Phlog: Blogging via Gopher

Neuer Wein aus uralten Schläuchen

Jens Ohlig <jens@ccc.de>

GPN4, Karlsruhe

Gopher?

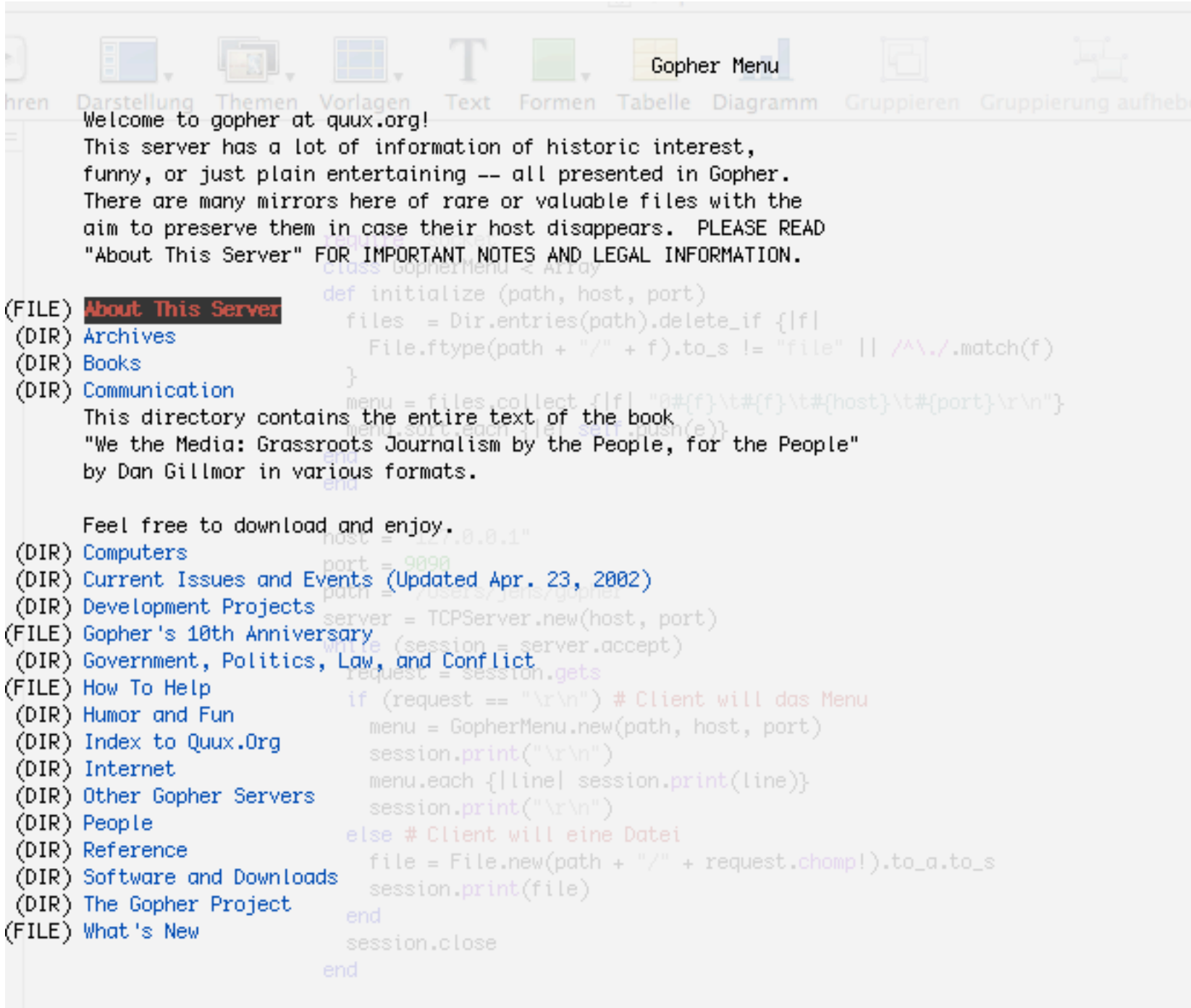
Gopher

- ein nordamerikanisches Nagetier
- das Maskottchen der University of Minnesota
- RFC 1436
- ein verteiltes Dokumentensystem für das Internet

Kampf der Giganten

- 1994: 4800 Gopher-Server, 1200 FTP-Server, 600 WWW-Server
- dann kam NCSA Mosaic
- Probleme: fehlende Erweiterbarkeit
- komische Lizenzen
- zu wenig Boulevard und Bilder

Wie sah das aus?



RFC 1436

- Designed for simplicity
- Grundsätzlich: Es gibt Menüs und Dateien
- Client sendet entweder nichts (bzw. “\r\n”) oder einen sogenannten Selector

Client-Server-Kommunikation

```
Client:          {Opens connection to rawBits.micro.umn.edu at port 70}

Server:          {Accepts connection but says nothing}

Client: <CR><LF> {Sends an empty line: Meaning "list what you have"}

Server:          {Sends a series of lines, each ending with CR LF}
0About internet GopherFStuff:About usFrawBits.micro.umn.eduF70
1Around University of MinnesotaFZ,5692,AUMFunderdog.micro.umn.eduF70
1Microcomputer News & PricesFPrices/Fpserver.bookstore.umn.eduF70
1Courses, Schedules, CalendarsFFevents.ais.umn.eduF9120
1Student-Staff DirectoriesFFuinfo.ais.umn.eduF70
1Departmental PublicationsFStuff:DP:FrawBits.micro.umn.eduF70
                  {.....etc.....}
.                 {Period on a line by itself}
                  {Server closes connection}
```

Gopher Item Types

```
0   Item is a file
1   Item is a directory
2   Item is a CSO phone-book server
3   Error
4   Item is a BinHexed Macintosh file.
5   Item is DOS binary archive of some sort.
    Client must read until the TCP connection closes.  Beware.
6   Item is a UNIX uuencoded file.
7   Item is an Index-Search server.
8   Item points to a text-based telnet session.
9   Item is a binary file
    Client must read until the TCP connection closes.  Beware.
+   Item is a redundant server
T   Item points to a text-based tn3270 session.
g   Item is a GIF format graphics file.
I   Item is some kind of image file.  Client decides how to display.
```

Experimentelle Types

- i: Inline-Text
- h: HTML
- M: MIME-Dateien

Was gab es noch?

- Telefonbücher!
- WAIS
- VERONICA

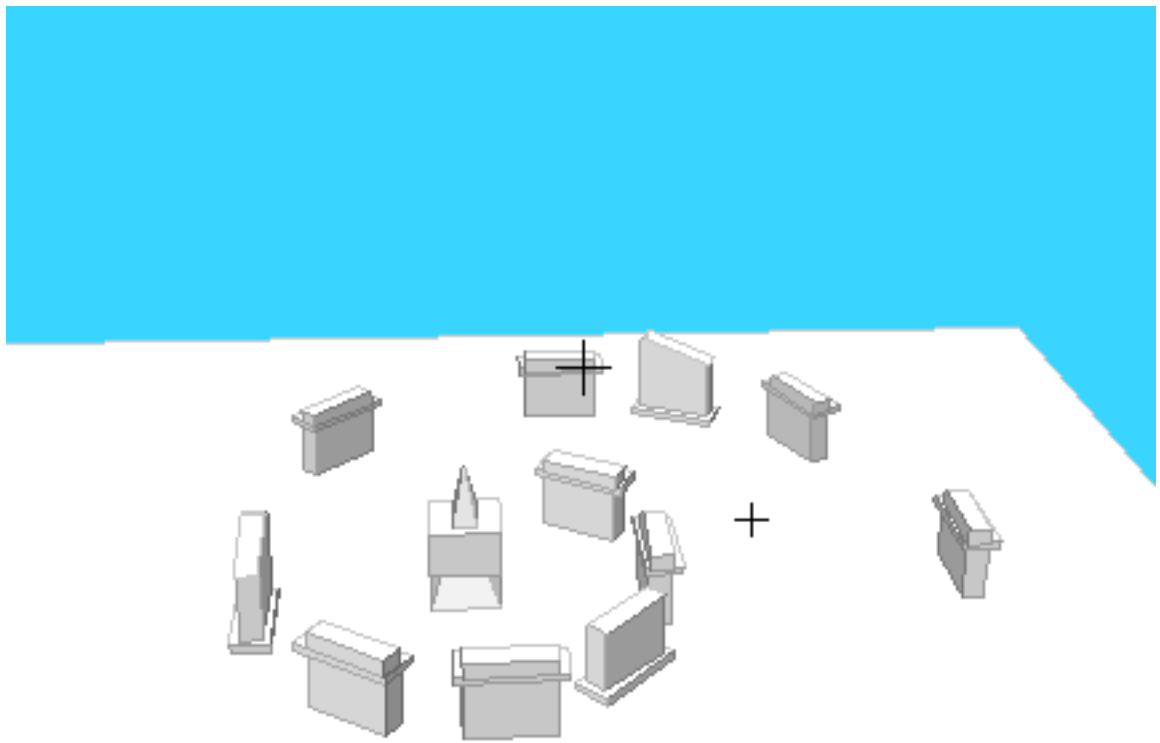
Welche Gopher-Server gibt es heute?

- gopher (das Original)
- GN (Gopher/WWW)
- Bucktooth (Perl)
- pygopherd (Python)

Gopher-Clients

- gopher (Original)
- Lynx
- Mozilla/Firefox (mit starken Abstrichen)
- IE (eigentlich gar nicht)

Irre Ideen damals: GopherVR



Irre Ideen damals: GopherMOO

You talk to Larry about some favorite places, and he points the slate at it.

```
Larry says, "everytime someone moves the gopher pointer, the slate's description changes"
Larry goes back up a level on Gopher Slate.
Larry picks '2. Electronic Frontier Foundation archives (menu)' on Gopher Slate.
Larry says, "so now, look at the slate again"

>look at slate

Gopher Slate: gopher.eff.org Electronic Frontier Foundation archives
1. About the Electronic Frontier Foundation's Gopher Service
2. About the Electronic Frontier Foundation
3. Electronic Frontier Foundation Files & Information (menu)
4. Computers & Academic Freedom mailing list archives & info (menu)
5. EFF-Austin Gopher Service (menu)
6. Other Gopher and Information Servers (menu)
Larry says, "if you want to choose an item, just say 'pick  on slate'"

>pick 2 on slate
```

Bastelprojekt: Mini-Gopher

- Gerade genug RFC 1436, um ein Verzeichnis zu serven

```ruby
require 'socket'
class GopherMenu < Array
def initialize (path, host, port)
  files  = Dir.entries(path).delete_if {|f|
    File.ftype(path + "/" + f).to_s != "file" || /^\./.match(f)
  }
  menu = files.collect {|f| "0#{f}\t#{f}\t#{host}\t#{port}\r\n"}
  menu.sort.each {|e| self.push(e)}
end
end

host = "127.0.0.1"
port = 9090
path = "/Users/jens/gopher"
server = TCPServer.new(host, port)
while (session = server.accept)
  request = session.gets
  if (request == "\r\n") # Client will das Menu
    menu = GopherMenu.new(path, host, port)
    session.print("\r\n")
    menu.each {|line| session.print(line)}
    session.print("\r\n")
  else # Client will eine Datei
    file = File.new(path + "/" + request.chomp!).to_a.to_s
    session.print(file)
  end
  session.close
end
```

Und weiter?

- XML-RPC via Gopher
- XPath via Gopher
- Tag-Browsing
- RSS
- Dokumentenbrowser

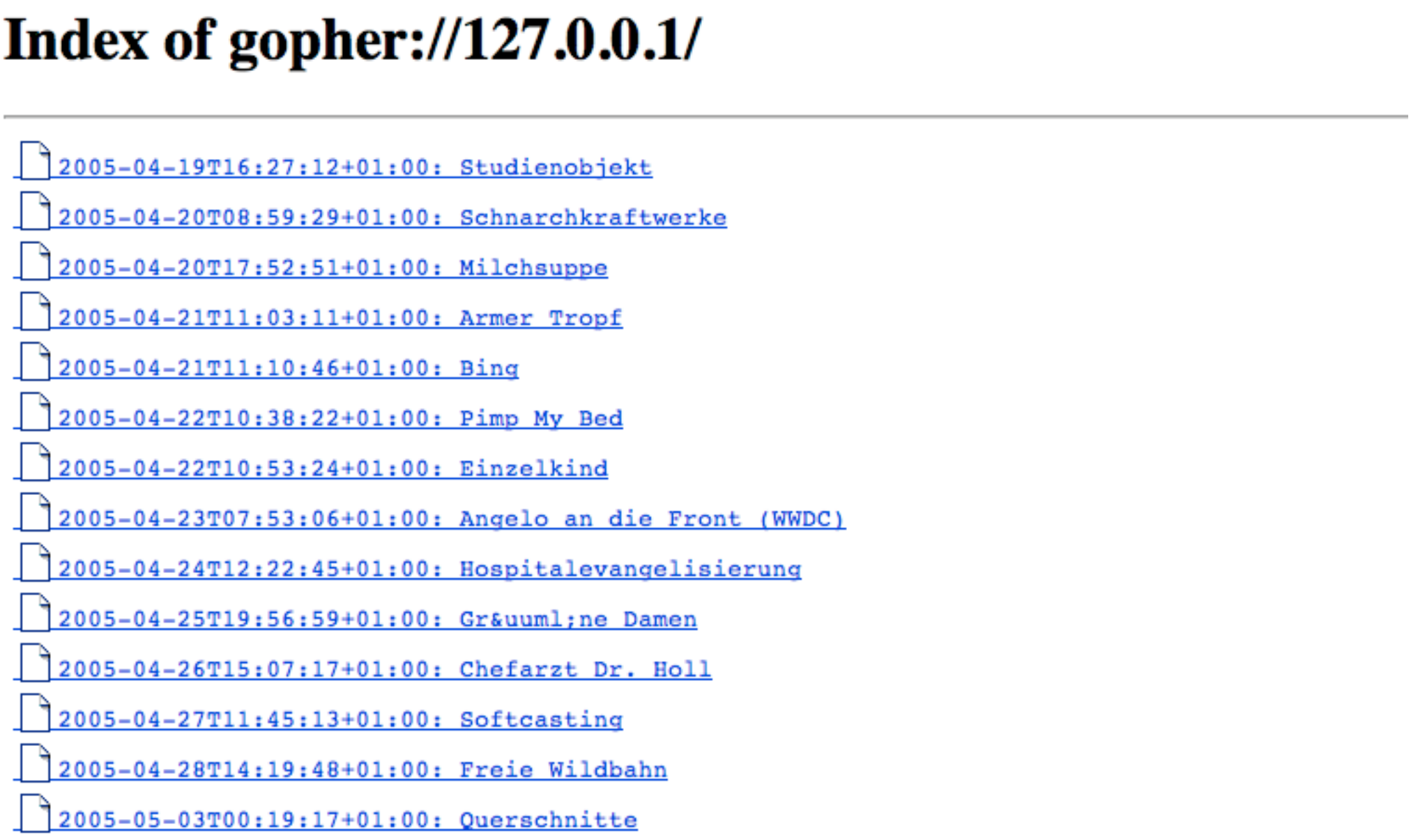
A match made in Heaven

- Gopher ist dann ideal, wenn es hierarchisierte Datenstrukturen darzustellen gilt
- XML ist eine Darstellungsform hierarchisierter Datenstrukturen
- ...
- Profit!

Mein eigenes Phlog: Gopher-Verzeichnis aus RSS

```
require "rexml/document"
include REXML
path = "/Users/jens/gopher"
doc = Document.new File.new("index.rdf")
dates = Array.new
titles = Array.new
content = Array.new
doc.elements.each("rdf:RDF/item/dc:date") { |e| dates.push(e.text) }
doc.elements.each("rdf:RDF/item/title") { |e| titles.push(e.text) }
doc.elements.each("rdf:RDF/item/content:encoded") { |e| content.push(e.text) }
dates.each_index {|i|
    filename =  "#{path}/#{dates[i]}: #{titles[i]}"
    f = File.new(filename,  "w+")
    f.puts(content[i])
    f.close
}
```

Et voilà!


Index of gopher://127.0.0.1/
2005-04-19T16:27:12+01:00: Studienobjekt
2005-04-20T08:59:29+01:00: Schnarchkraftwerke
2005-04-20T17:52:51+01:00: Milchsuppe
2005-04-21T11:03:11+01:00: Armer Tropf
2005-04-21T11:10:46+01:00: Bing
2005-04-22T10:38:22+01:00: Pimp My Bed
2005-04-22T10:53:24+01:00: Einzelkind
2005-04-23T07:53:06+01:00: Angelo an die Front (WWDC)
2005-04-24T12:22:45+01:00: Hospitalevangelisierung
2005-04-25T19:56:59+01:00: Grüne Damen
2005-04-26T15:07:17+01:00: Chefarzt Dr. Holl
2005-04-27T11:45:13+01:00: Softcasting
2005-04-28T14:19:48+01:00: Freie Wildbahn
2005-05-03T00:19:17+01:00: Querschnitte