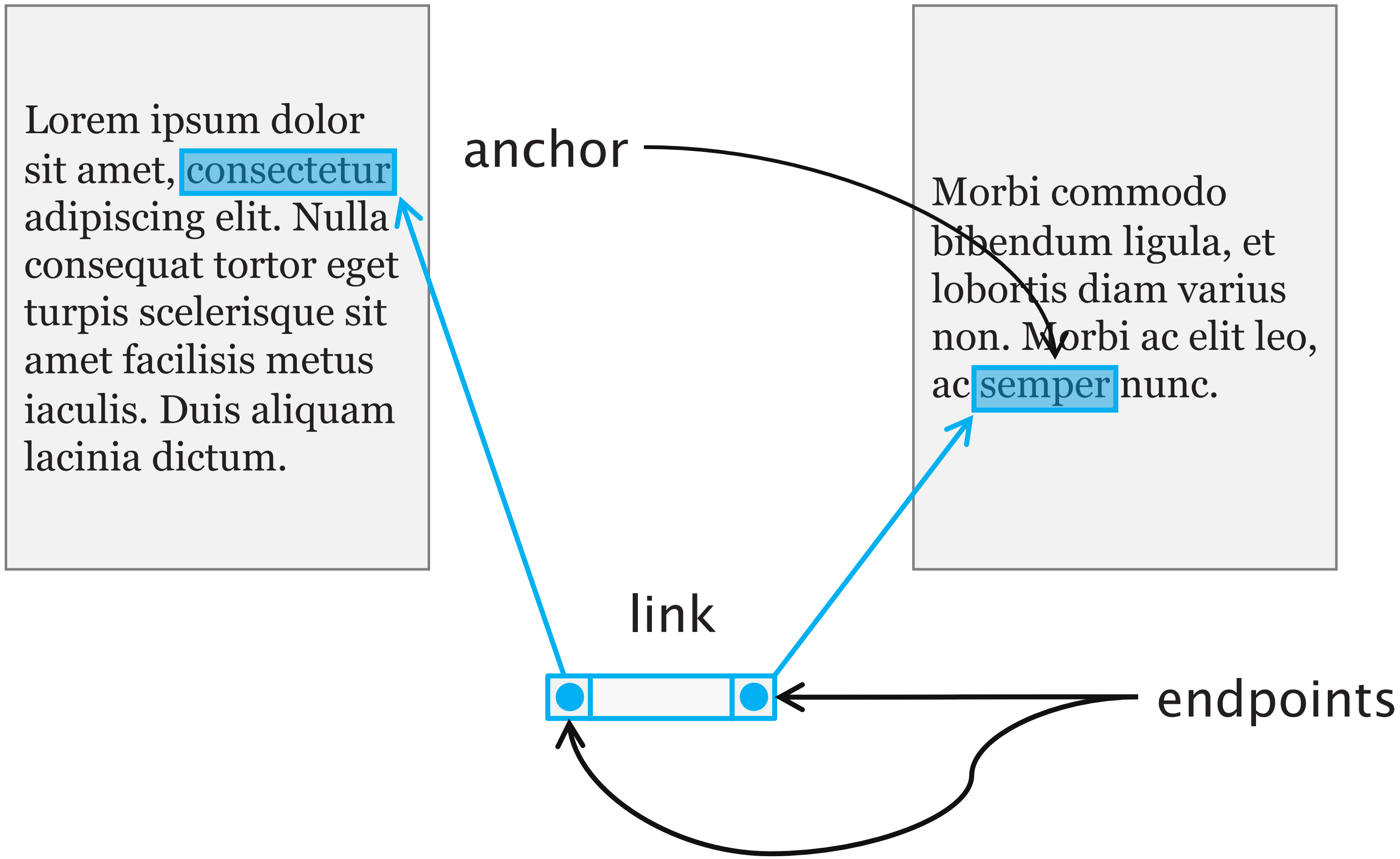
Embedded links



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla consequat tortor eget turpis scelerisque sit amet facilisis metus iaculis. Duis aliquam lacinia dictum.

Morbi commodo bibendum ligula, et lobortis diam varius non. Morbi ac elit leo, ac semper nunc.

First-class links



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla consequat tortor eget turpis scelerisque sit amet facilisis metus iaculis. Duis aliquam lacinia dictum.

anchor

Morbi commodo bibendum ligula, et lobortis diam varius non. Morbi ac elit leo, ac semper nunc.

link

endpoints

Dexter Hypertext Reference Model

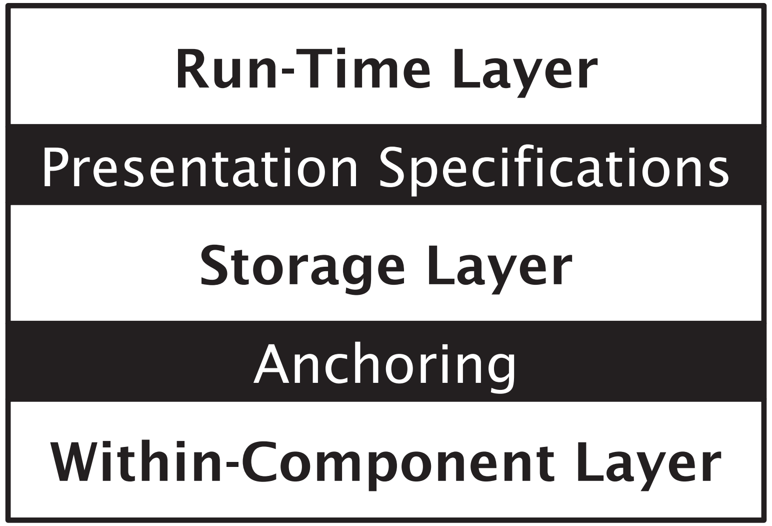
http://www.flickr.com/photos/hunkdujour/186391965/

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



Run-Time Layer

Presentation Specifications

Storage Layer

Anchoring

Within-Component Layer

presentation, dynamics, user interaction

database of nodes and links

node content, structure within nodes

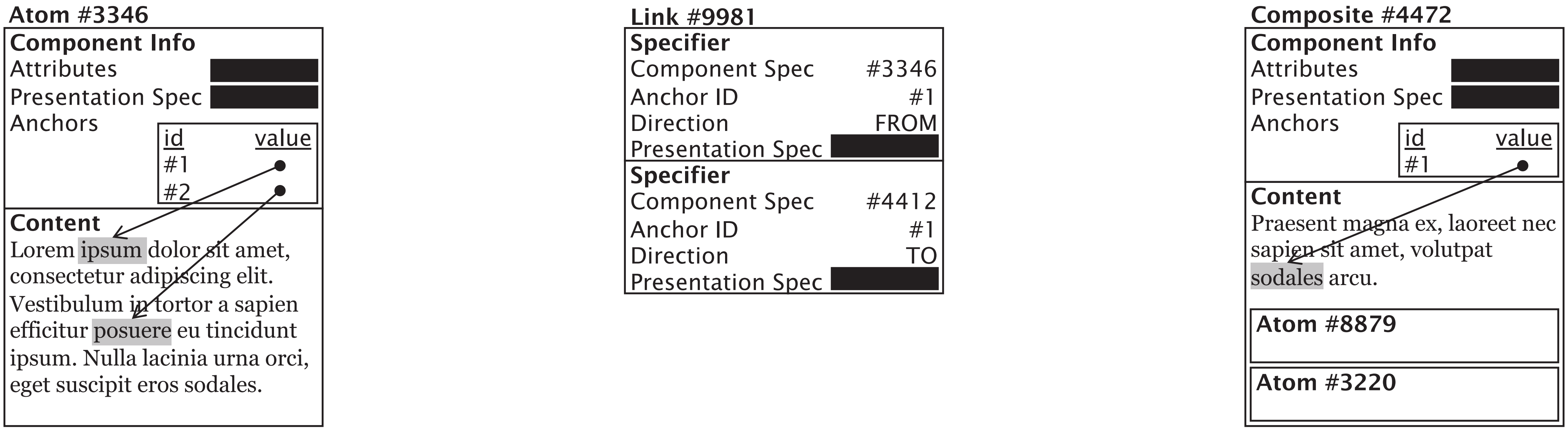
Halasz, F. and Schwartz, M. (1994) *The Dexter hypertext reference model*, Communications of the ACM, 37(2), pp. 30-39.

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
Anchors

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

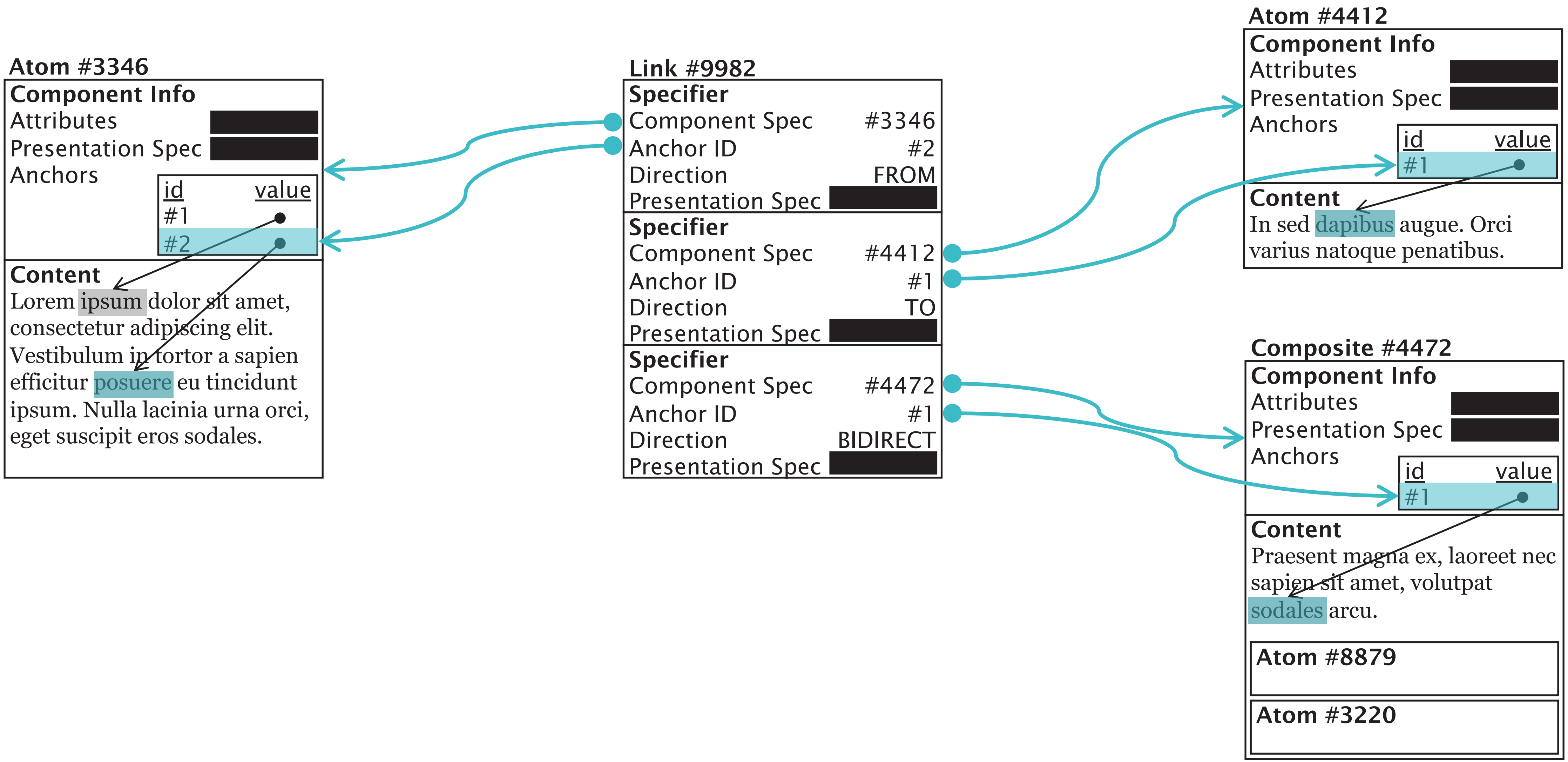
Component Info
Attributes
Presentation Spec
Anchors

id	value
#1	●

Content
Praesent magna ex, laoreet nec sapien sit amet, volutpat sodales arcu.

Atom #8879

Atom #3220



Atom #3346

Component Info
Attributes [redacted]
Presentation Spec [redacted]
Anchors

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 #9982

Specifier

Component Spec	#3346
Anchor ID	#2
Direction	FROM
Presentation Spec	[redacted]

Specifier

Component Spec	#4412
Anchor ID	#1
Direction	TO
Presentation Spec	[redacted]

Specifier

Component Spec	#4472
Anchor ID	#1
Direction	BIDIRECT
Presentation Spec	[redacted]

Atom #4412

Component Info
Attributes [redacted]
Presentation Spec [redacted]
Anchors

id	value
#1	●

Content
In sed dapibus augue. Orci varius natoque penatibus.

Composite #4472

Component Info
Attributes [redacted]
Presentation Spec [redacted]
Anchors

id	value
#1	●

Content
Praesent magna ex, laoreet nec sapien sit amet, volutpat sodales arcu.

Atom #8879

Atom #3220

Hyper-G

Hyper-G

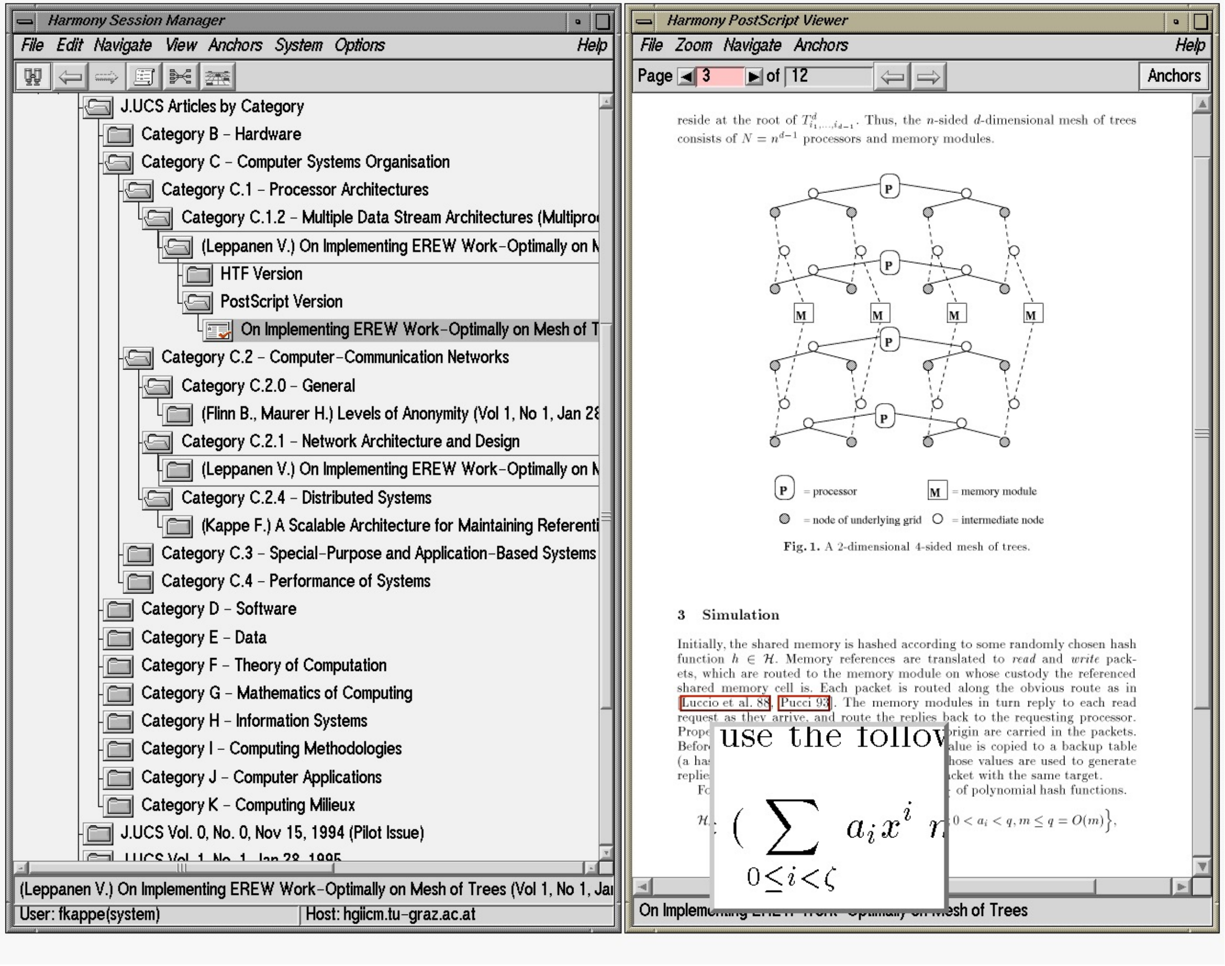
Developed at the Technical University of Graz, Austria around 1989-1990

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

Andrews, K., Kappe, F. and Maurer, H. (1995) *The Hyper-G Network Information System*, Journal of Universal Computer Science, 1(4).

Harmony client



- J.UCS 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 M
          - 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 M
      - Category C.2.4 – Distributed Systems
        - (Kappe F.) A Scalable Architecture for Maintaining Referenti
    - 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
- J.UCS Vol. 0, No. 0, Nov 15, 1994 (Pilot Issue)
- J.UCS Vol. 1, No. 1, Jan 28, 1995

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

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

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

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. Prope … origin are carried in the packets. Befor … alue is copied to a backup table (a has … hose values are used to generate replie … cket with the same target.

Fo … of polynomial hash functions.

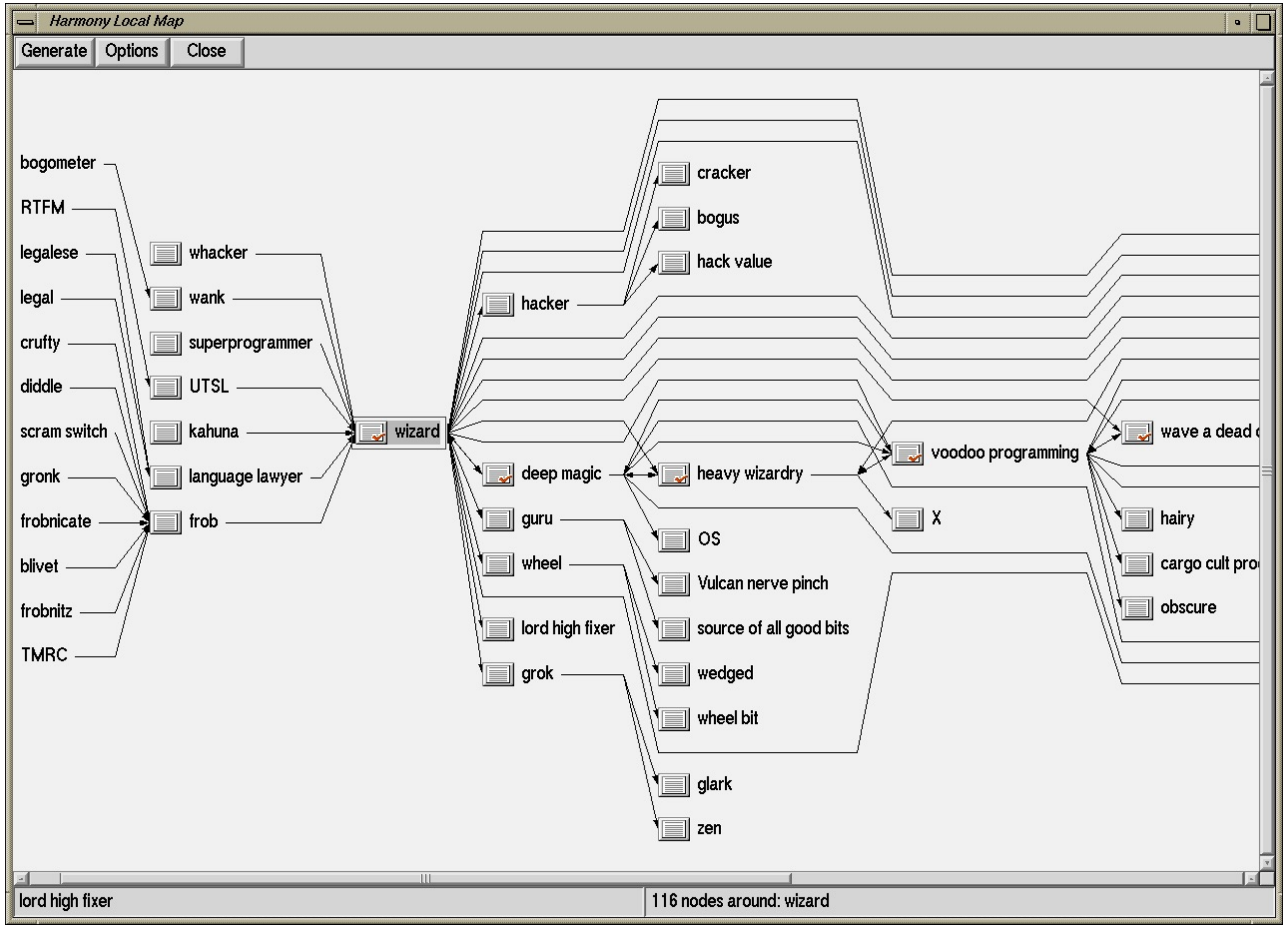
$\mathcal{H}$ … $0 < a_i < q, m \leq q = O(m)\big\}$,

use the follov

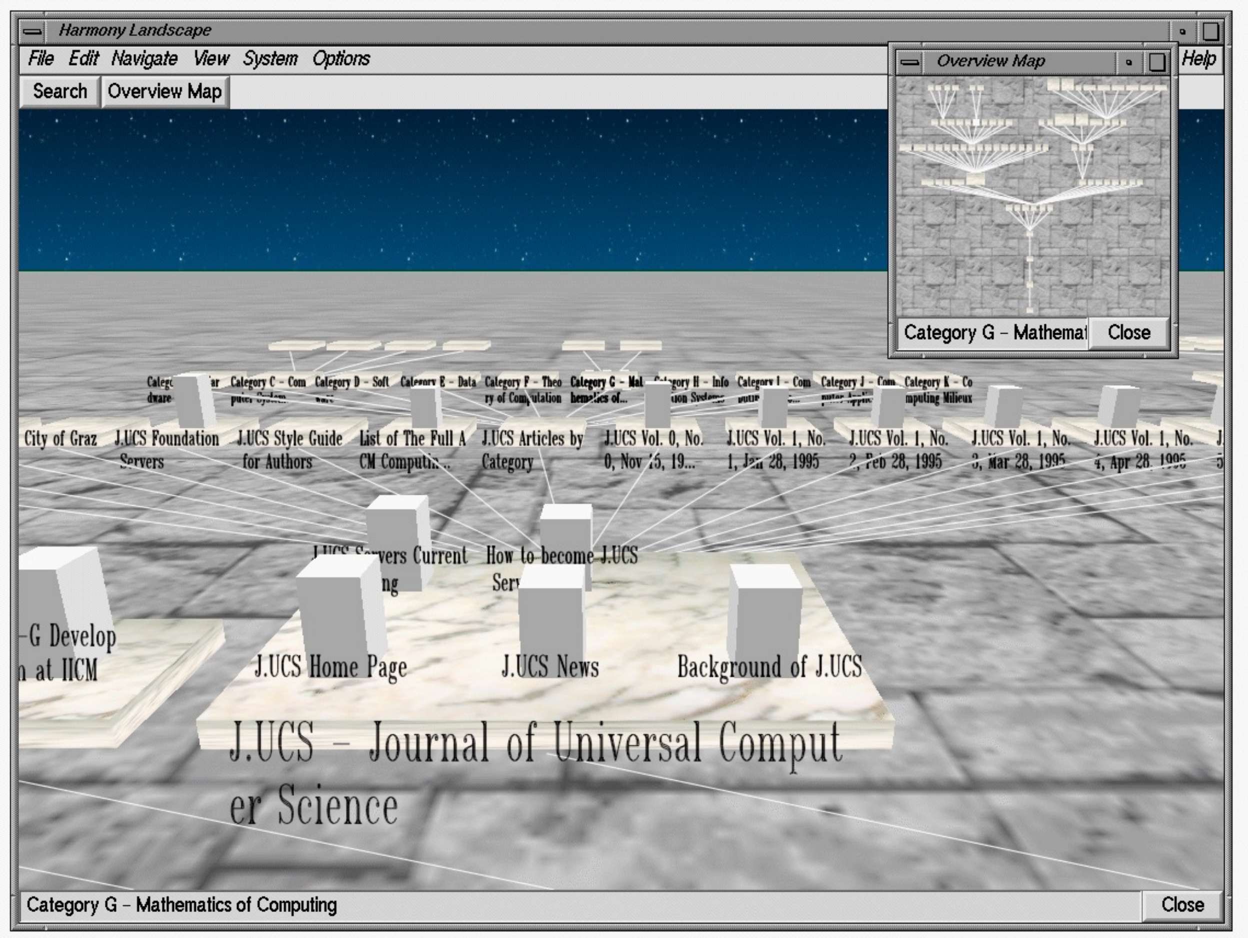
$$\Big(\sum_{0 \leq i < \zeta} a_i x^i$$

On Implementing EREW Work–Optimally on Mesh of Trees

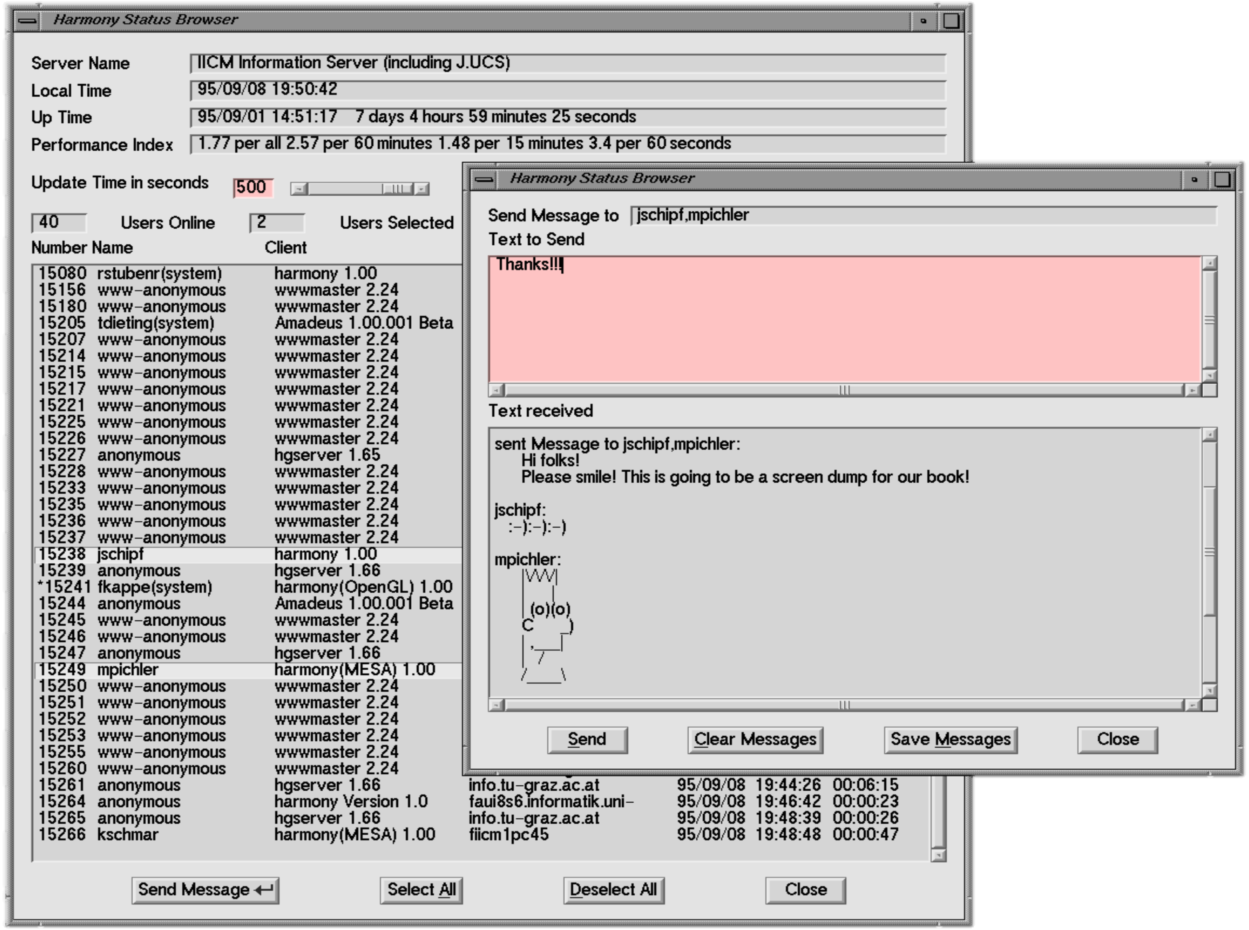
Harmony client: link structure view



Harmony client: landscape view


Harmony Landscape
File Edit Navigate View System Options
Help
Search
Overview Map
Overview Map
Category G – Mathemat
Close
Category C – Computer Systems
Category D – Software
Category E – Data
Category F – Theory of Computation
Category G – Mathematics of...
Category H – Information Systems
Category I – Com
Category J – Computer Applications
Category K – Computing Milieux
City of Graz
J.UCS Foundation Servers
J.UCS Style Guide for Authors
List of The Full ACM Computin...
J.UCS Articles by Category
J.UCS Vol. 0, No. 0, Nov 15, 19...
J.UCS Vol. 1, No. 1, Jan 28, 1995
J.UCS Vol. 1, No. 2, Feb 28, 1995
J.UCS Vol. 1, No. 3, Mar 28, 1995
J.UCS Vol. 1, No. 4, Apr 28, 1995
J.UCS Servers
Current
How to become J.UCS Serv
-G Develop
n at IICM
J.UCS Home Page
J.UCS News
Background of J.UCS
J.UCS – Journal of Universal Computer Science
Category G – Mathematics of Computing
Close

Harmony client: client-client communications



```
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

40   Users Online    2   Users Selected
Number Name                 Client
15080  rstubenr(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  jschipf              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          info.tu-graz.ac.at           95/09/08 19:44:26 00:06:15
15264  anonymous            harmony Version 1.0    faui8s6.informatik.uni-      95/09/08 19:46:42 00:00:23
15265  anonymous            hgserver 1.66          info.tu-graz.ac.at           95/09/08 19:48:39 00:00:26
15266  kschmar              harmony(MESA) 1.00     fiicm1pc45                   95/09/08 19:48:48 00:00:47
```

```
Send Message to   jschipf,mpichler
Text to Send
Thanks!!!

Text received
sent Message to jschipf,mpichler:
      Hi folks!
      Please smile! This is going to be a screen dump for our book!

jschipf:
   :-):-):-)

mpichler:
      |VVV|
      |    |
      | (o)(o)
      C      _)
      | ,___|
      |   /
      /____\
```

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

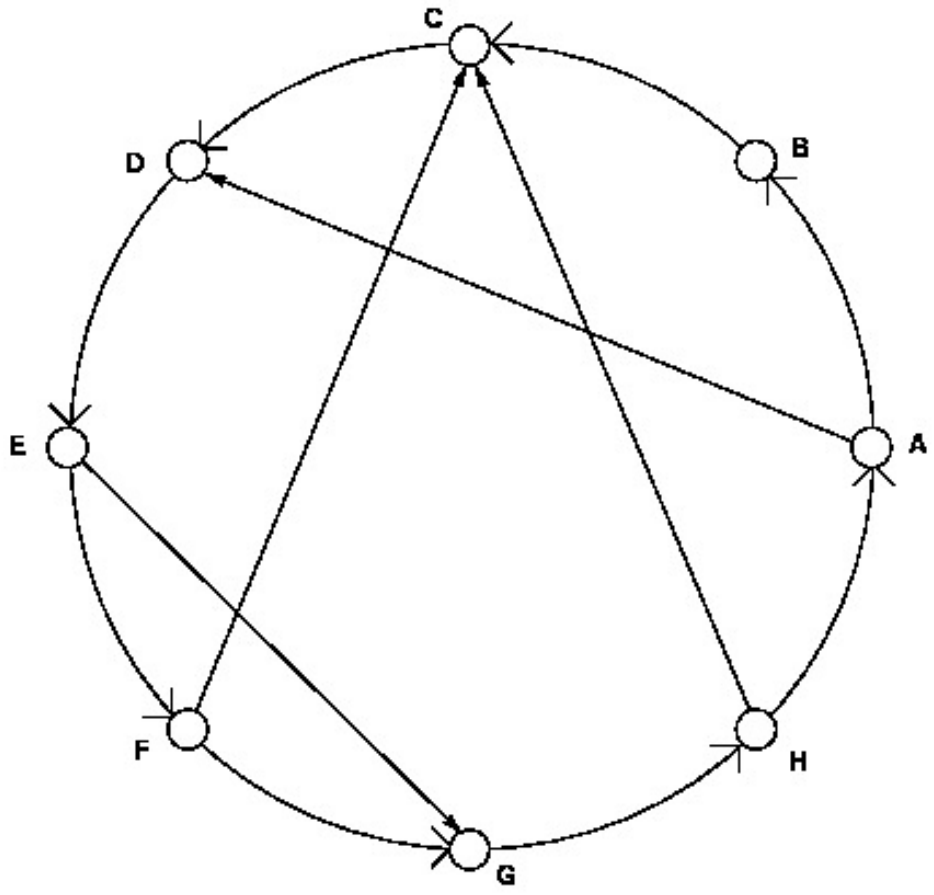
Changes that affect surface links involve multiple servers that need to be notified

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

- Own internet protocol (HG-CSP)
- Own markup language (HTF)
- Own browser (and other tools) (Harmony) - though simple browsing through a web browser too

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

Microcosm

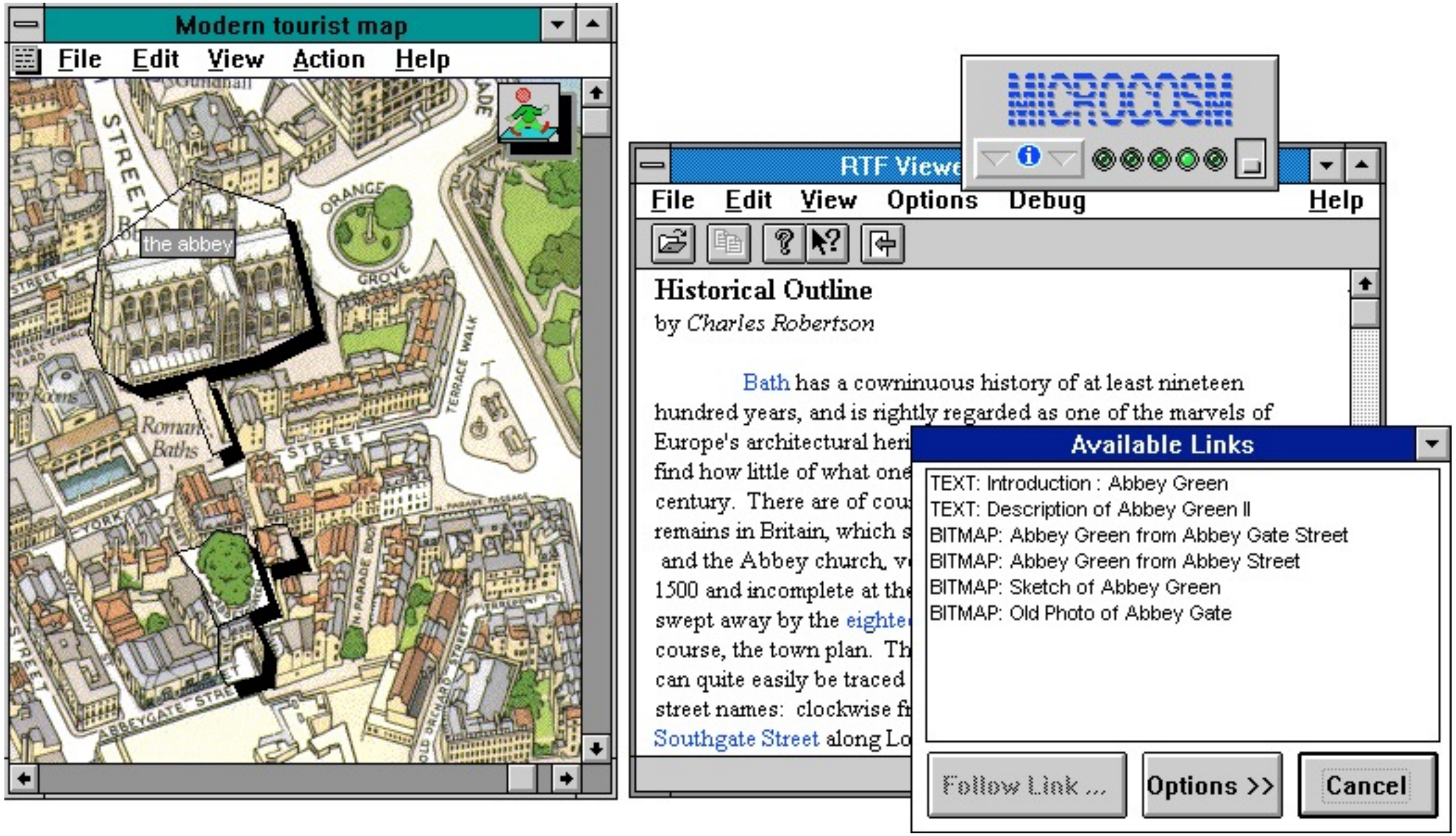
Overview

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
- Informed much subsequent work within ECS: DLS, COHSE, etc

Fountain, A., Hall, W., Heath, I. and Davis, H. (1990). MICROCOSM: an open model for hypermedia with dynamic linking. Proceedings of ECHT90. pp. 298-311.

Microcosm client

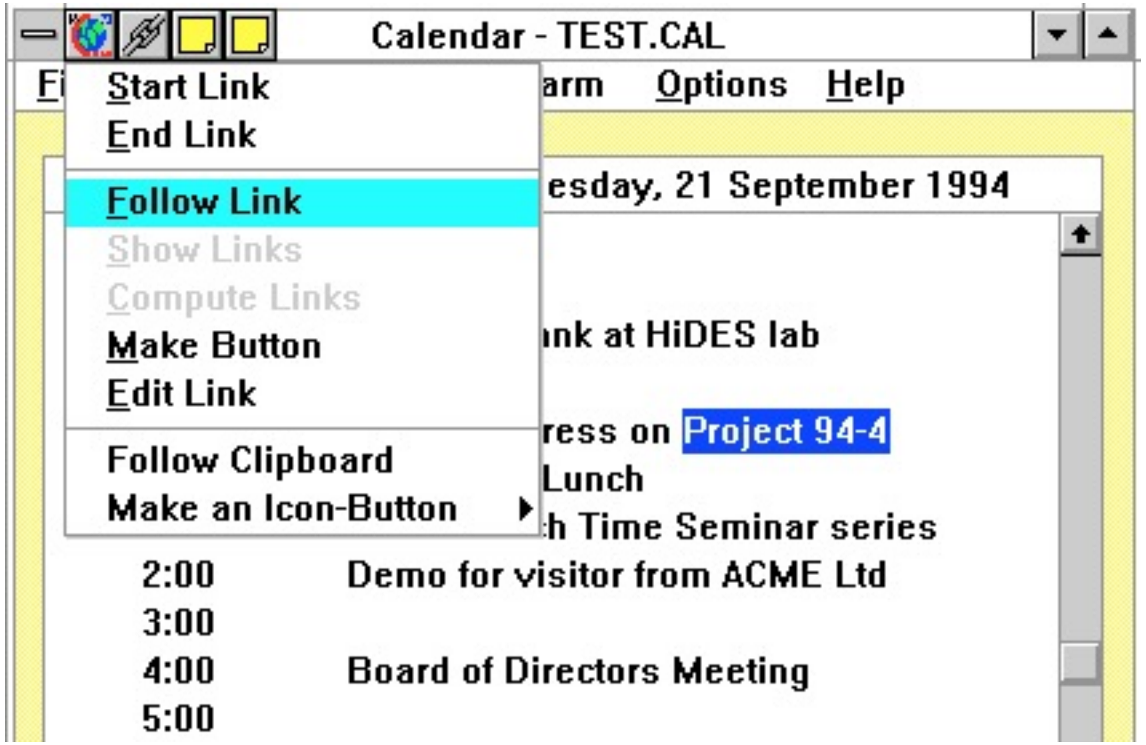


Historical Outline
by *Charles Robertson*

Bath has a cowninuous history of at least nineteen
hundred years, and is rightly regarded as one of the marvels of
Europe's architectural heri
find how little of what one
century. There are of cou
remains in Britain, which s
and the Abbey church, v
1500 and incomplete at the
swept away by the eighte
course, the town plan. Th
can quite easily be traced
street names: clockwise fr
Southgate Street along Lo

TEXT: Introduction : Abbey Green
TEXT: Description of Abbey Green II
BITMAP: Abbey Green from Abbey Gate Street
BITMAP: Abbey Green from Abbey Street
BITMAP: Sketch of Abbey Green
BITMAP: Old Photo of Abbey Gate

Microcosm universal viewer



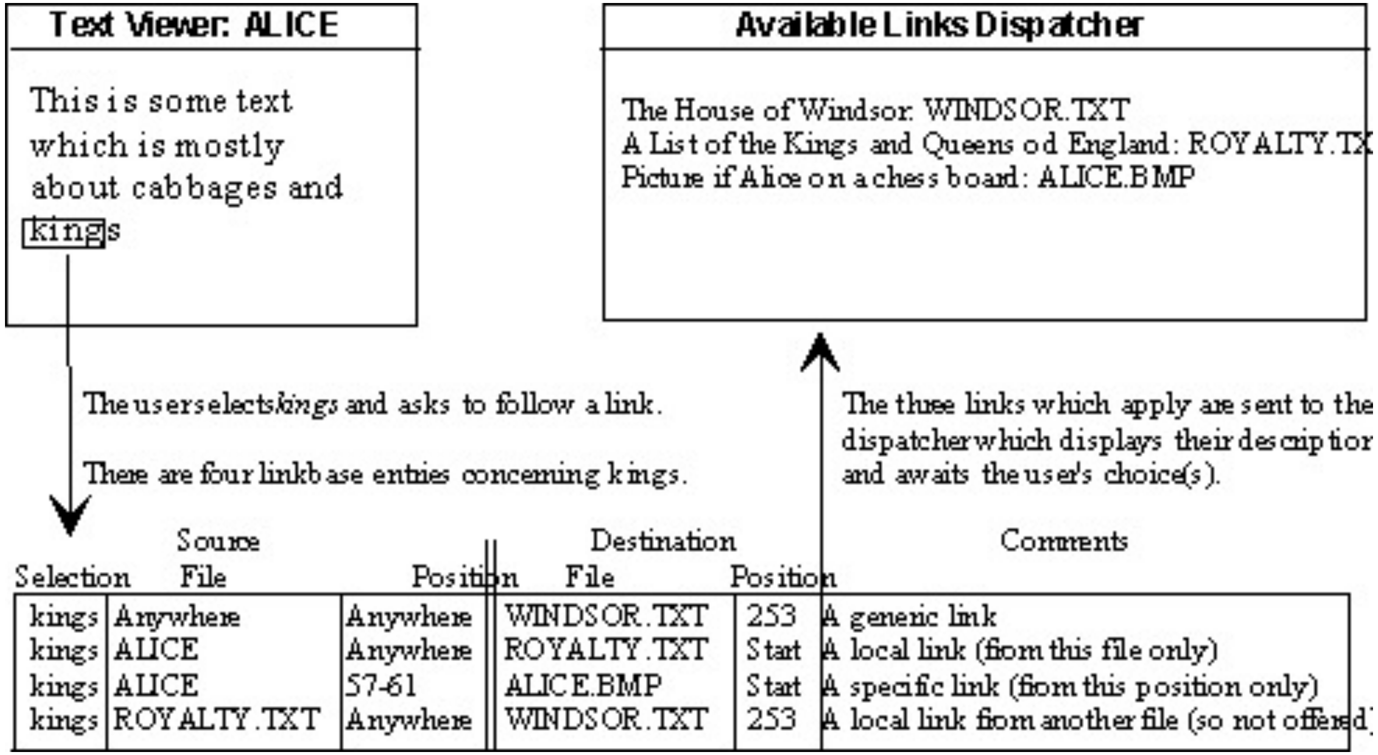
```
esday, 21 September 1994

nk at HiDES lab

ress on Project 94-4
Lunch
h Time Seminar series
2:00        Demo for visitor from ACME Ltd
3:00
4:00        Board of Directors Meeting
5:00
```

Specific, local and generic links

Position of source anchor held in the linkbase



Text Viewer: ALICE

This is some text which is mostly about cabbages and kings

Available Links Dispatcher

The House of Windsor: WINDSOR.TXT
A List of the Kings and Queens od England: ROYALTY.TX
Picture if Alice on a chess board: ALICE.BMP

The user selects *kings* and asks to follow a link.

There are four linkbase entries concerning kings.

The three links which apply are sent to the dispatcher which displays their descriptio and awaits the user's choice(s).

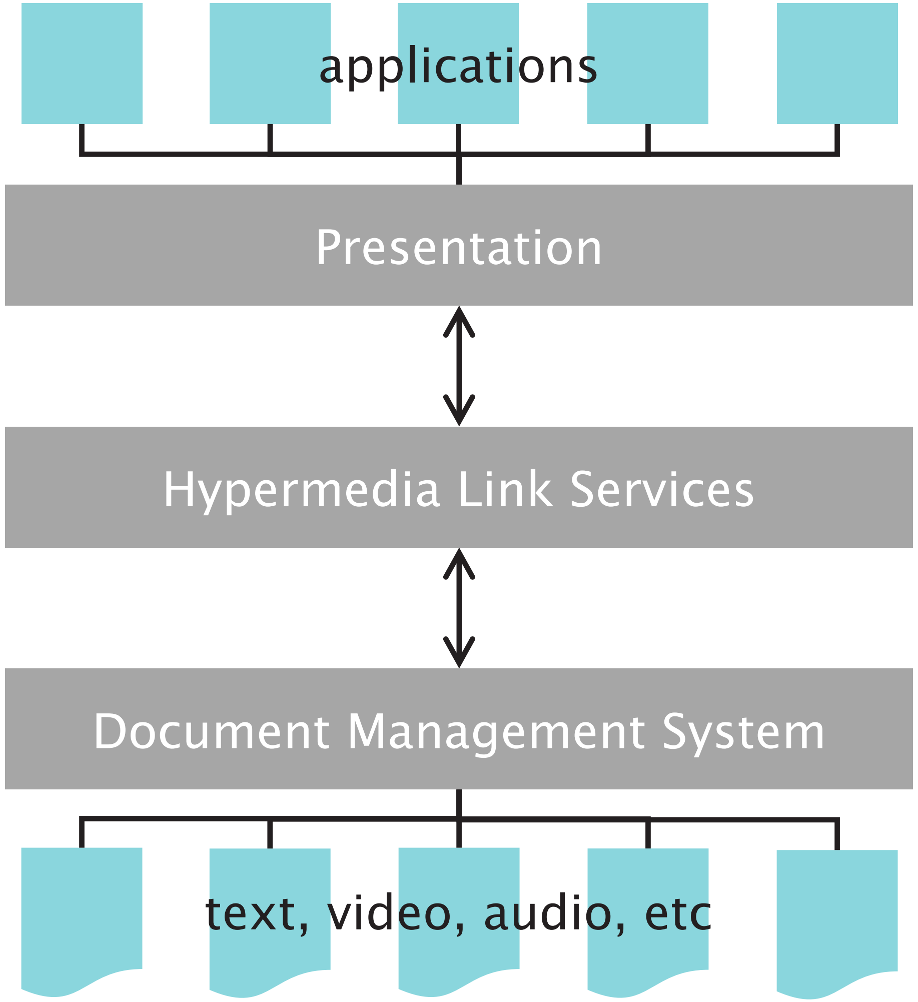
	Source		Destination		Comments
Selection	File	Position	File	Position	
kings	Anywhere	Anywhere	WINDSOR.TXT	253	A generic link
kings	ALICE	Anywhere	ROYALTY.TXT	Start	A local link (from this file only)
kings	ALICE	57-61	ALICE.BMP	Start	A specific link (from this position only)
kings	ROYALTY.TXT	Anywhere	WINDSOR.TXT	253	A local link from another file (so not offered

Microcosm architecture

Separation of concerns:

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

Link services mediate the other components

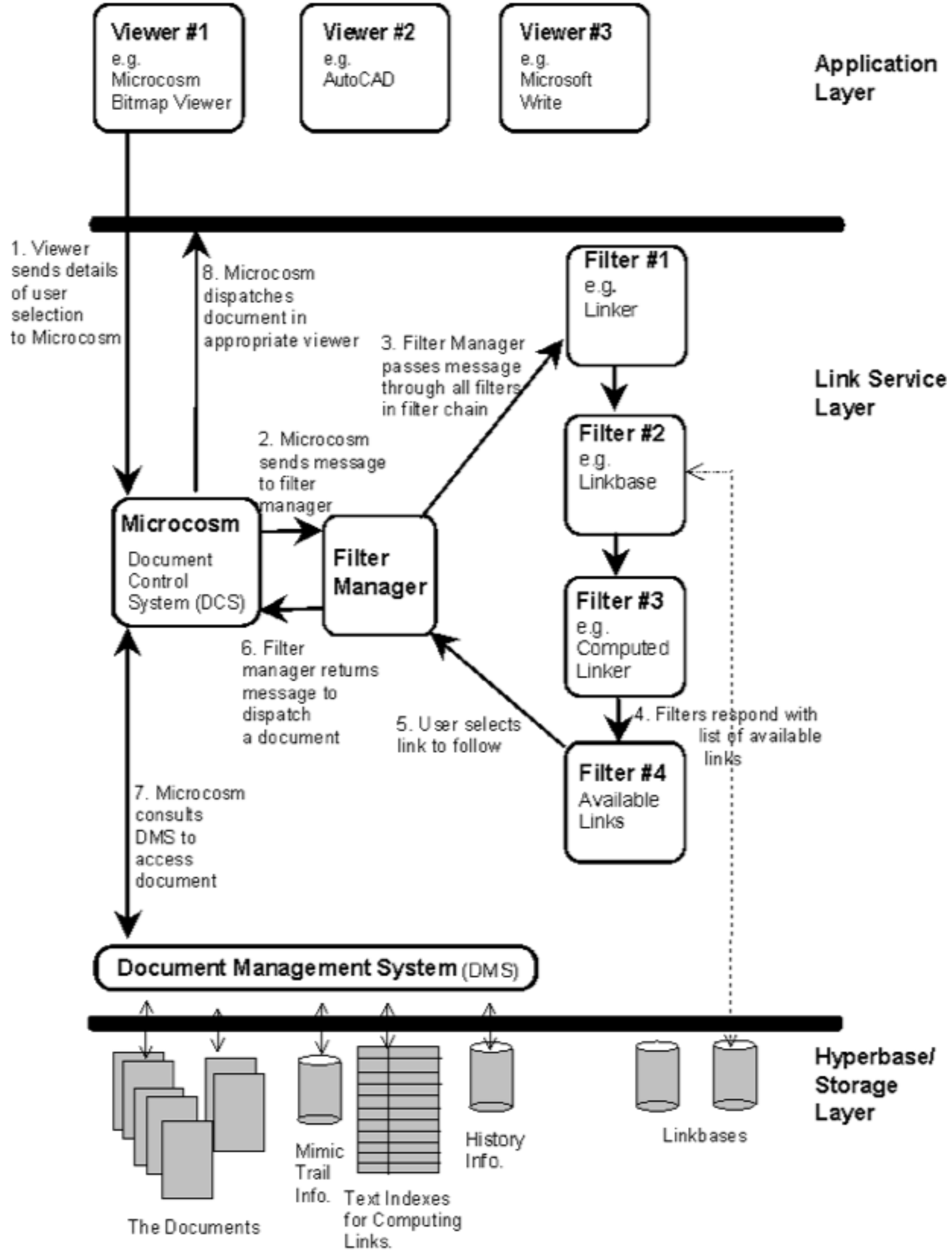


Microcosm architecture

Link service comprises a sequence of *filters*

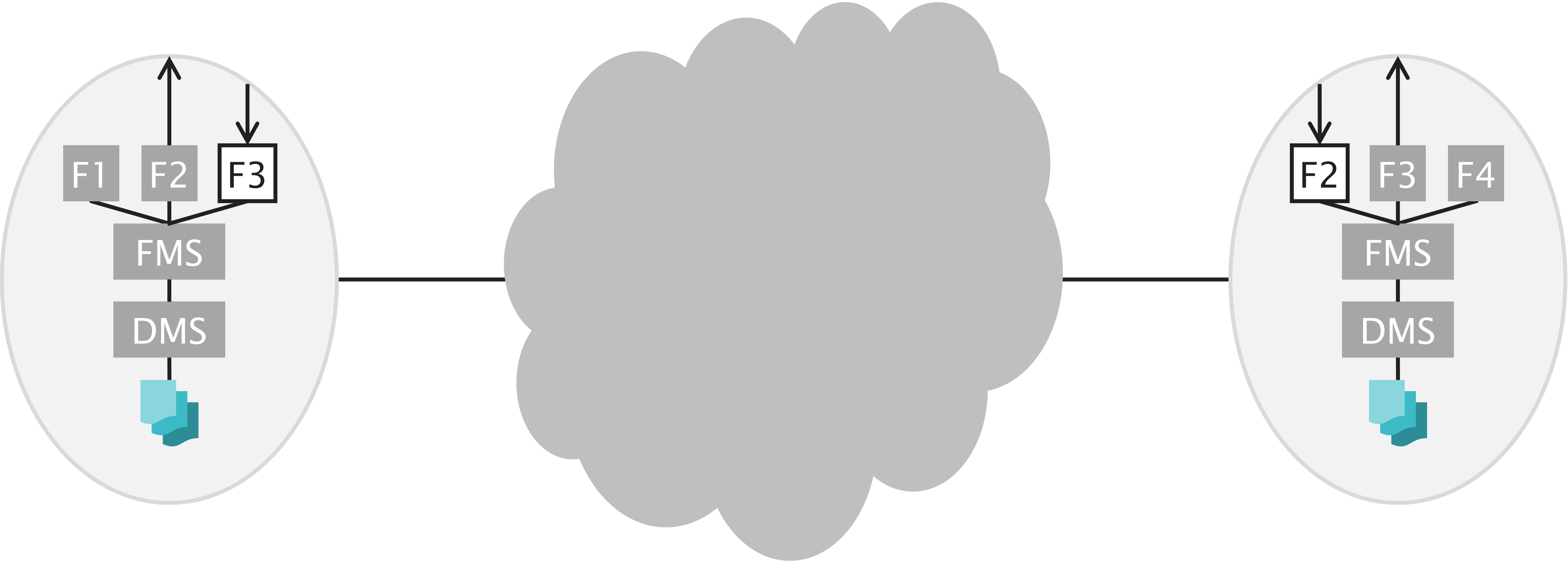
Filters are composable services that generate or manipulate links

Filter manager organises filters into chains


Viewer #1
e.g. Microcosm Bitmap Viewer
Viewer #2
e.g. AutoCAD
Viewer #3
e.g. Microsoft Write
Application Layer
1. Viewer sends details of user selection to Microcosm
8. Microcosm dispatches document in appropriate viewer
3. Filter Manager passes message through all filters in filter chain
Filter #1
e.g. Linker
Link Service Layer
2. Microcosm sends message to filter manager
Filter #2
e.g. Linkbase
Microcosm
Document Control System (DCS)
Filter Manager
Filter #3
e.g. Computed Linker
6. Filter manager returns message to dispatch a document
5. User selects link to follow
4. Filters respond with list of available links
Filter #4
Available Links
7. Microcosm consults DMS to access document
Document Management System (DMS)
Hyperbase/ Storage Layer
The Documents
Mimic Trail Info.
Text Indexes for Computing Links.
History Info.
Linkbases

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

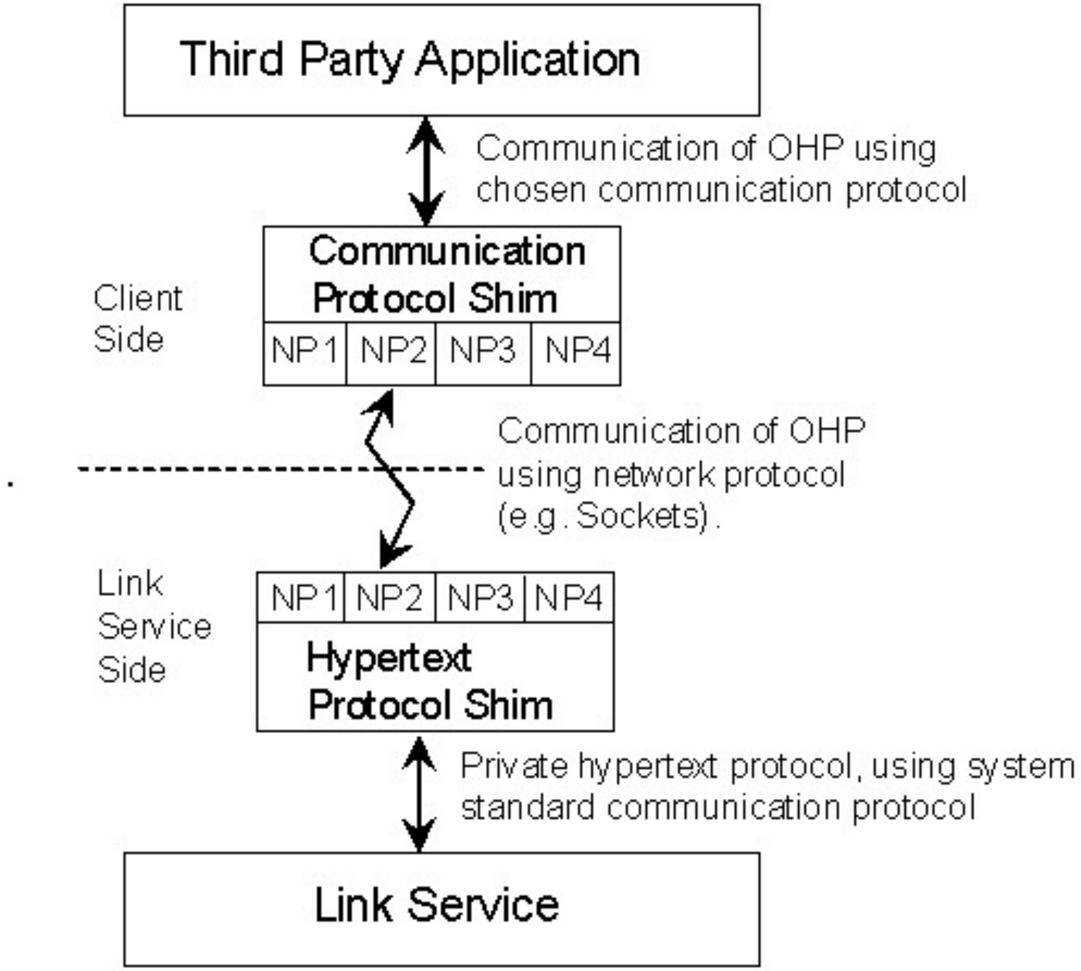
Initially a naïve attempt to “shim” existing linkservers so that they could be used by standard client integrations

- An early realization by the community that writing clients is far harder (and less interesting) than writing servers

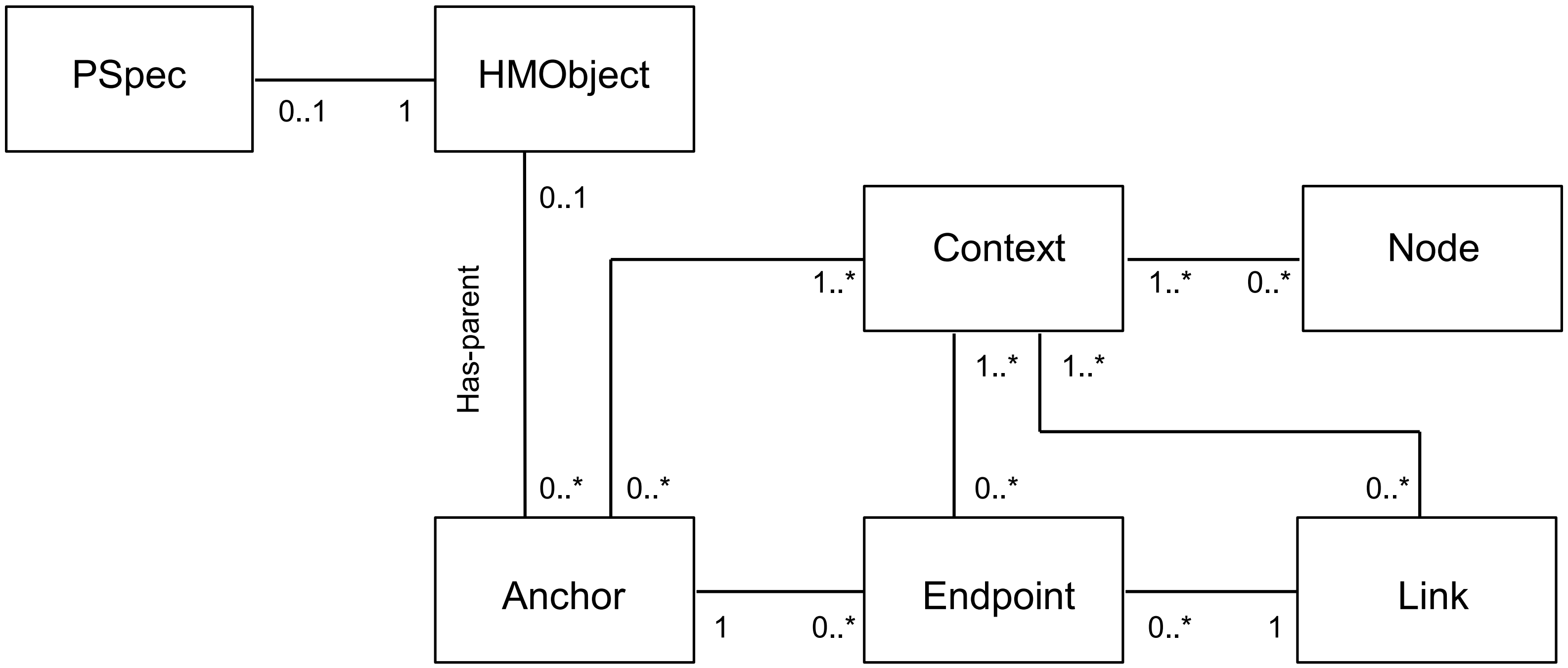
OHP formed the basis for later integration efforts (FOHM)

Reich, S., Will, U.K., Nürnberg, P., Davis, H.C., Grønbæk, K, Millard, D.E. and Haake, J.M. (1999) *Addressing interoperability in open hypermedia: the design of the Open Hypermedia Protocol*. New Review of Hypermedia and Multimedia 5(1), pp. 207-248.

OHP architecture


Third Party Application
Communication of OHP using chosen communication protocol
Communication Protocol Shim
NP1
NP2
NP3
NP4
Client Side
Communication of OHP using network protocol (e.g. Sockets).
Link Service Side
NP1
NP2
NP3
NP4
Hypertext Protocol Shim
Private hypertext protocol, using system standard communication protocol
Link Service

OHP data model



Location specifiers (LocSpec)

Used to identify the position of an anchor within an object

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

Exists within links (as part of the endpoint) and within nodes

Generalisation of Dexter's anchor values

- Mechanism for identifying locations in other formats
- Function to evaluate to identify anchor position

Grønbæk, K. and Trigg, R. (1996) *Toward a Dexter-based model for open hypermedia: unifying embedded references and link objects*. In HYPERTEXT '96: Proceedings of the the seventh ACM conference on Hypertext, pp.149-160.

Evaluation

Advantages

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

Disadvantages

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

Anderson, K.M. et al (1997) *A critique of the Open Hypermedia Protocol*. Journal of Inline Digital Information, 1(2). | Available at: https://journals.tdl.org/jodi/index.php/jodi/article/view/5

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

Davis, H.D. (1998) *Referential integrity of links in open hypermedia systems*. Proceedings of Hypertext 98. pp. 207-216.

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)

Responsible link owner or responsible 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

UNIVERSITY OF Southampton

Next Lecture: Open Hypermedia on the Web