

Sampling frequency tuning tool

IKT 407

Group 1

Initial Summary:

- Introduction
- Literature survey
- Problem statement
- Requirements

Introduction:

- Crawler

- Periodical Crawler

- Incremental Crawler

- Others

- Frequency

- What is it

- Why important

■ Our jobs

- Improve the crawler with new algorithm
- Focus on accessibility's changes
- Assume each webpage costs same
- Ignore sub-pages and hyperlinks

■ Further study

Literature survey:

- 《Incremental Web Crawling as a Competitive Game of Learning Automata》

Svein Arild Myrer Morten Goodwin Olsen

- Web Content Accessibility Guidelines 1.0

- Documentation of Harvestman

Problem statement:

- Accessibility' s changes
 - What shall we do in our project
 - How can we monitor those changes

- Algorithm
 - Regular algorithm using nowadays
 - How does it work
 - Disadvantage

Site:	s3	s2	s1	s3	s2	s1	s3	s2	s1
Time:	t1	t2	t3	t4	t5	t6	t7	t8	t9

- Any other way?
 - Focused Crawler
 - Incremental Crawler

■ Implements

Requirements:

- Web Content Accessibility Guidelines 1.0

<http://www.w3.org/TR/WCAG10/>

- Web structure mining

- HTML-tag

<http://www.w3.org/TR/WCAG10-HTML-TECHS/>

- CCS

<http://www.w3.org/TR/WCAG10-CSS-TECHS/>

- Algorithm
 - Incremental crawler
 - Knapsack problem
 - Relationship
 - How des it work
 - Other

<i>Knapsack</i>	<i>Web</i>
Knapsack size	Crawler capacity
Fraction of item	Polling rate of web page
Value	<not used in this project>
Weight of item	Polling cost (always same)
Number of item	Number of web page

```
for  $t_i$  in  $T$ 
  for page  $i$  in  $I$ 
     $r = \text{random}(0, 1)$ 
    if  $r < F_i(n)/N$ 
      visit page  $i$ 
    if  $j$  is updated && not exceed capacity
       $F_i(n)++$ 
```

else if j is not updated && exceed capacity

$F_i(n)$ —

N : resolution

Large $N \Rightarrow$ accurate but slow

Small $N \Rightarrow$ inaccurate but fast

The first $F_i(n)$ should be $\text{random}(1, N)$

■ Check implements & Result verification

THANKS ALL