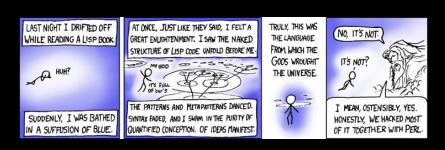
### Literature Management with Perl and Emacs

#### Stefan Washietl

EMBL-European Bioinformatics Institute, Hinxton/Cambridge, UK



## Part 1. A PubMed Interface for Emacs

Part 2. Automatic downloading of PDFs

## A PubMed Interface for Emacs

Part 1.

#### **About Emacs**

# EMACS The Extensible, Customizable Self-Documenting Display Editor

bу

#### Richard M. Stallman

Abstract: EMACS is a display editor which is implemented in an interpreted high level language. This allows users to extend the editor by replacing parts of it, to experiment with alternative command languages, and to share extensions which are generally useful. The ease of extension has contributed to the growth of a large set of useful features. This paper describes the organization of the EMACS system, emphasizing the way in which extensibility is achieved and used.

Keywords: Display, Editor, Extensible, Interactive, Self-documenting

#### Motivation

 Apparently no direct way to get PubMed references as BibTeX into Emacs.

	Min. no. of clicks	No. of applications
Shell script	5	3
Hubmed	5	2
JabRef	6	2
PubMode	0	1

#### Motivation

 Apparently no direct way to get PubMed references as BibTeX into Emacs.

	Min. no. of clicks	No. of applications
Shell script	5	3
Hubmed	5	2
JabRef	6	2
PubMode	0	1

A few clicks can change user-experience and individual workflows.

#### **PubMode**

- PubMode does not impose any particular workflow. It works with ever workflow you like and plays well with BibTeX mode, AucTeX, RefTeX.
- ▶ 100% Emacs Lisp without any dependencies
- Can be installed with one line in .emacs

```
(add-to-list 'load-path " /.emacs.d/pubmode/lisp")
```

#### **PubMode**

- Implements the whole public API provided by NCBI including
  - LinkOut
  - Related articles
  - Spell correction
- Generates robust and customizable BibTeX
- Does not neglect corner cases. Tries also to get all your

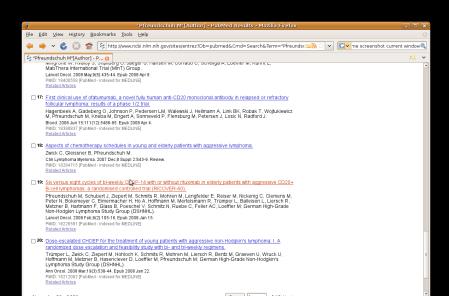
```
"von Sørensen jr." or "Bompfünewerer Consortium" right.
```

► Tested and ready for prime-time.

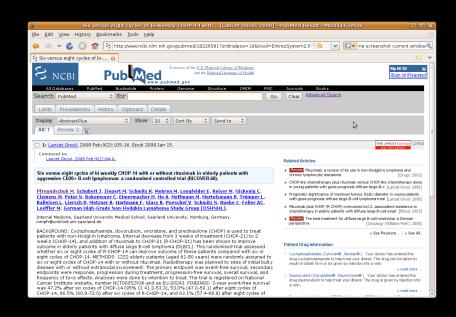
## Part 2.

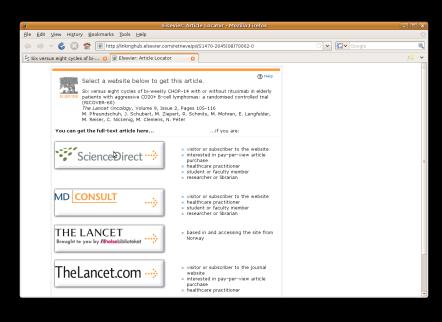
Automatic downloading of PDFs

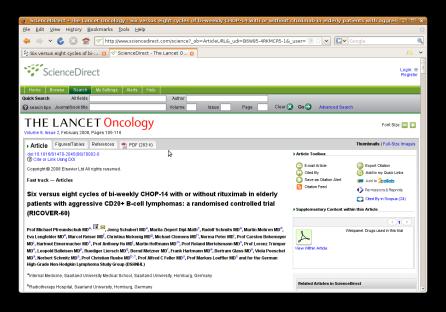
i ait 2.

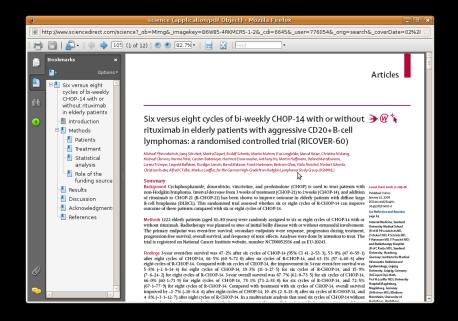












#### Click #5 (3 more to actually save it)

#### Difficulties for automatic PDF retrieval

- No direct PDF link considered in LinkOut specification
- No other efforts by publishers to provide a standardized interface to PDFs
- Use of meta-tags not widely adopted; Most publishers ignore them, some use them incorrectly.
- Many publishers present PDFs without any care for web-standards, semantics and accessibility.

## Idiotically linked PDFs: a business



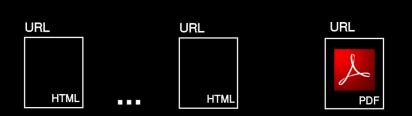


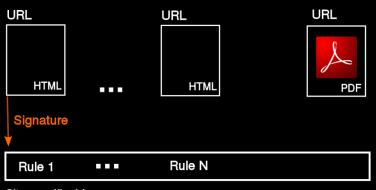
#### An open source solution

- Fully automatic crawler not feasible:
  - Would need to crawl up to three levels which is too slow
  - Publisher sites don't like crawlers
- We propose a rule-based semi-automatic agent to find PDFs

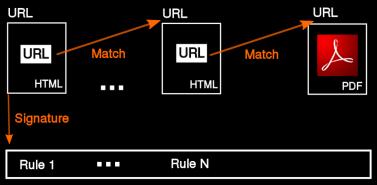




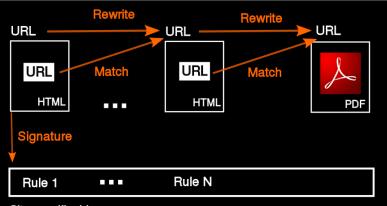




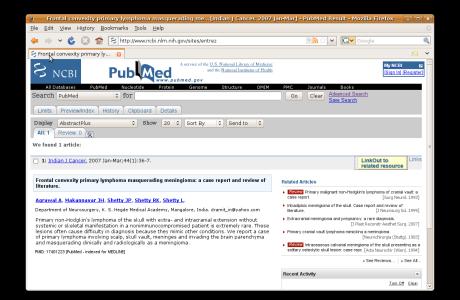
Site specific driver

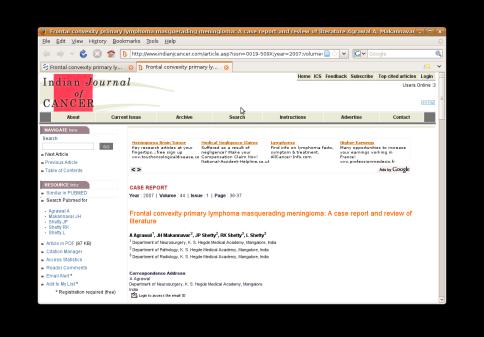


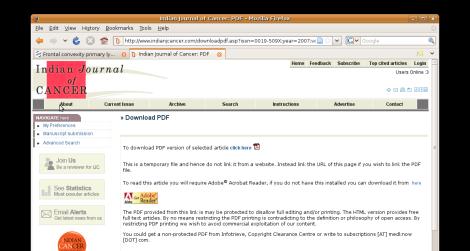
Site specific driver

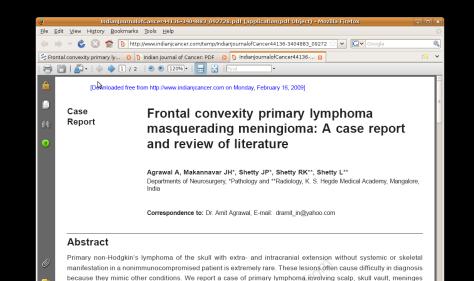


Site specific driver









and invading the brain parenchyma and masquerading clinically and radiologically as a meningioma.

```
<site name="Medknown">
    <signature>
        <url>!article.asp\?issn=\S*?aulast=\S*!</url>
        </signature>
```

```
<site name="Medknown">
  <signature>
    <url>!article.asp\?issn=\S*?aulast=\S*!</url>
  </signature>
  <rule>
    <pattern>
      <rewrite>!(.*)!$1;type=2!</rewrite>
    </pattern>
    <pattern>
      <match>!(temp/\S*?\.pdf)&quot;!</match>
    </pattern>
  </rule>
```

```
<site name="Medknown">
  <signature>
    <url>!article.asp\?issn=\S*?aulast=\S*!</url>
  </signature>
  <rule>
    <pattern>
      <rewrite>!(.*)!$1;type=2!</rewrite>
    </pattern>
    <pattern>
      <match>!(temp/\S*?\.pdf)&quot;!</match>
    </pattern>
  </rule>
  <test>
    <url>http://www.indianjcancer.com/article.asp?issn=...</url>
  </test>
</site>
```

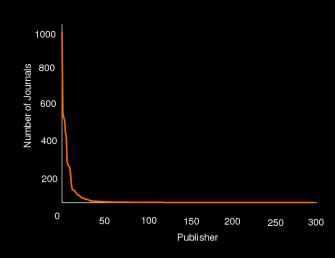
#### Driver in YAML format

```
Medknown:
  signature:
    url:
      - '!article.asp\?issn=\S*?aulast=\S*!'
  rule:
    - pattern:
        - rewrite: '!(.*)!$1;type=2!'
        - match: '!(temp/\S*?\.pdf)"!'
  test:
    url:
      - http://www.indianjcancer.com/article.asp?issn=...
```

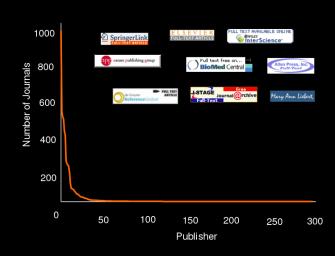
► 6574 journals (Nov 2008)

- 6574 journals (Nov 2008)
- ▶ 302 publisher

- ► 6574 journals (Nov 2008)
- ▶ 302 publisher



- 6574 journals (Nov 2008)
- ▶ 302 publisher



#### Some sites are easy

```
Springer:
    signature:
    url:
        - '!springerlink!'
    rule:
        - pattern:
        - match: '!(/content/\S*fulltext\.pdf)!'
```

#### Some sites are easy

```
Springer:
   signature:
     url:
       - '!springerlink!'
   rule:
     - pattern:
         - match: '!(/content/\S*fulltext\.pdf)!'
  NPG:
    signature:
      url:
        - '!nature\.com!'
    rule:
      - pattern:
          - match: '!(/\S*\.pdf)!'
```

### Implementation

- Perl module available that implements an intelligent agent that
  - interprets site driver files and searches for PDFs
  - supports all necessary browser technologies (cookies, proxy...)
  - validates file type identity
  - downloads PDF
- Currently 17 proof-of-concept drivers available covering vast majority of papers in PubMed.
- No site encountered yet that could not be crawled using simple signature/match/rewrite rules.
- Automatic test suite to routinely validate set of drivers

#### Conclusions

- The lack of automatic access to PDFs hindered the development of any sane and complete literature management system for years.
- I have shown a simple yet robust open source solution for the problem which can be part of any larger system in any language.
- Note for Pubget and QUOSA: I wish you good luck with a business essentially built on proprietary software that can be replaced by 220 lines of Perl.

github.com/wash