

Summary of Harvesting Working Group meeting

2011 General Assembly – The Hague
May 11-12.

How many websites are there?

First item on the agenda was a summary of an e-mail discussion about the size of ccTLDs that occurred on the IIPC members mailing list. The summary in appendix A was presented.

In particular the group discussed the different manners in which we classify a domain as ,active' (has a webserver, is not parked etc.). Also, the fact that the same content (i.e. site) was often served from multiple domains.

To discuss duplicate (or ,alias') domains, Bjarna Andersen of Netarchive.dk presented how his institution has handled this. See slides: alias_bjarne_iipc_may2011.pptx.

Dynamic websites

René Voorburg (KB) presented the Google #fragments way of making dynamic sites crawlable. See slides: hashbang.odp

Lewis Crawford (BL) then presented his attempt at crawling an IIPC test page he set up on Facebook. See slides: facebook.ppt

Results from BnF and Netarchive.dk were also presented. BnF's results were mixed, however Netarchive.dk seemed to have done quite well. See appendix B for the crawl configurations that they used.

Heritrix 3.1

Gordon Mohr of the Internet Archive presented the new features in the upcoming Heritrix 3.1. The features are summarized here: <https://webarchive.jira.com/wiki/display/Heritrix/Release+Notes+-+Heritrix+3.1.0-beta>

Heritrix 3.1 is in beta release and a final release will be made shortly.

Some members expressed a desire to see more regular (even if more limited in scope) stable releases of Heritrix.

Joint session on Quality Assurance

There was a Joint Session on Quality Assurance with the other working groups.

Browsers as crawlers

There was a general discussion on this topic. It was mostly intended to explore possibilities. This included ideas such as integrating a 'browser helper' into the crawler.

Some issues were identified with using browsers as crawlers:

- Scaling
- Duplicates
- Overloading a site
- Exercising the page (forms!)
- Browser wouldn't obey robots.txt

Some of these issues might be addressed by having the browser go through a 'crawler' proxy.

The idea was also proposed that a human operator might assist a crawler by using a browser by creating 'click through' macros.

Purpose of the HWG

Finally the group considered the focus of its work. It has moved from a purely technical working group to encompass less technically oriented members, such as curators.

The discussion made clear that the members want the group to retain its technical focus, tackling such challenges as rich media, dynamic websites etc. There was also a desire for the group to enable curators to use what solutions are already available, i.e. knowledge transfer.

Collection building issues, such as permissions etc, have typically not been on the group's agenda, but some members felt that it may be advantageous to include this.

It is possible that in the future the HWG will schedule part of its time on concurrent highly technical and collection building topics.

Future work within the HWG

The following items of work received commitment.

Create a good generic (W)Arc writer proxy

I.e. a proxy that writes material as requests are passed through it. There are currently several such tools in use in various member organizations. Gordon Mohr (IA) has agreed to collect information about the existing tools in order to build a requirements document. Kristinn Sigurðsson (NULI) offered to assist.

Report on scalability of Selenium type approaches.

Many members look to Selenium as a solution to automated quality assurance. It is however not well understood how it scales. Lewis Crawford (BL), Søren Vejrup Carlsen (Netarchive.dk) and Vinay Goel (IA) agreed to work on this.

Run Selenium with a plugin

Is it possible for Selenium to interact with websites, most likely using INA's plugin. Matt Holden (INA) and Lewis Crawford (BL) agreed to work on this.

Distributed crawling

Youssef Eldakar (Bibliotheca Alexandrina) expressed a strong interest in how crawling could be performed in a distributed manner. He and Gordon Mohr (IA) will investigate this topic.

Disseminate crawling knowledge for difficult high value sites

In particular YouTube and Facebook. Share crawling profiles on a WIKI.

Possibly organize a tutorial for future meetings.

Kristinn Sigurðsson (NULI) will organize. Vinay Goel (IA) and Bjarne Andersen (Netarchive.dk) agreed to provide the initial 'seed' material.

Appendix A

Country code TLDs:

Country	Registered ccTLDs	Active domains	Large domains	Population	Domains/100 people
Austria	1.040.000	86%		8.360.000	12,4
Czech Republic	750.000	70%*		10.500.000	7,1
Denmark	1.100.000	80%	11%	5.500.000	20,0
Finland	250.000			5.400.000	4,6
France	2.000.000	50%**	11%	62.600.000	3,2
Iceland	32.500	85%	9%	320.000	10,2
Netherlands	5.000.000			16.500.000	30,3
Norway	515.709			4.900.000	10,5

* Excludes non-response and 'parked' domains

** More than 10 documents

Global TLDs:

TLD	Registered gTLDs
.COM	95.006.677
.NET	14.003.416
.ORG	9.639.660
.INFO	8.200.168
.BIZ	2.254.683
.US	1.888.739
Total	130.993.343

Source: <http://www.whois.sc/internet-statistics/>

Appendix B

BnF configurations for crawling Facebook

The main idea in our profile is to crawl only URIs from facebook.com which are directly related to a specific Facebook user or group. We identify those URIs by the numeric user or group ID or by the user or group name contained in the URI.

We use a Heritrix (1.14.4) profile which is based on a SURT prefixed scope. At first in the decide rule sequence, we 'reject' anything from facebook.com. Then, we 'accept' only URIs from facebook.com containing a user ID or a group name that we want to crawl. This makes sure that the robot will stay on user or group related pages and would not break out to crawl the entire Facebook site.

```
<newObject name="scope" class="org.archive.crawler.deciderules.DecidingScope">
  [...]
  <newObject name="decide-rules" class="org.archive.crawler.deciderules.DecideRuleSequence">
    <map name="rules">
      <newObject name="rejectByDefault" class="org.archive.crawler.deciderules.RejectDecideRule"/>
      <newObject name="acceptIfSurtPrefixed"
class="org.archive.crawler.deciderules.SurtPrefixedDecideRule">
        <string name="decision">ACCEPT</string>
        <string name="surts-source-file"/>
        <boolean name="seeds-as-surt-prefixes">true</boolean>
        <string name="surts-dump-file">surts-dump.txt</string>
        <boolean name="also-check-via">>false</boolean>
        <boolean name="rebuild-on-reconfig">true</boolean>
      </newObject>
      <newObject name="rejectPath" class="org.archive.crawler.deciderules.MatchesListRegExpDecideRule">
        <string name="decision">REJECT</string>
        <string name="list-logic">OR</string>
        <stringList name="regexp-list">
          <string>^http://.*\.facebook\.com/.*$</string>
        </stringList>
      </newObject>
      <newObject name="acceptPathWithParameter"
class="org.archive.crawler.deciderules.MatchesListRegExpDecideRule">
        <string name="decision">ACCEPT</string>
        <string name="list-logic">OR</string>
        <stringList name="regexp-list">
          <string>^http://.*\.facebook\.com/. *123456789.*$</string> <!-- numeric Facebook user or group
ID -->
          [...]
          <string>^http://.*\.facebook\.com/. *name.*$</string> <!-- user or group name on Facebook -->
          [...]
        </stringList>
      </newObject>
```

All the other decide rules and parameters (max-hops etc.) are nothing special to Facebook.

Netarchive.dk configurations for crawling Facebook.

Filter:

```
<newObject name="facebook.com" class="org.archive.crawler.deciderules.MatchesListRegExpDecideRule">
  <string name="decision">REJECT</string>
  <string name="list-logic">OR</string>
  <stringList name="regexp-list">
    <string>.*facebook\.com\/ajax\/intl\/language_dialog\.php.*</string>
    <string>.*developers\.facebook.com\/.*</string>
    <string>.*facebook\.com.*(careers|privacy|help|campaign|facebook|badges|find-
friends|mobile|ilike\/artist|\/\/|browse\/\?type=favorite_pages).*</string>
    <string>.*facebook\.com\/((terms|profile)\.php|family).*</string>

    <string>.*facebook\.com.*(ROADRUNNER_READY|(\.(11|0\4))|\/J|\.18|action=recommend|\/v9\/0w|\/\
d{10}\.d{4})$.*</string><string>.*facebook\.com.*(login|[a-zA-Z0-9_-
]{80,}|captcha_challenge_code=|flixster|time=|user_birthday|authp=nonce|causes|redirect_uri=|\/\d{1,3}\
.\d{1,3}\.\d{1,3}\.\d{1,3}$).*</string>
    <string>.*facebook\.com\/directory\/(pages|people)\/[A-Z0-9]{1,2}$.*</string>
    <string>.*(connect|\/[^da]{2}-[dk]{2})\.facebook\.com.*</string>
    <string>.*facebook\.com.*(inbox|addfriend|browse\/\?type=likes|share_dialog).*</string>
    <string>.*connect\.facebook\.com.*[a-z0-9]{25,}.*</string>
  </stringList>
</newObject>
```

Extractor:

```
<newObject name="ExtractorImpliedURIDoubleSlash"
class="org.archive.crawler.extractor.ExtractorImpliedURI">
  <boolean name="enabled">>true</boolean>
  <string name="trigger-regexp">(^http.*:\/\/.*)\/(.*)</string>
  <string name="build-pattern">$1/$2</string>
  <boolean name="remove-trigger-uris">>false</boolean>
</newObject>
```