

Palestine Polytechnic University



College of Engineering & Technology
Electrical & Computer Engineering Department

Graduation Project

**An Autonomous high performance Web Crawler using Multi- Agent
(AWCMA)**

Project Team

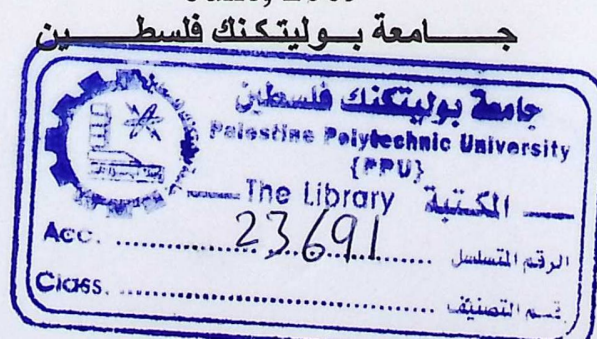
Haitham Hijazi

Moaid Zalloum

Project Supervisor
Dr. Radwan Tahboub

Hebron-Palestine

June, 2009



الخليـل- فلسطـين
كلية الهندسة والتكنولوجيا
دائرة الهندسة الكهربائية

اسم المشروع

**An Autonomous high performance Web Crawler using Multi- Agent
(AWCMA)**

أسماء الطلبة

هيثم حجازي مؤيد زلوم

بناء على نظام كلية الهندسة والتكنولوجيا وإشراف ومتابعة المشرف المباشر على المشروع وموافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية وذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة أنظمة حاسوب.

توقيع المشرف

.....

توقيع اللجنة الممتحنة

.....
توقيع رئيس الدائرة
.....

Abstract

Web navigation and search are vital research fields in which efficient ways for indexing and downloading pages are required.

Due to the tremendous growth of the web documents and high rate of changes (updates); traditional crawling methods are becoming less efficient as specially as the network traffic it causes.

In this project a multi agent system is used to realize mobile crawlers using the agent messages technology so that the crawling process is done at the web server side, pages parsed and processed locally then the result messages returned back to the crawler manager side to be integrated into the database.

A monitoring agent will be used in these web servers to detect changes in their documents and tells the search engine for immediate updates.

A new idea is introduced in this project that is to have a hybrid system which encompasses the agent technology with traditional crawling method, to solve the problem of overloading the web server resources when the crawling processes are done there, to achieve that we will use high power computers as Coocrawlers to handle the crawling process of the web server suffering from resource utilization under specific conditions tested by performance Tester agents.

المخلص

إن عملية التنقيب والبحث عبر الشبكة العنكبوتية من الحقول الحيوية للبحث الذي يتضمن الحاجة إلى طرق فعالة في فهرسة وتحميل صفحات الويب .

مع التزايد الهائل لمحتويات الانترنت وسرعة التغييرات والتحديثات لهذه المحتويات ، أصبحت الطرق التقليدية لفهرسة الويب (web crawling) غير فعالة وتسبب حمل زائد على الشبكة.

في هذا المشروع نستخدم تقنية العميل المتنقل (mobile agent) وهو برنامج لديه القدرة على الانتقال من جهاز حاسوب إلى آخر عبر الشبكة. وبالتالي يصبح لدينا ما يسمى بالمفهرسات المتحركة (mobile crawlers) والتي تقوم باختيار الصفحات وتحميلها في جهة خادم الويب (web server) المراد فهرسته وتحليلها وضغطها من ثم إرجاع النتائج إلى جهة محرك البحث ليكون قابلاً للاستعلام لاحقاً وبالتالي نكون قد خففنا العبء على الشبكة.

أيضاً سنستخدم مفهوم العميل المراقب (Monitoring agent) الذي سيراقب التغييرات في خادم الويب ونقلها إلى محرك البحث ليكون محدثاً أولاً بأول.

والفكرة الجديدة في هذا الموضوع هو عملية مراقبة الأداء للخوادم بشكل مستمر لاستخدام المفهرسات المساعدة بدلاً من العمل على مصادر هذه الخوادم وإرهاقها وتتم عملية استخدام المفهرسات المساعدة ضمن شروط واختبارات معينة يقوم بها العميل الفاحص للأداء.

Acknowledgment

We would like to express our deep gratitude to Dr Radwan Tashboul offered his time and efforts to successfully supervise this work.

Dedication

To all who are interested

To our families

Specially to Adv Wail Hijazi

To our friends

And to our university

We also want to thank Dr Amaal Dewail for answering our questions about the module spent at the first steps of our trip.

Finally we thank Eng Elad for his continuous consultations.

We will be always indebted to our parents, brothers and sisters who have inspired us and for their cordial patience and continuous moral support through our studying period for this course.

To all who opened his hand to help us we say - Thank you.

Acknowledgment

We would like to express our deep gratitude to Dr Radwan Tahboub offered his time and efforts to successfully supervise this work.

We are deeply indebted to our Palestine Polytechnic University in Hebron, West Bank whose full cooperation made it possible for us to gain technical award to do this course.

We wish to express our appreciation to Palestine Polytechnic University staff who was involved in our Bachelor of computer systems engineering encouraging us, socially and academically during the course period.

We also want to thank Dr Amal Dewaik for answering our questions about the mobile agent at the first steps of our trip.

Finally we thank Eng Eid Bader for continuous consultations.

We will be always indebted to our parents, brothers and sisters who have inspired us and for their cordial patience and continuous moral support through our studying period for this course.

To all who opened his hand to help us we say - Thank you.

List of Contents

Chapter 2: Theoretical Background

Title Page.....	I
Signature Page.....	II
Abstract	III
Dedication	V
Acknowledgments	VI
List of Contents.....	VII
List of Tables.....	X
List of Figures.....	XI

Chapter 1: Introduction

1.1 Preliminaries	2
1.2 Project vitality.....	4
1.3 Project goals.....	4
1.4 Literature Review	5
1.5 Requirements	6
1.6 Time table.....	7
1.7 Estimated cost.....	9
1.8 Project Risk management.....	10
1.9 Road Map.....	12
4.3 Agent messages and behaviours in JADE	57
4.4 Detailed modeling	60
4.5 GUI design	76

Chapter 2: Theoretical Background

2.1 Web crawler.....	14
2.2 A survey on web crawlers.....	22
2.3 Mobile agent.....	23
2.4 Indexing	26
2.5 Software description.....	27
2.6 Summery.....	31

Chapter 3: Design Concepts

3.1 Project Objectives in details	33
3.2 Architectural design	34
3.3 Design Options.....	44
3.4 AWCMA proposed algorithm	45
3.5 System modeling.....	48

Chapter 4: Detailed System Design

4.1 Overview	53
4.2 AWCMA block diagram in details.....	53
4.3 Agent messages and behaviours in JADE.....	57
4.4 Detailed modeling.....	60
4.5 GUI design.....	76

4.6	Challenges and improvements.....	78
-----	----------------------------------	----

List of Tables

Chapter 5: Implementation And Validation

5.1.	Development Environment Overview.....	80
5.2.	Implementation development and phases.....	81
5.3.	Testing and validation.....	86
5.4.	Challenges.....	96

Chapter 6: Results, Conclusions And Future Work

6.1.	Experiments and mathematical description.....	98
6.2.	Conclusions.....	107
6.3.	Future work.....	108
6.4.	Challenges.....	109

APPENDIX A	110
------------	-----

APPENDIX B	168
------------	-----

References	200
------------	-----

List of Tables

Table 1.1: Tasks description	7
Table 1.2: Time plan.....	7
Table 1.3: Tasks description- second semester.....	8
Table 1.4 : Time plan- second semester.....	8
Table 1.5: Software resources cost.....	9
Table 1.6: Hardware resources cost.....	9
Table 1.7: Risks analysis.....	10
Table 1.8: Risk impact IDs	10
Table 1.9: Risks effects minimization strategies	11
Table 4.1. CMA description (URLs).....	61
Table 4.2. CMNA description (Titles).....	62
Table 4.3. CMNA description (Keywords).....	63
Table 4.4. CMNA (Description).....	64
Table 4.5. Monitoring time description.....	65
Table 4.6. Serial crawling description	66
Table 4.7. Parallel crawling description	67
Table 4.8. Crawling strategy description	68

Table 4.9. NRCA Sequence diagram description	70
--	----

Fig. 3.9 Data flow diagram- Level 0	48
---	----

Fig. 3.10 Data flow diagram	49
-----------------------------------	----

List of Figures

Fig. 1.1 Google crawler example at PPU	3
Fig. 2.1 Search engines architecture	15
Fig. 2.2 URL frontier	16
Fig. 2.3 High level architecture of a standard web crawler	18
Fig. 2.4 Basic structure of mobile agent	24
Fig. 2.5 Mobile agent life cycle	25
Fig. 2.6 The main architectural elements of JADE platform	28
Fig. 2.7 The conceptual diagrams of Java SE platform	29
Fig. 3.1 General idea of the AWCMA	35
Fig. 3.2 The first strategy of AWCMA (Parallel).....	36
Fig. 3.3 The second strategy of AWCMA (Serial).....	37
Fig. 3.4 Block diagram of the AWCMA	38
Fig. 3.5 User interface of the AWCMA- prototype 1.....	39
Fig. 3.6 User interface of the AWCMA- prototype 2.....	40
Fig. 3.7 Results of AWCMA- prototype 3.....	41

Fig. 4.14 Crawling strategy flow chart	58
--	----

Fig. 3.8 Results of AWCMA- prototype 4.....	42
Fig. 3.9 Data flow diagram- Level 0.....	48
Fig. 3.10 Data flow diagram- Level 1.....	49
Fig. 3.11 Use case at crawler manager	50
Fig. 3.12 Use case at web server	50
Fig 3.13 Sequence diagram.....	51
Fig 4.1 Block diagram of the AWCMA.....	53
Fig 4.2 Web servers directory hierarchy.....	56
Fig 4.3. Message structure in JADE	57
Fig 4.4. The JADE asynchronous message passing paradigm.....	58
Fig 4.5. Agent thread path of execution.....	58
Fig 4.6. General flow chart of AWCMA system.....	60
Fig 4.7. CMNA flow chart for URLs.....	61
Fig 4.8. CMNA flow chart for web pages titles.....	62
Fig 4.9. CMNA flow chart for web pages Keywords.....	63
Fig 4.10. CMNA flow chart for web pages Description.....	64
Fig 4.11. Monitoring time flow chart.....	65
Fig 4.12. Serial crawling flow chart	66
Fig 4.13. Parallel crawling flow chart.....	67
Fig 4.14. Crawling strategy flow chart.....	68

Fig 4.15	Crawling and monitoring agent activity diagram.....	69
Fig 4.16	Interacting with AMS.....	70
Fig 4.17	NRA activity diagram.....	71
Fig 4.18	PTA activity diagram.....	72
Fig 4.19	CMA activity diagram.....	73
Fig 4.20	PRA activity diagram.....	74
Fig 4.21	PRA activity diagram.....	75
Fig 4.22	Configuration phase.....	76
Fig 4.23	Results phase.....	76
Fig 4.24	Performance statistics phase	77
Fig 4.25	System info and ethics view	77
Fig 4.26	Cocrawlers.....	78
Fig 5.1	NetBeans IDE.....	80
Fig 5.2	Prototype modeling	82
Fig 5.3	Implementation phases.....	83
Fig 5.4	NSClient4j service.....	85
Fig 5.5	Entering invalid number.....	86
Fig 5.6	Entering a unreasonable number.....	87
Fig 5.7	Selecting a invalid URL.....	88

Fig 5.8	Forgetting to select the strategy.....	89
Fig 5.9	Safe mode.....	90
Fig 5.10	Unsafe mode (Switch to the Coocrawlers).....	91
Fig 5.11	RMA at program startup.....	92
Fig 5.12	Transferring crawling and monitoring agent.....	93
Fig 5.13	Transferring crawling and monitoring agent in serial strategy (step 1).....	94
Fig 5.14	Transferring crawling and monitoring agent in serial strategy (step 2).....	94
Fig 5.15	Parallel strategy when selecting container -1.....	95
Fig 6.1	Content size Vs saved bandwidth.....	100
Fig 6.2	Data transferred in AWCMA system Vs traditional crawling.....	101
Fig 6.3	Data transferred in UCYMicra system Vs traditional crawling.....	102
Fig 6.4	Serial and parallel strategy comparison	103
Fig 6.5	Serial and parallel strategy comparison	106
A.1	a snap shot from keyword crawler during process.....	111
A.2	configuring the crawler.....	112
A.3	Win Web crawler ..running.....	115
A.4	Win Web crawler ..data show.....	116
A.5	Win Web crawler ..session configurations.....	117

Chapter one

Introduction

- 1.1 Preliminaries
- 1.2 Project vitality
- 1.3 Project goals
- 1.4 Literature review
- 1.5 Requirements
- 1.6 Time schedule
- 1.7 Cost estimates
- 1.8 Risk analysis and management
- 1.9 Road map

1.1 Preliminaries

Indexing the Web has become a challenge due to its growing and dynamic nature, the directly accessible Web (also mentioned as surface Web) exceeds 2.5 billion documents (in 2007) while the indirect Web (dynamically generated documents) is almost three orders of magnitude larger, moreover the Web is changing 40% of its contents every month while no search engine succeeds coverage of more than 16% of the estimated web size [22].

Suppose you want to make a search about a sudden event or result that has been recently published to the web, a question rises here can search engines detect these newly published documents? if that the case what are the effects over the network?

“Even search engines such as Google or Altavista cover only a limited part of the Web, and the majority of their archives are not updated”[23].

A web crawler (also known as a web robot or spider) is a program for downloading web pages. Given a set (s) of “seed” Uniform Resource Locators (URLs), the crawler repeatedly removes one URL from s , downloads the corresponding page, extracts all the URLs contained in it and the keywords for index goals, and adds any previously unknown URLs to (s) [11] .

To demonstrate what is a web crawler we will take Google crawler [28] (Googlebot) as our example, suppose you are an administrator of the university web site www.ppu.edu, and you want the Google search engine to index your web pages to be queried by any user, then you have to go to the link “<http://www.google.com/addurl>” and add your site there, after that Googlebot (crawler) will visit your web site , download pages, extract all URLs included in your site ,and continue.

Another option that is your web site may be already included in other popular sites as a hyperlink so it will be crawled (i.e. www.ppu.edu is included in the international universities union web site).

By this way the indexer of Google will be rich of thousands of websites, figure 1.1 demonstrates this example.

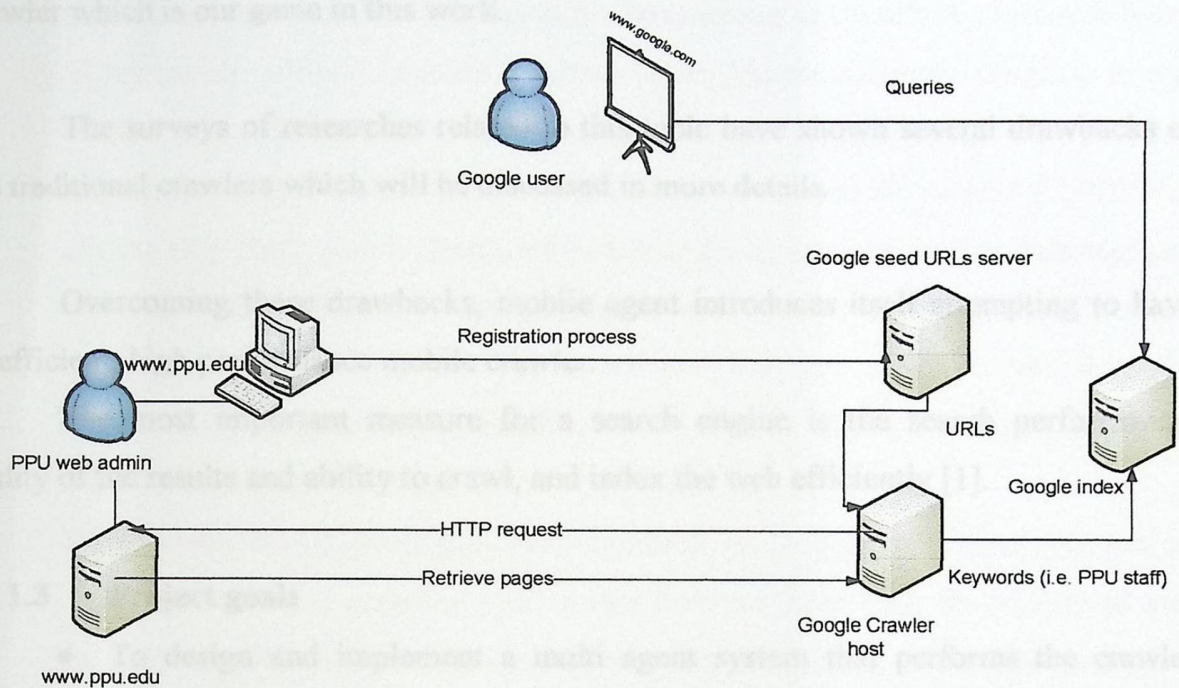


Figure 1.1 Google's crawler example at PPU.

Although the web crawling algorithm is conceptually simple, designing a high performance web crawler comparable to the ones used by the major search engines is a complex endeavor. All the challenges inherent in building such a high performance crawler are ultimately due to the scale of the web [11].

Mobile Agent (MA) is an executing program that migrates during execution and presents methods for maintaining and using distributed system [13].

In this project we will use mobile agent as crawler (mobile crawler) to crawl the web servers locally and return the results back to the crawler manager.

1.2 Project vitality

The World Wide Web is an essential organ in the body of information explosion running these days.

The most important element in this organ is the search service which has piles of usefulness to the web user as specially as the researchers.

Going further in more details the most crucial part of search infrastructure is the web crawler which is our game in this work..

The surveys of researches related to this topic have shown several drawbacks of the traditional crawlers which will be discussed in more details.

Overcoming these drawbacks, mobile agent introduces itself attempting to have an efficient, high performance mobile crawler.

The most important measure for a search engine is the search performance, quality of the results and ability to crawl, and index the web efficiently [1].

1.3 Project goals

- To design and implement a multi agent system that performs the crawler responsibilities in an efficient way.
- To overcome some of the drawbacks of current web crawlers related with performance, bandwidth and relevance.
- To keep the search engine's index updated in an approximate real time without much overload on the network.

1.4 Literature review

- “UCYMICRA”

Is a distributed indexing using migrating crawlers , UCYMICRA extends the concept of mobile agent as a crawler to build a fully distributed system based on mobile agent, its goals are to keep up with document changes rapidly, to minimize network utilization and to be upgradeable at run time [22]. We noticed that this system requires form the web administrator to specify the peak-off hours of his server while it is more accurate to implement a simple program to test whether it is appropriate to crawl locally or to give up the responsibility to other helping factor (i.e. Coocrawlers in AWCMA), we noticed also that UCYMICRA got use only from mobile agents while stationary agents can do communicate and exchange information(i.e. configurations and results) through asynchronous messages, finally we suggest not to transmit the web contents even they are filtered and compressed, however only keywords and URLs are needed in AWCMA system to be indexed.

The owners of UCYMICRA recommended in their future work portion to implement a hybrid crawling system that borrows form both the traditional and the fully distributed crawling system, we say to them is it possible to have a hybrid crawling system that borrows from both the traditional and the multi-agent crawling system? This will possibly be answered through our research.

- “USING THE WEB EFFICIENTLY: MOBILE CRAWLERS”

In this paper a use of mobile agents is introduced to crawl the web servers , the main goal was there to minimize the load over network by filtering, and selecting the wanted relevant pages only from the web server, then returns with compressed results to crawler manager system [24].

- “Multi-Agent System for Specific Domain Search Engine Based on Distributed Classification Approach” (2007)

H. Arafat Ali used multi agent platform to achieve parallelism in focused crawling to overcome bandwidth and resource utilization in web server hosts.

This paper shows also how this way decreased the execution time of crawling and speed it up, after that a comparison was conducted between the client-server paradigm and mobile agent paradigm in crawling technology [25].

- “Mercator” Allan Heydon and Marc Najork

Is an extensible , scalable , high performance web crawler that are entirely written in java , it discussed the main components of any crawler and alternative design options for each component , Mercator is scheduled to be included in the next version of the AltaVista Search Intranet product a version of the AltaVista software sold mostly to corporate clients who use it to crawl and index their intranets [5].

1.5 Requirements

1.5.1 User requirements

- The system should have Graphical User Interface (GUI) in the host of crawler manger to control and monitor the crawling process.
- The GUI in crawler manager must show the extracted URLs and keywords.

1.5.2 System requirements

1.5.2.1 Functional requirements

- The system should take the seed URLs specified by the crawler manager user.
- The system should supply crawling configurations (i.e. max memory utilization, monitoring time and crawling strategy , ...).
- The system should send Performance Tester Agent (PTA) with the specified URLs to the web servers.
- In serial strategy only one Crawling and Monitoring Agent (CMNA) is employed so the user has to select the URL.
- In parallel strategy the user should select the URLs to be crawled .
- The system should create and dispatch CMNA to the specified web server.

- The mobile agent should read the pages specified, parse them, extract URLs ,titles and key words.
- The CMNA should send messages to the crawler manager with the crawled pages, new URLs ,titles and keywords of the pages to be indexed.

1.5.2.2 Non -Functional requirements

- Performance: the system must obey high performance measures (i.e. processes a large number of pages within a reasonable time).
- Efficiency: the system of crawling should be efficient in terms of network load and web servers' resources.

1.6 Time table

Table 1.1 distributes the tasks of project to IDs so that each one has its own time line.

In accordance table 1.2 shows clearly tasks' time plane managed over 16 weeks

Table 1.1. Tasks description

Task ID	Description
T1.1	Selecting the project
T1.2	Information gathering ,literature review ,reading
T1.3	System analysis , requirements determination
T1.4	Start designing
T1.5	Documentation writing

Table 1.2. Time plan

Task/week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T1.1																
T1.2																
T1.3																
T1.4																
T1.5																

Table 1.3. Tasks description- second semester

Task ID	Description
T2.1	Design and analysis
T2.2	Components implementation
T2.3	Component testing
T2.4	Integrity testing
T2.5	Documentation writing

Table 1.4. Time plan- second semester

Task/week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T2.1																
T2.2																
T2.3																
T2.4																
T2.5																

1.7. Estimated Cost

1.7.1. Software resources

The following table show the estimated costs for software components and tools used during our work.

Table 1.5. Software resources cost

Software component	Cost
Microsoft Windows Vista	\$250
Microsoft Office 2007 enterprise	\$170
Microsoft Office Visio	\$200
Net Beans IDE	Free
JADE Package	Free
NSClient4j	Free
Total	\$620

1.7.2. Hardware resources

Table 1.6. Hardware resources cost

Hardware component	Cost
1 PC with core 2 duo CPU 2.6 GHz, 160 GB HDD, 2 GB RAM, monitor, keyboard and mouse.	\$2000
4 PC Pentium 4 (1500-3000) MHz, 80 GB HDD, 512 MB RAM, monitors, keyboards and mouses.	
A LAN with 10 computers (for testing)	Free
Total	\$2000

1.7. Estimated Cost

1.7.1. Software resources

The following table show the estimated costs for software components and tools used during our work.

Table 1.5. Software resources cost

Software component	Cost
Microsoft Windows Vista	\$250
Microsoft Office 2007 enterprise	
Microsoft Office Visio	3
Net Beans IDE	2
	4
	2
JADE Package	
NSClient4j	
Total	

1.7.2. Hardware resources

Hardware component	Negligible
1 PC with core 2 duo 6 700 2GB RAM 160GB HD	
keyboard and mouse	
4 PC Pentium 4 (1500-3000) MHz, 2GB RAM, 160GB HD	
keyboards and mouses	
A LAN with 10 computers (for testing)	
Total	

1.8. Project Risk Management

The project may face some problems and risks that we have to declare in the early time of the project analyzing and manipulation, and we must avoid those problems so that the system could work in its high efficiency, so when we find a risk we will try to solve it without affecting the total project as much as we can..

Table 1.7 Risks analysis

Risk ID	Description	Probability	Impact ID
RA	One of the teamwork is absent	50%	1
RB	Little experience in using some programming tools	40%	3
RC	Hardware or network failure	20%	3
RD	Extend the project time than determined	40%	2
RE	Closure of university (suspension)	10%	4
RF	Irrelevant results	40%	2

Table 1.8 Risk impact IDs

Impact ID	Type of impact
1	Catastrophic
2	Critical
3	Marginal
4	Negligible

Table 1.7 Risks effects minimization strategies

Risk ID	Strategies to decrease the side effects of the risk
RA	the plane that if one is absent the other do his duty
RB	Read more tutorials about the tool and ask available experts
RC	Main failures have to be mended by specialists , backup plan
RD	Try to neglect some irrelevant sides of project
RE	Meet in other places
RF	Try to make the testing at each stage of project

1.9. Road Map

The work consists of seven chapters, each chapter talks specifically about an area of project.

Chapter One: “Introduction”, this chapter gives an introduction about the system, its importance, other related systems, project time plan, project cost and risk management.

Chapter Two: “Theoretical Background”, this chapter gives a clear picture about the system theoretical background related to the main concepts of crawling, agent, mobile agents, and web services , security related to mobile crawlers

Chapter Three: “Conceptual design”, this chapter shows system block diagram, design options that can be used in the system, and how each system components interface with each other.

Chapter four: “detailed design”, this chapter shows a detailed lens view to the design and modeling of the system including flowcharts , AUML , and GUI design.

Chapter five: “implementation and validation”, this chapter speaks freely about implementational issues and testing the application under various conditions with validation features, it speaks also about the software packages used and the development model adopted to carryout the work.

Chapter six: “Results, Conclusions and future work”, this chapter explains how much we achieved from our goals through experiments and resulted figures , statistics , and numbers, this chapter included the mathematical description of the system and the future work to be done.

Chapter Two

Theoretical background

Preface

In this chapter we explore the theoretical background of the web crawler and its properties. We start with a survey on many crawler systems. An introduction to the crawler is given in this chapter.

2.1 Web crawler

It is a challenging task to build a crawler that can handle the huge amount of data on the web. The crawler has to be able to handle the huge amount of data on the web. The crawler has to be able to handle the huge amount of data on the web. The crawler has to be able to handle the huge amount of data on the web.

2.1.1 How crawler works

Before proceeding to the crawler's life cycle we have to explore lightly the basic components from which it consists.

A crawler is a program that can be used to fetch the content of a web page.

- 2.1 Web crawler
- 2.2 Survey on web crawlers
- 2.3 Mobile agent
- 2.4 Indexing
- 2.5 Software description
- 2.6 Summary

Chapter Two

Theoretical background

Preface

In this chapter we explored the theoretical background of the web crawler and its proprieties, policies, components and a survey on many crawler systems.

An introduction for mobile agents was included in this chapter.

2.1. Web crawler

It is considered the most essential component in search engines, it is challenging task as tricky reliability, performance, and other issues [29].

“Crawling is the most fragile application since it involves interacting with hundreds of thousands of web servers and various name servers, which are all beyond the control of the system.” [1]

2.1.1. How crawler works

Before proceeding in crawler's life cycle we have to explore lightly the basic components from which it consists.

- A component (called the URL frontier) for storing the list of URLs to download.
- A component for resolving host names into IP addresses (DNS).
- A component for downloading documents using the HTTP protocol.
- A component for extracting links from HTML documents.

These are the main components of any crawler used in search engines.

Hear we can show the crawler as a part of search engines.

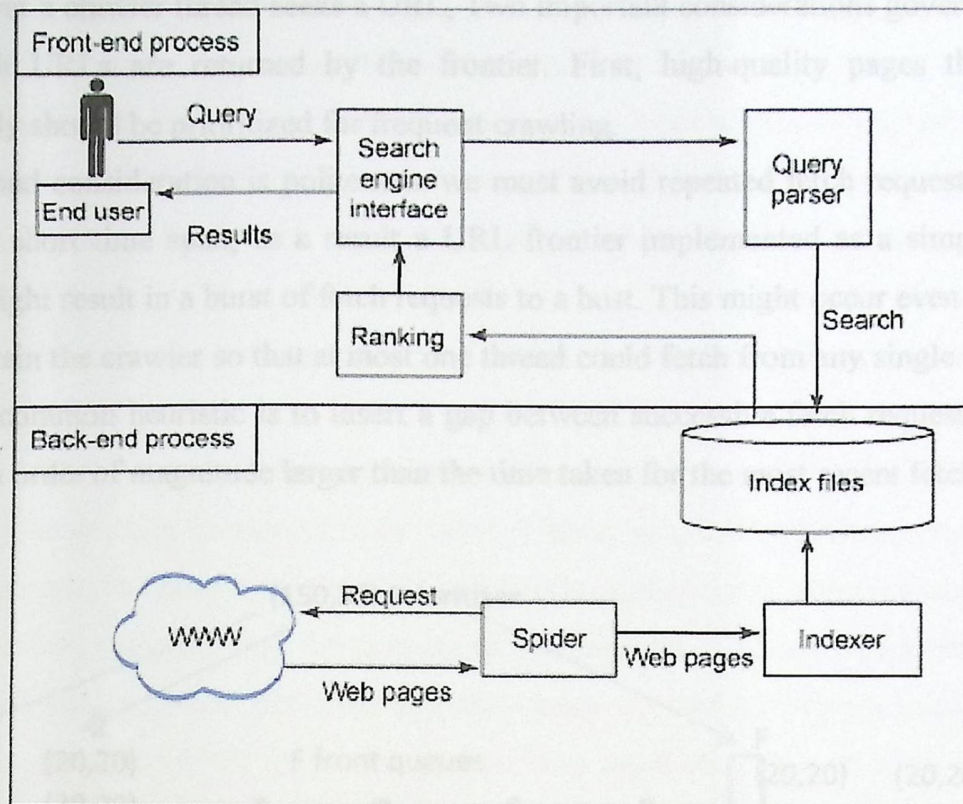


Figure 2 .1. Search engines architecture [1]

By describing the full function of each component we can easily draw the life cycle of the crawling process.

2.1.1.1 URL frontier

A single URL Server serves lists of URLs to a number of crawlers [2], the URL frontier at a node is given a URL by its crawl process, it maintains the URLs in the frontier and regurgitates them in some order when it is needed.

Whenever a crawler thread seeks a URL, Two important considerations govern the order in which URLs are returned by the frontier. First, high-quality pages that change frequently should be prioritized for frequent crawling.

The second consideration is politeness: we must avoid repeated fetch requests to a host within a short time span; as a result a URL frontier implemented as a simple priority queue might result in a burst of fetch requests to a host. This might occur even if we were to constrain the crawler so that at most one thread could fetch from any single host at any time. A common heuristic is to insert a gap between successive fetch requests to a host that is an order of magnitude larger than the time taken for the most recent fetch from that host.

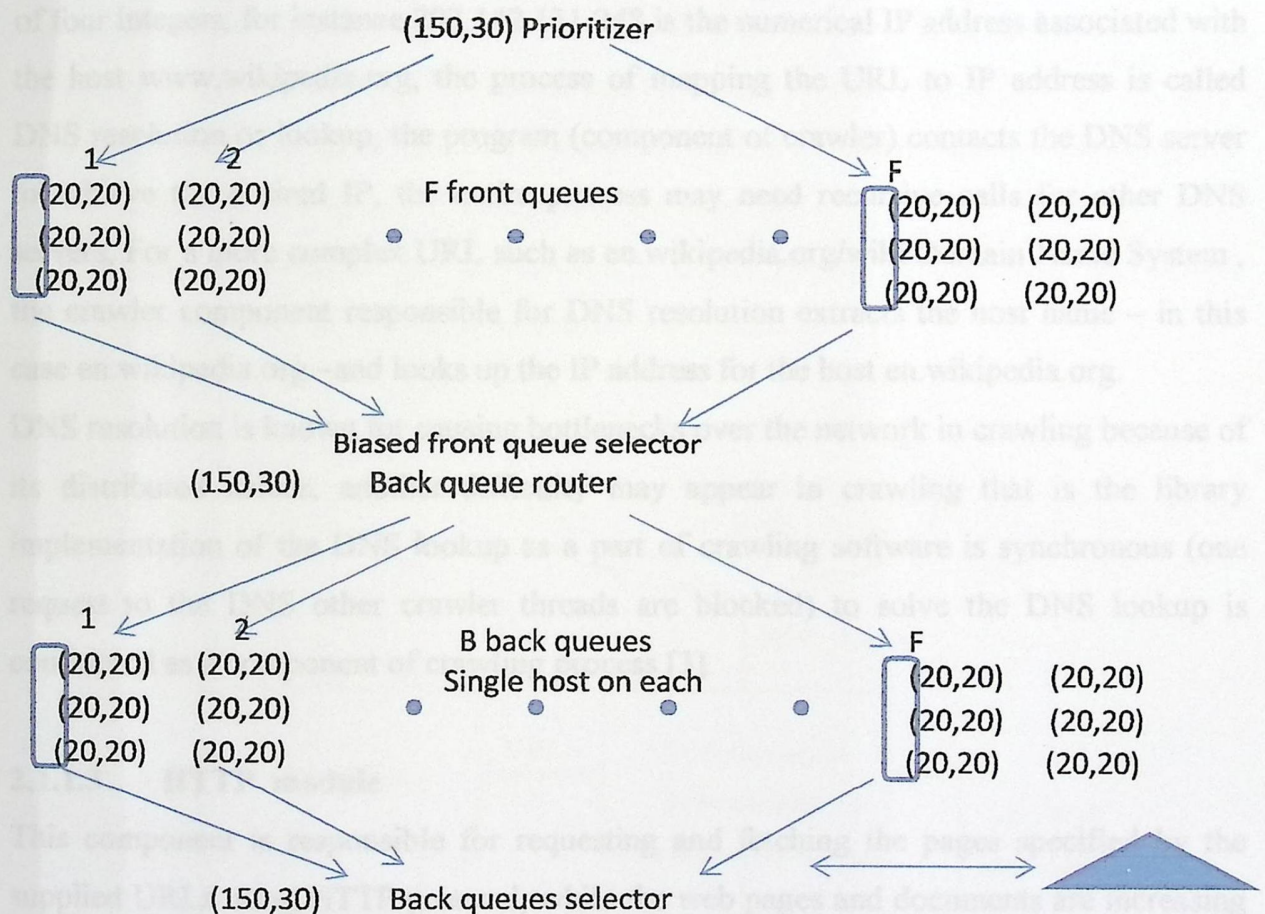


Figure 2.2. URL frontier [1]

Figure 2.2 .The URL frontier .URLs extracted from already crawled pages flow in at atop of the figure. Crawler requesting URL extracts from the bottom

The two major sub modules are F: front queues in the upper portion of the last figure and B: back queues (FIFO QUEUES) in the lower portion of the figure.

The front queues implement the prioritization, while the back queues implement politeness. [3].

Each URL is assigned an integer between 0-15 based on its fetch history

2.1.1.2 DNS (Domain Name Service) resolver

Each web server, host connected to the internet has a unique IP, which generally consists of four integers, for instance 207.142.131.248 is the numerical IP address associated with the host www.wikipedia.org, the process of mapping the URL to IP address is called DNS resolution or lookup, the program (component of crawler) contacts the DNS server to achieve the desired IP, the entire process may need recursive calls for other DNS servers, For a more complex URL such as [en.wikipedia.org/wiki/Domain Name System](http://en.wikipedia.org/wiki/Domain_Name_System) , the crawler component responsible for DNS resolution extracts the host name – in this case en.wikipedia.org –and looks up the IP address for the host en.wikipedia.org.

DNS resolution is known for causing bottlenecks over the network in crawling because of its distributed nature, another difficulty may appear in crawling that is the library implementation of the DNS lookup as a part of crawling software is synchronous (one request to the DNS other crawler threads are blocked) to solve the DNS lookup is considered as a component of crawling process [3].

2.1.1.3 HTTP module

This component is responsible for requesting and fetching the pages specified by the supplied URLs using HTTP protocol, while the web pages and documents are increasing rapidly the excessive use of HTTP get requests affected the overall network performance and utilization.

2.1.1.4. Download module

To have polite crawlers, downloading from one domain must be in a reasonable rate, while URLs from different domains shall be downloaded in parallel to maximize network

usage, to implement that two approaches are used 1. to use multithread each take a URL at a time from to do list 2. to use one process that asynchronously poll hundred of connections (like Google)[4], using multi thread make the program structure simpler because the duty of switching between threads are left to OS thread scheduler rather than to be coded in the program[5].

These are generally the main component of any crawler; simply we can summarize the process

- The URL frontier feeds the crawler manager with seed URL/s.
- Crawler manager resolves the URL/s to IP/s to connect with its server.
- Crawler makes the HTTP request and access the specified server.
- Crawler manager downloads the page/s wanted and start parsing, analyzing and extracting new URL/s and Keywords for indexing.
- Pages downloaded go to repository, URLs go to URL frontier for further crawling, and keywords go to indexer for search use.

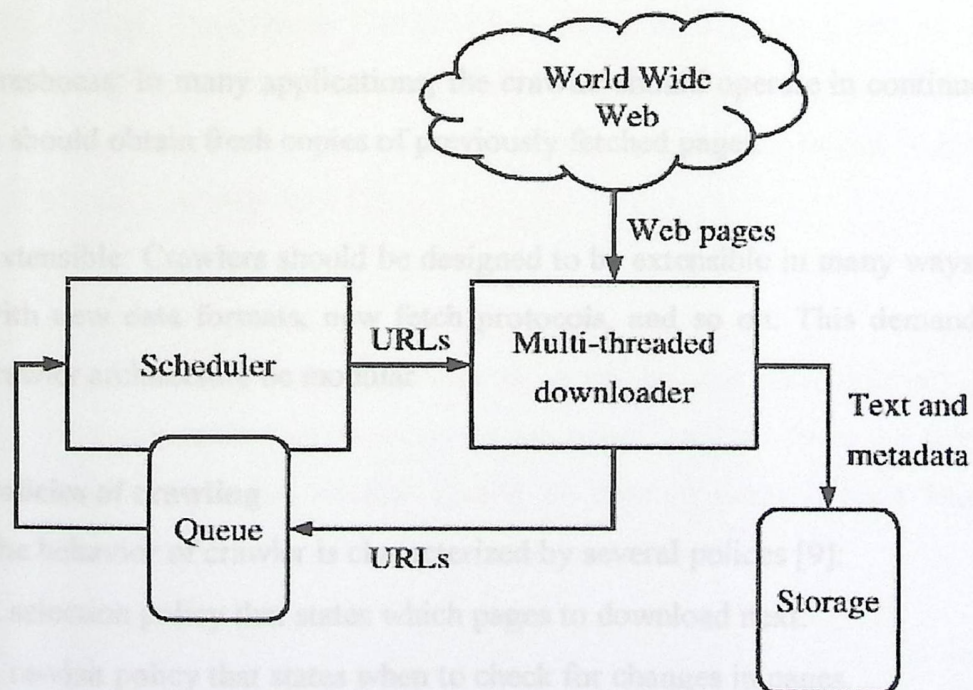


Figure 2.3 .high level architecture of a standard web crawler [2]

2.1.2. Essential features of crawlers [4].

The crawlers in general must obey many features to have a good web crawling these are:

1. **Politeness:** Web servers have both implicit and explicit policies regulating the rate at which a crawler can visit them. These politeness policies must be respected as we will show later.
2. **Robustness:** Some web servers create what is called “spider trap” which cause the crawler to fetch infinite number of pages from a particular domain, crawlers must be able to resilient such trap.
3. **Scalable:** The crawler architecture should permit scaling up the crawl rate by adding extra machines and bandwidth because of the vast increasing in web size.
4. **Performance and efficiency:** The crawl system should make efficient use of various system resources including processor, storage, and network bandwidth.
5. **Quality:** the crawler should harvest the useful pages that are related rather than gathering a huge fraction of web with no use to user query
6. **Freshness:** In many applications, the crawler should operate in continuous mode: It should obtain fresh copies of previously fetched pages.
7. **Extensible:** Crawlers should be designed to be extensible in many ways – to cope with new data formats, new fetch protocols, and so on. This demands that the crawler architecture be modular

2.1.3. Policies of crawling

The behavior of crawler is characterized by several policies [9]:

- A selection policy that states which pages to download next.
- A re-visit policy that states when to check for changes in pages.
- A politeness policy that states how to avoid overloading websites.
- A parallelization policy that states how to coordinate distributed web crawlers.

2.1.3.1 Selection policy

This policy simply decides which page to process next; Selection policies are based on graph exploration algorithms.

The most common used search algorithms are breadth first search (BFS), search with Page rank calculations, focused crawling strategy, OPIC (online page importance computation) strategy and depth-first search (DFS) [9], web crawler generally implements breadth-first based on local site priority.

Breadth-first crawl has a higher probability of visiting pages with a high page rank, breadth first strategy Visits all URLs pointed by the first page before visiting any URLs pointed to by pages fetched later so it crawl pages as they are discovered[7] .

Back-links selection based on page rank metric, pages with higher page rank crawled first, page rank take the ingoing links to the web site in its function calculation [9].

Focused crawling tries to seek, crawl, and index pages in specific topic which invest the network resources and achieve respectable coverage at a rapid rate [8].

OPIC (Online Page Importance Computation) Here, each pages given an amount of “cash” that is distributed equally among the set of pages that it points to. From a given page, the crawler assigns download priority among neighboring linked pages based on the amount of cash that they have [6].

2.1.3.2. Re-visit policy

The web is dynamic with lots of additions, deletions and modifications happening frequently. This puts additional pressure on the search engines from the freshness and relevance standpoint [6], the crawler should be able to crawl a page based on the frequency of its change and must operate continuously [3].

2.1.3.3. Politeness policy

The web crawler comes with cost include [9]:

- Network resources. Crawlers require considerable bandwidth.
- Server overload.
- Poorly written crawlers.
- Personal crawlers.

The partial solution of this problem is the use of robots exclusion protocol or (robot.txt) which is the responsibility of administrators to indicate the parts of their servers not allowed to be accessed by the crawler [2], also another protocol rose on the surface to avoid multiple visits to the same domain that is to use delay between visits to the same site this point is very important especially to the servers visited many times a day [11].

2.1.3.4. Parallelization policy

This policy aims to run multiple processes to carry out crawler responsibilities, it will achieve [10] :

- Maximizing download rate.
- Minimizing network overhead.
- The central manger assigns dynamically new URL\s to each crawling process.
- This will allow the central server balance the load over crawlers. [2].

While this policy serves a lot, it has some disadvantages that many processes crawl the same pages (overlap), and the communication bandwidth which could be utilized highly in processes communication among each others, and the quality of pages (their importance) that is not taken in processes consideration when run multiply [12].

2..2. A survey on web crawlers [4,5,11,26,27]

The first web crawler was born in spring 1993 [5], because of the comparative natural of search engines, the designs of these crawlers have not been publicly described, we can say that we have some exception such as Google crawler , internet archive and Mercator.

- Original Google crawler : consisted of five components :
 - a) URL server process: reads URLs from a file and distribute them to multiple crawler machines.
 - b) Crawler process: ran on a different machine, was single thread, used asynchronous I/O to fetch up from 300 servers.
 - c) Store server process: to which the downloaded pages are stored and compressed.
 - d) Indexer process: This reads the pages from the store server, extract the links from HTML pages and save them to different disk file.
 - e) URL resolver process: which reads the links file, normalize the URLs and save the absolute URLs to be read again by the URL server.
- (Heritrix) Internet archive crawler: also ran on multiple machine crawling, each machine was assigned to crawl up to 64 sites, used asynchronous I/O to fetch pages, once the pages downloaded the links extracted [4,27], "If a link referred to the site of the page it was contained in, it was added to the appropriate site queue; otherwise it was logged to disk. Periodically, a batch process merged these logged "cross-site" URLs into the site-specific seed sets, filtering out duplicates in the process" [5].
- Mercator: very popular implementation of web crawler it consists of the following :
 - a) A component (called the URL frontier) for storing the list of URLs to download.
 - b) A component for resolving host names into IP addresses.
 - c) A component for downloading documents using the HTTP protocol.
 - d) A component for extracting links from HTML documents.

- e) A component for determining whether a URL has been encountered before [26,5].

2.3. Mobile agents

2.3.1. Definition of an agent

It can be defined as: a software object that is situated within an execution environment. Taken that there is no commonly agreed definition of exactly what an agent is [18]

While transferring to another definition says: "The term agent is used to represent two orthogonal concepts. The first is the agent's ability for autonomous execution. The second is the agent's ability to perform domain oriented reasoning." [20]

The proprieties of such agents can be described as following:

- Autonomy: has its control over itself.
- Social ability: through communicating with other agents.
- Reactive: to act as a sensor for the environment changes and respond accordingly.
- Pro activeness: initiative to directed goal behavior.
- Learning: through previous experience.
- Continuity: continuous running process [20], [15].

2.3.2. Definition of mobile agent

A mobile agent is a program or object (code and data) that can migrate from host to host in a network of heterogeneous computer systems and fulfill a task specified by its owner using the resources of the hosts [19].

2.3.3. Mobile agent components

The three basic components of mobile agent are:

- The code: it is the executed aspect of mobile agent; simply mobile agent holds single code.

- The state: it is the data execution environment of the agent such as stacks, program counter.
- The data: variables, identifiers, knowledge [15].

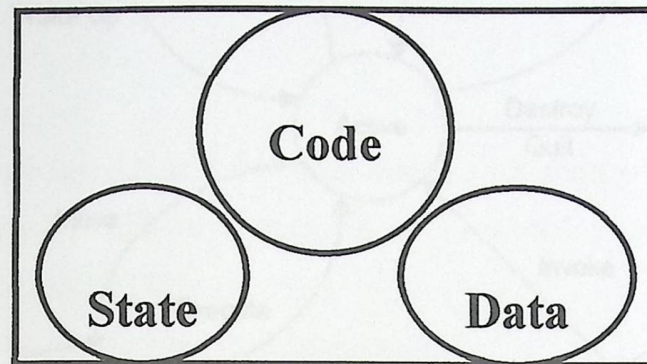


Figure 2.4 .Basic structure of mobile agent[15]

2.3.4. Mobile agent life cycle

An agent, as a physical software process, has a physical life cycle that has to be managed by the agent platform [19].

This life cycle can be drawn through figure 2.5 and the following points.

- Creation: the agent is created (installed).
- Invoke: The invocation of a new agent.
- Destroy: The forceful termination of an agent.
- Quit: the termination
- Suspend: no thing to do.
- Resume: resume from a suspended state.
- Wait: Puts the agent in a waiting state, agent initiated.
- Wake Up: Brings the agent from a waiting state.
- Move (Migrate): moves the agent to another system.
- Execute: to execute in the active state.

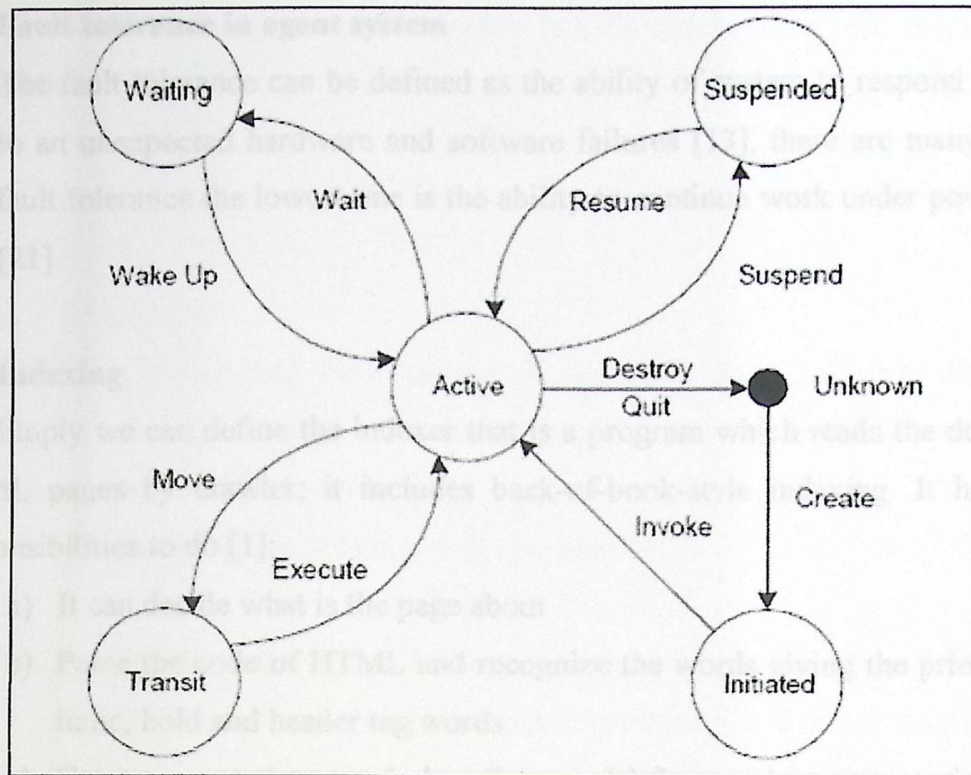


Figure 2.5 .Mobile agent life cycle [19]

2.3.5. Mobile agent security requirements

The mobility nature of agent exposes it to many threats: agent to platform, agent to agent, platform to agent and other to agent threats so mobile agents have to provide essential requirements for security [20].

- Confidentiality: means all the activates and information are hidden.
- Integrity: ensure that no unauthorized agents or process can modify the data or the code.
- Authentication of origin: ensure the originality of communication from its claimant.
- Availability: ensure that to provide the communication, data and services to agents in a timely manner.
- Non-repudiation: the originating entities are not allowed to deny their communications [15].

2.3.6. Fault tolerance in agent system

The fault tolerance can be defined as the ability of system to respond gracefully to an unexpected hardware and software failures [13], there are many levels of fault tolerance the lowest one is the ability to continue work under power failure [21].

2.4. Indexing

Simply we can define the indexer that is a program which reads the downloaded HTML pages by crawler; it includes back-of-book-style indexing. It has several responsibilities to do [1]:

- a) It can decide what is the page about
- b) Parse the code of HTML and recognize the words giving the priority to the italic, bold and header tag words.
- c) Creates a metadata vocabulary (keywords) for search engine queries.

It is considered challenging task as the web page contains a large number of words, and imagine thousands of pages!

2.4.1. Methods of indexing [1,2].

- 1) Full text indexing: Google, Alta vista, Excite are example of full text indexing as it implies from its name it store each word parsed and found in HTML page in the search engine's database.
- 2) Keyword indexing: Lycos is the example; here only the important phrases or words are stored.
- 3) Human indexing: Yahoo, while the spider was responsible for indexing hear the human interact to index, a person chooses some words and phrases that describes the page.

So indexing the web includes building of a web category that contains the words found in the page parsed and their occurrences.

The indexers have to be updated periodically as new contents are being crawled.

The success of search engines relies heavily on building and organizing the indexers efficiently.

2.5. Software description

2.5.1. The JADE

“JADE (Java Agent Development Framework) is a software development framework aimed at developing multi agent systems and applications conforming to FIPA standards for intelligent agents” [12].

It is fully implemented in java programming language [15].

2.5.1.1. JADE architecture

The JADE platform consists of agent containers which are distributed over the network, agents live there which could be described as Java process that provides the JADE runtime and all the services needed for agents [14].

There is a special container, main container which is the first one launched and represents the bootstrap point of the platform.

All other containers register to the main container and they are given logical names e.g. Container-1, Container-2...etc. [12].

The following figure shows the main architectural elements of JADE platform .

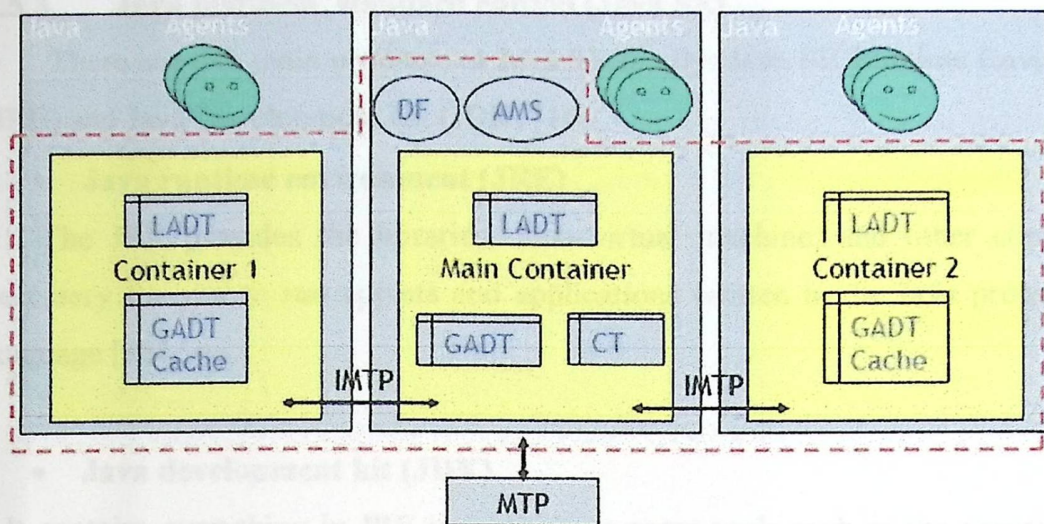


Figure 2.6 .The main architectural elements of JADE platform [15]

2.5.1.2. FIPA (Foundation of Intelligent Physical Agents)

FIPA is an international organization established in 1996 [15] that aims to promote the industry of intelligent agents by adopting specifications that support interoperability among agent systems, the heart of FIPA's model for agent systems is agent communication that is how the agent semantically can pass message to others agents to execute what are they required for [19].

2.5.2. Java programming language

It is an object oriented programming language that was released by Sun Microsystems in 1995; it derives much of its syntax largely from C++ with simpler object model.

Java applications are compiled to java bytecode that can run on any Java Virtual Machine (JVM) which guarantees the portability of this programming language and platform independence without caring to the architecture and operating system in which it is run, this capability referred to as "Write once, run everywhere" [2].

Java facilities data and applications (Applet) distribution over the internet [13].

2.5.3. Java platform, standard edition (Java SE)

There are two main products in Java SE family: Java SE Runtime Environment (JRE) and Java Development Kit (JDK) [16].

- **Java runtime environment (JRE)**

The JRE provides the libraries, Java virtual machine, and other components necessary for you to *run* applets and applications written in the Java programming language [16].

- **Java development kit (JDK)**

It contains everything in JRE in addition to some tools such as the compilers and debuggers needed for developing applications, figure 2.5 will show the whole components of Java SE platform [16].

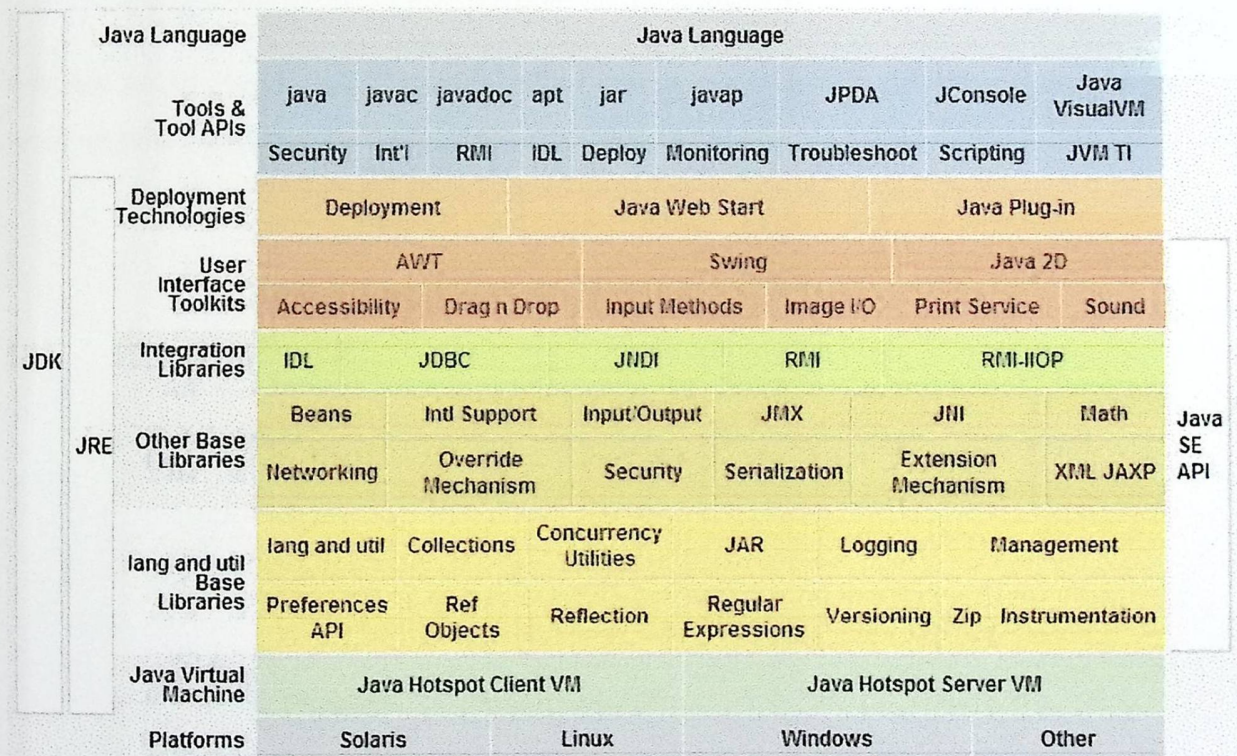


Figure 2.7 .The conceptual diagrams of Java SE platform component technologies [16]

2.5.4. NetBeans IDE (Integrated Development Environment)

Is a free, open source IDE for software developers. You get all the tools you need to create professional desktop, enterprise, web, and mobile applications with the Java language, C/C++, and even dynamic languages such as PHP, JavaScript, etc... , it can run on many platforms including Windows , Linux, Mac OS X and Solaris , add to that its ease of install and use [17].

2.6. Summary

A web crawler is a program that downloads the pages from web servers using HTTP protocol, extract the pages to find URLs and Keywords, store the pages in caches and indexers for further search queries.

Web crawlers have many features and polices, many algorithms were developed to achieve high performance and efficiency in crawling

Indexing is an essential part of search operation to have a powerful search engine.

A mobile agent is a program or object (code and data) that can migrate from host to host in a network of heterogeneous computer systems and fulfill a task specified by its owner.

There are many features and security issues related with mobile agents that make any related work on agents extensible and upgradeable .

Design

- 3.1 Project objectives in details
- 3.2 Architectural design
- 3.3 Design options
- 3.4 AWCMA proposed algorithm
- 3.5 System model

Chapter Three

Conceptual Design

- 3.1 Project objectives in details**
- 3.2 Architectural design**
- 3.3 Design options**
- 3.4 AWCMA proposed algorithm**
- 3.5 System model**

Chapter Three

Conceptual Design

This chapter involves the issues related with the conceptual design of the project, brows the architectural design, the system modeling in addition to go deep in project objectives.

3.1. project objectives in details

- To design and implement multi-agent system that does the crawler responsibilities in an efficient way.

Our goal in this system is to use the technology of JADE agents to do the responsibilities of web crawlers, in other words the mobile agent will be sent to the web servers, select the pages, parse and analyze the pages, monitor the changes and updates, and return the result messages (keywords, Titles and URLs) to the search engine side to be manipulated and saved to the database.

- To overcome some of the drawbacks of current web crawlers related with performance and bandwidth utilization.

During our search on web crawlers we noted that they suffer from many drawbacks as specially as issues related with the network over which they work, and some problems in updating the web search engines' indexes in approximate real time.

These drawbacks were put in mind to be solved; multi-agent technology is suggested trying to overcome some of them.

- To keep the search engine's index updated in an approximate real time without much overload on the network.

This could be achieved by employing the crawling agent as a monitoring agent , with time ticker depending on the configurations done at crawler manager, the light weight of the message and transferring only the changed contents are two main factors to save the bandwidth over the network.

3.2. Architectural Design

3.2.1 exploration of the system (definition)

AWCMA (Autonomous Web Crawler using Multi Agents) takes the advantage of mobile agents to migrate to web servers over Internet and make all the processes related to any traditional crawler at the side of web server (filter, extract keywords and URLs, compress) then returns the results back to the crawler manager of the specified search engine ,and keep a monitoring agent so that to detect any changes or updates either by time, after that the crawler manger integrate the payload of mobile crawlers into the database of the search engine to be queried later. The figure 3.1 below demonstrates the general idea of the system.

The first strategy adopts the parallelism in crawling by sending agent\agents to the specified web servers, retrieves their URLs, keywords and sending them via messages in parallel, the crawling agent also takes the rule of monitoring these web servers for updates and changes by either time or event triggers and sends these updates to the home to be integrated into the search engine database.

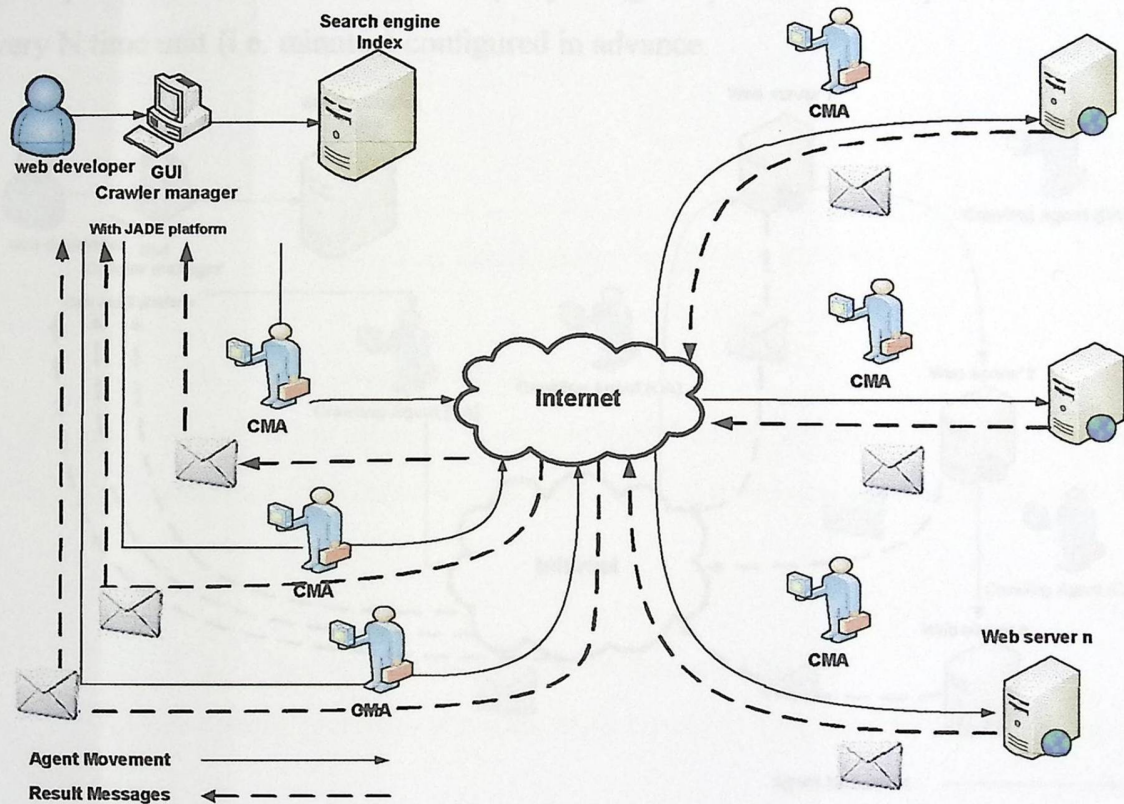


Figure 3.2. The first strategy of AWCMA (Parallel)

The second strategy adopts the serial paradigm to carryout crawling, by sending agent\agents to a seed web server and processes its contents then transfers to another server till the last one.

Taking in consideration that the agent doesn't carry the information retrieved, rather the contents are also sent via messaging technology to the crawler manager.

The updates are taken in mind here by repeating the process to the specified web servers every N time unit (i.e. minutes) configured in advance.

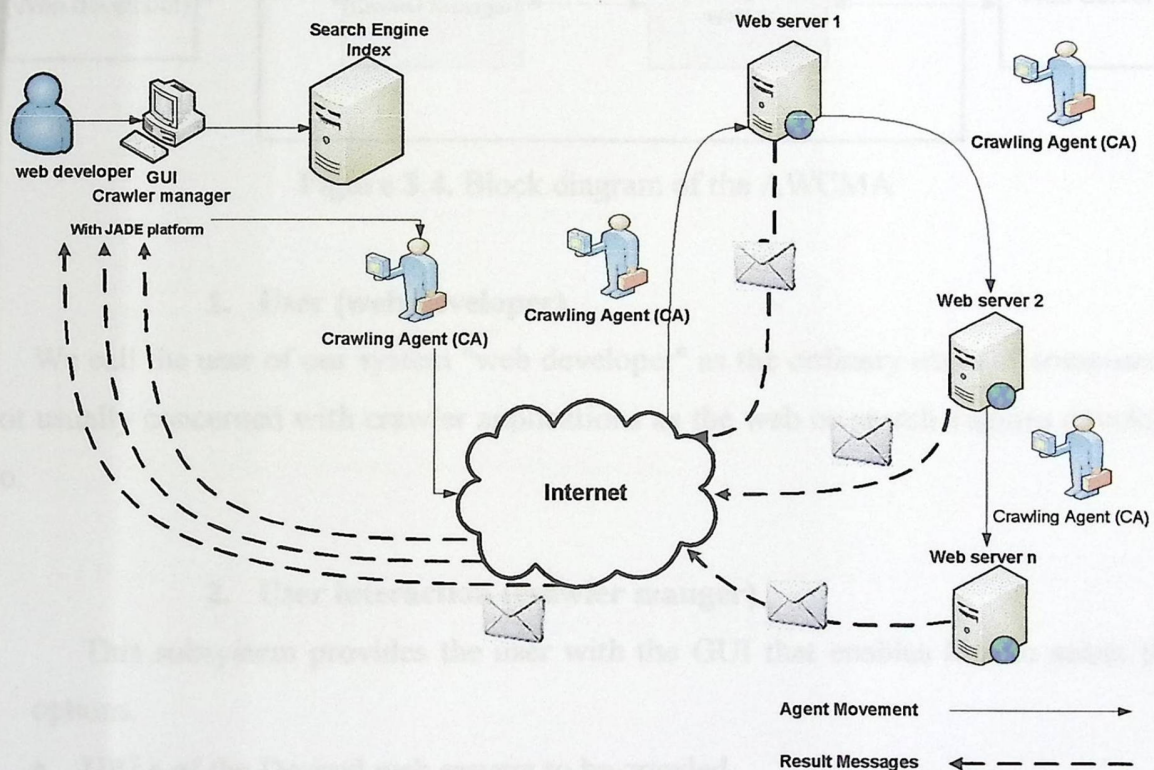


Figure 3.3. The second strategy of AWCMA (Serial)

3.2.2. System block diagram

As shown in figure 3.2 the block diagram of the AWCMA system which can be explained in this following sub sections.

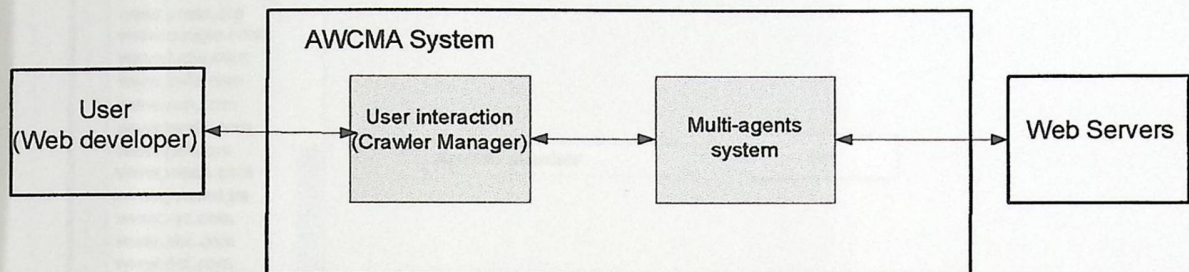


Figure 3.4. Block diagram of the AWCMA

1. User (web developer)

We call the user of our system "web developer" as the ordinary users of computer are not usually concerned with crawler applications as the web or search engines developers do.

2. User interaction (crawler manger)

This subsystem provides the user with the GUI that enables him to select these options.

- URLs of the Desired web servers to be crawled.
- The time sleeps of the two successive crawling operations to monitor the updates and changes of the web server.
- The maximum memory utilization allowed for a specific web server.
- The user will receive a performance report after (Performance Tester Agent) PTA did his work.
- The user will notice the end of crawling of each web server by receiving the results (URLs, keywords) to be stored in the database.

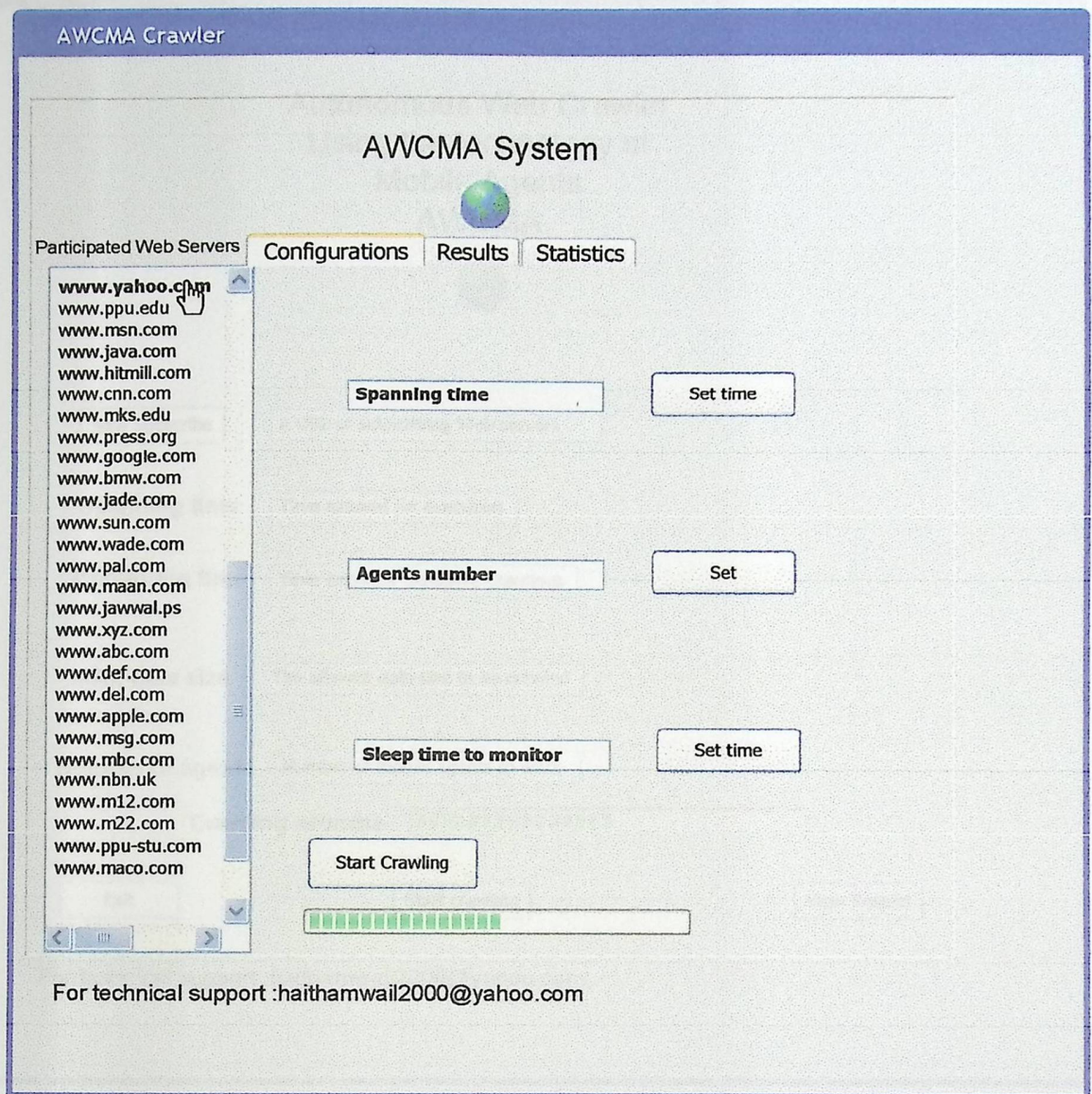


Figure 3.5. User interface of the AWCMA- prototype 1

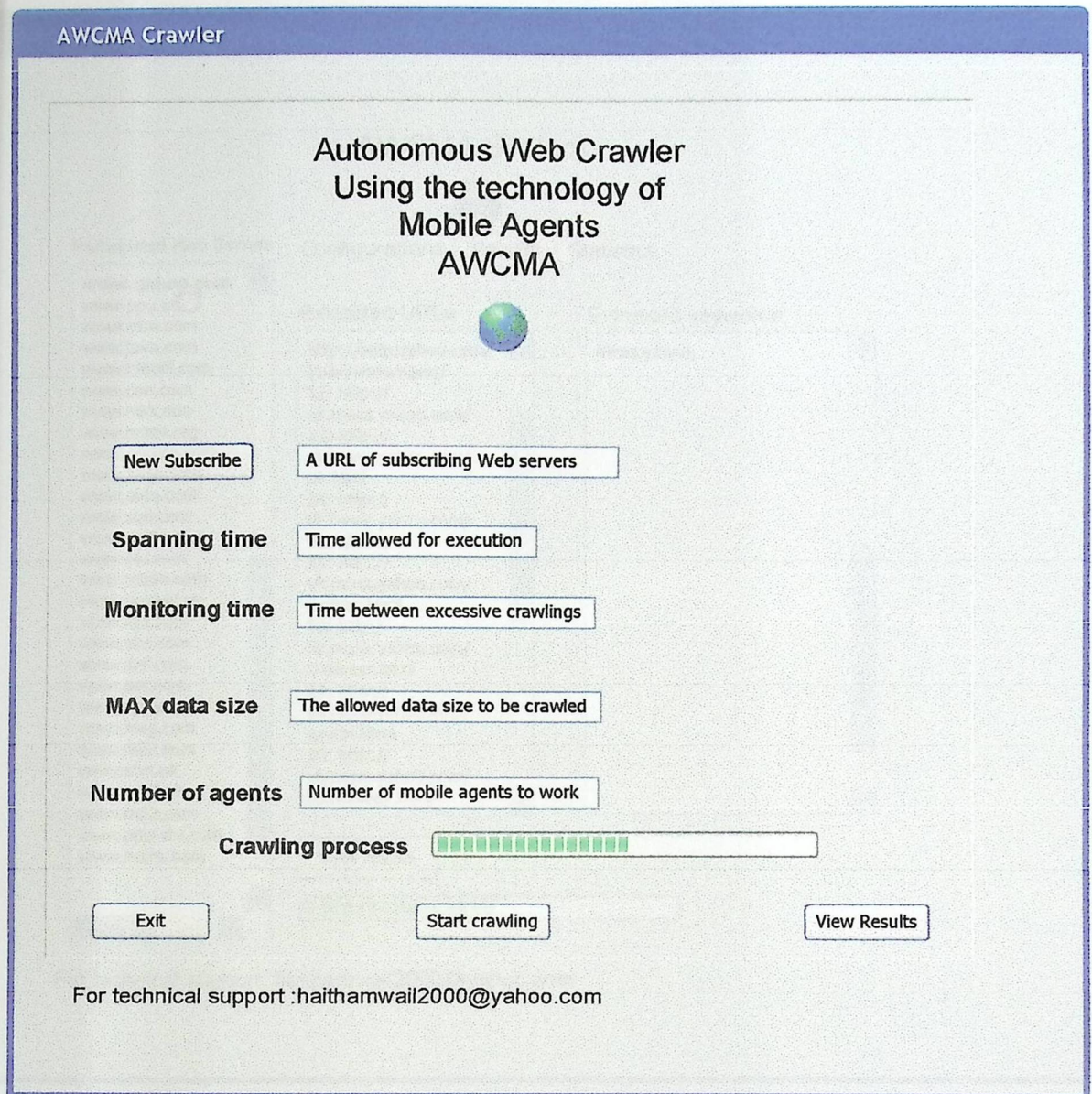


Figure 3.6. User interface of the AWCMA- prototype 2

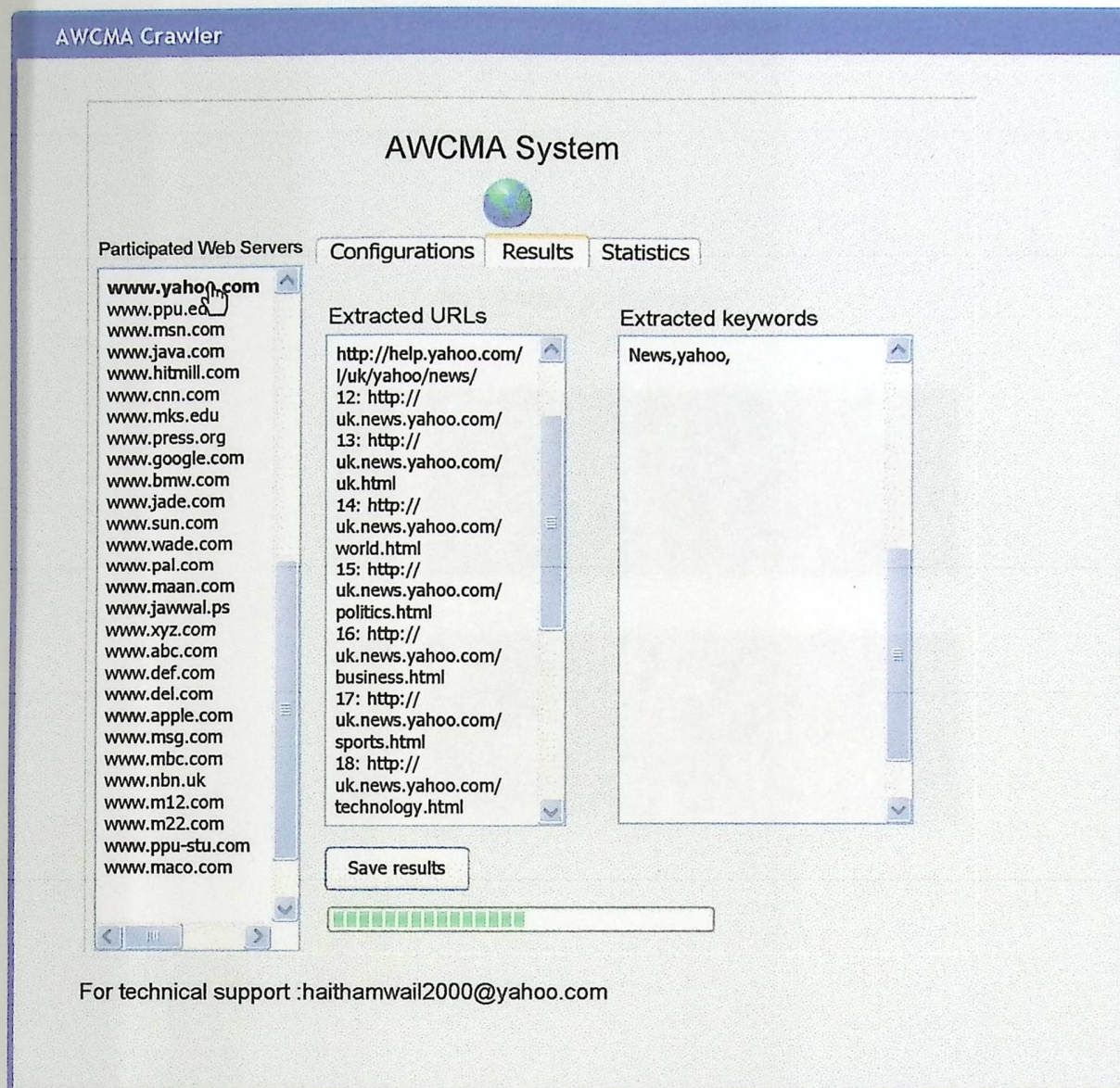


Figure 3.7. Results of AWCMA- prototype 3

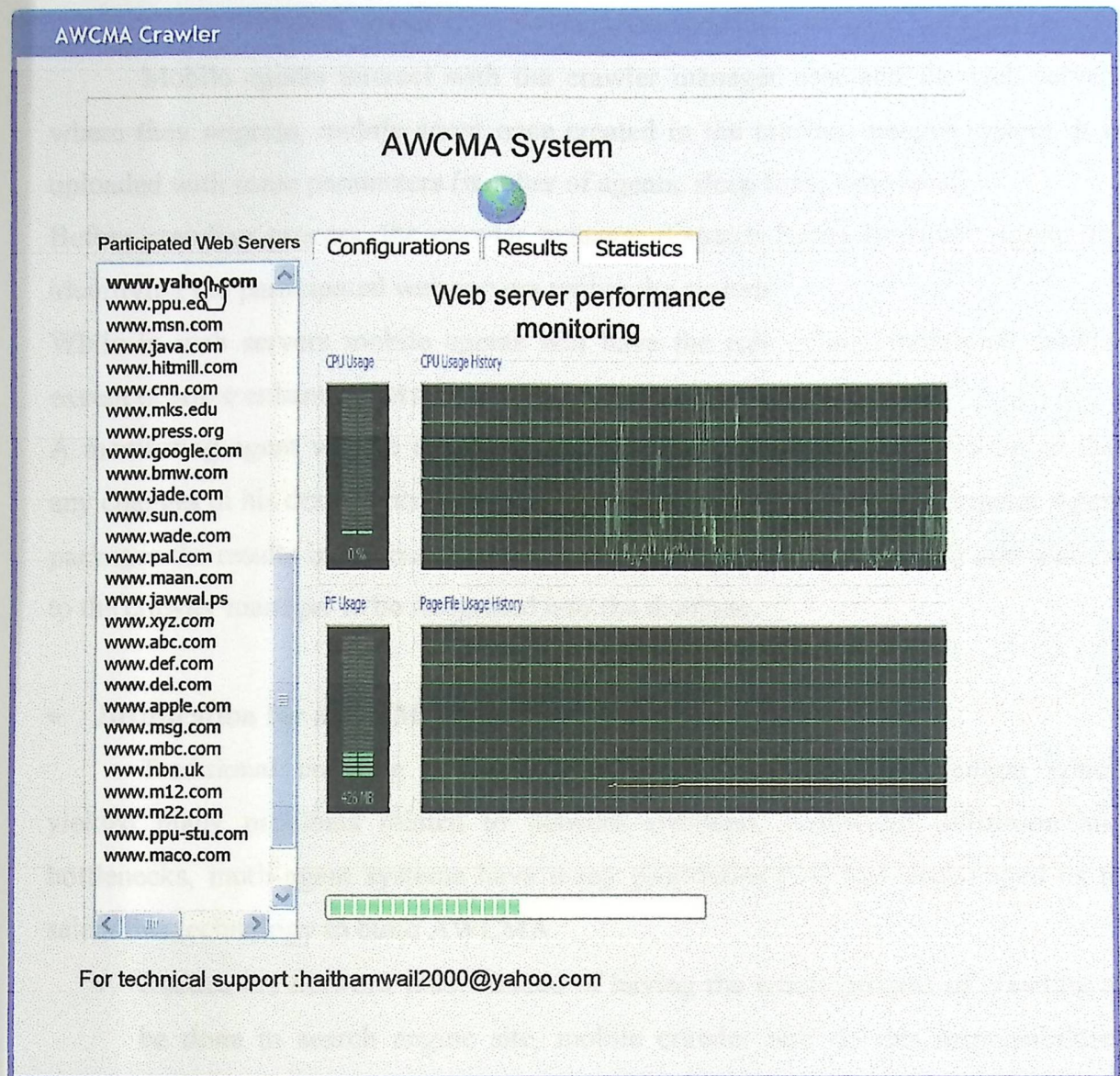


Figure 3.8. Results of AWCMA- prototype 4

AWCMA provides the user to interact with mobile agent for each participating web server (web site) through:

- Creating Name Resolver Agent (NRA)
- Creating Crawling and Monitoring Agent (CMNA).
- Supply the CMNA with monitoring time of crawling for each web server.

3. Mobile agent

Mobile agents interact with the crawler manager user and the web servers where they migrate, mobile agent once created in the crawler manger system, it is uploaded with some parameters (number of agents, sleep time, time span).

Before crawling process, the crawler manager dispatch Name Resolver Agents for identifying the participated web servers within the system.

While in web servers mobile agents will have the rule of any traditional crawler except of some enhancements that will be discussed later.

A monitoring agent will be employed depending on participant's restriction so that any changes in his documents will be noticed immediately. Finally the Crawler Agent packages the results in a message envelope (Crawling Payload Message) and sends it to the crawler manager to be integrated into the database.

- **Justification for using Mobile agent**

Traditional crawling systems generally use client-server paradigm which yielded many problems related to network overload, bandwidth utilization and bottlenecks, mutli-agent systems have many proprieties [24] that encouraged us to select this technology to build AWCMA.

- Reduce the network load: instead of having the whole process of crawling to be done in search engine site, mobile crawler will do this responsibilities locally in web servers, find the results and send them to the crawler manager.
- They overcoming network latency: that will enable a scalable web crawler with reasonable time.
- They adapt dynamically: this will serve in changes detection in web servers.
- They are robust and fault-tolerant: web crawlers are generally exposed to many failures (i.e. platform, web server, DNS).

4. Web servers

This module represents the ordinary web servers which contain the whole pages and documents, in AWCMA each web server has to have a platform for hosting mobile agents that would be supplied after subscription.

An important point that should be noticed here that the crawling process would be done locally at these web servers.

3.3. Design options

- Mobility of agents: this point shows two clear choices, to have stationary agents in each participating web server that do the crawling process and notify any changes, send them to the crawler manager through messages. Or to have mobile agents that migrates to the participating web servers and do the crawling processes locally, create a monitoring agent and return the results back to the crawler manager also by messages.
- Mobile agent platforms: we have three choices to adapt the most convenient platform for creating, marinating, dispatching and killing the agents. These alternatives are:
 - Aglets software development kit (ASDK): which we found through searching [30] that it is a good way for crawling thanks to its 1. Supporting the "data push" model which reduces the network latency and notice rapidly the changes in web documents 2. The crawling important issue is the graph- traversal problem- the web can be seen as a graph and there is many alternative paths for nodes with different cost estimation- aglets has the ability to clone, to have an initial itinerary and react to local conditions that permits numerous traversal strategies to be explored, but unfortunately no new versions have been released, no more complete tutorials and materials available, beside the low activities related with this platform nowadays.
 - JADE: it is also one of promising platforms for developing agents, the encouraging points that lead us to choose this platform is the

following: 1) the fertile soil of updated resources and releases with a very high level of activity, its support to web services so you can easily manipulate your agents from any appropriate web application [31], in addition to the measurements that show the good performance of JADE in message system and its ability to be candidate for developing heavy load distributed systems [32]

- Number of agents:
 - The initial imagination was to employ only a single agent for each participating web server to handle crawling process.
 - The later imagination was to employ multi-agents for each participating web server specified by its administrator; this will affect the money paid for subscription.
 - The appropriate choice was to separate the use of agents number to the strategy adopted.
- Crawling paradigm
 - Distributed paradigm that distributes the whole crawling process over multi-agents employed. In fact no fully distributed paradigm is presented because of crawler manager still centralized .

3.4. AWCMA proposed algorithm

Input : 1. The URLs of the participated web servers
2. Maximum memory usage allowed
4. Monitoring time
5. Crawling strategy.

Output: 1. performance report.
2. Keywords
3. New URLs

Symbols : URLs within URL box :B

The crawler manager :CM

Performance Tester Agent :PTA

Participated Web Server : WS

Maximum memory usage: MU, CPU Usage : CU

Monitoring time :Mt

Crawler and monitoring agent: CMNA

Time unit : ζ

Maximum memory and CPU utilization :x ,y .

Start time of operation: t.

Process :

If (a new participant in AWCMA)

While (B is not NULL)

{

CM do

- Specify MU and CU
- Dispatch PTA for the specified web server
- Creates a monitoring and crawling agent..

PTA at WS do

- Check the performance.
- If(MU < x && CU < y) {
Returns with safe mode.
}

At CM again // safe mode

Re Choose the strategy and monitoring time

If (strategy== serial)

Allow created CMNA to be dispatched to the first web server

While (t==Mt){

While (WS< number of web servers){

.DispatchTo (WS)

WS=WS+1


```

t=t+ζ
Return messages
}
Else{ // parallel
For each WS selected
DispatchTo(WS)
While (t==Mt){
Return messages
t=t+ζ
}
Dispose (CMNA)
}
Else {
Switch on coocrawlers.
}
}

```

Figure 3.9: Data flow diagram- Level 0

3.5. System Modeling

3.5.1. Data flow diagrams

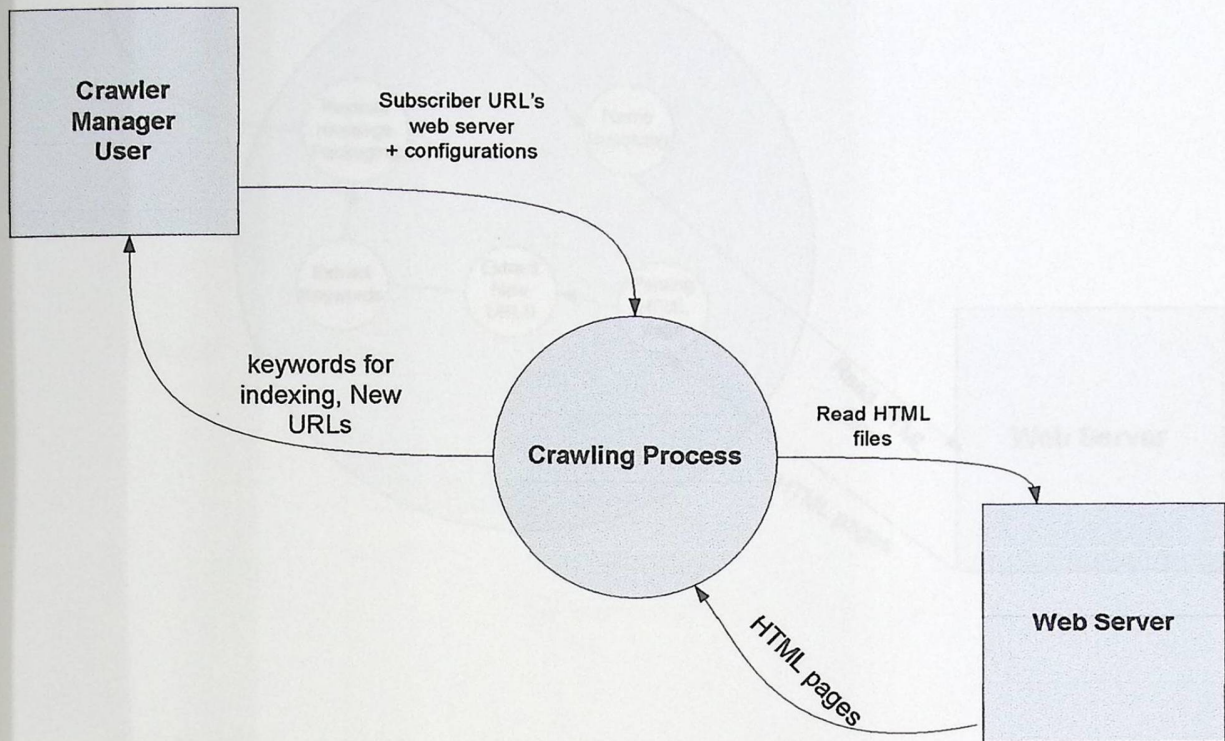


Figure 3.9. Data flow diagram- Level 0

Figure 3.9 shows the Data flow diagram which describes the data flow within the system, the core processes which is shown in the circle is the crawling process which indicates the whole process and how it interacts with the system entities.

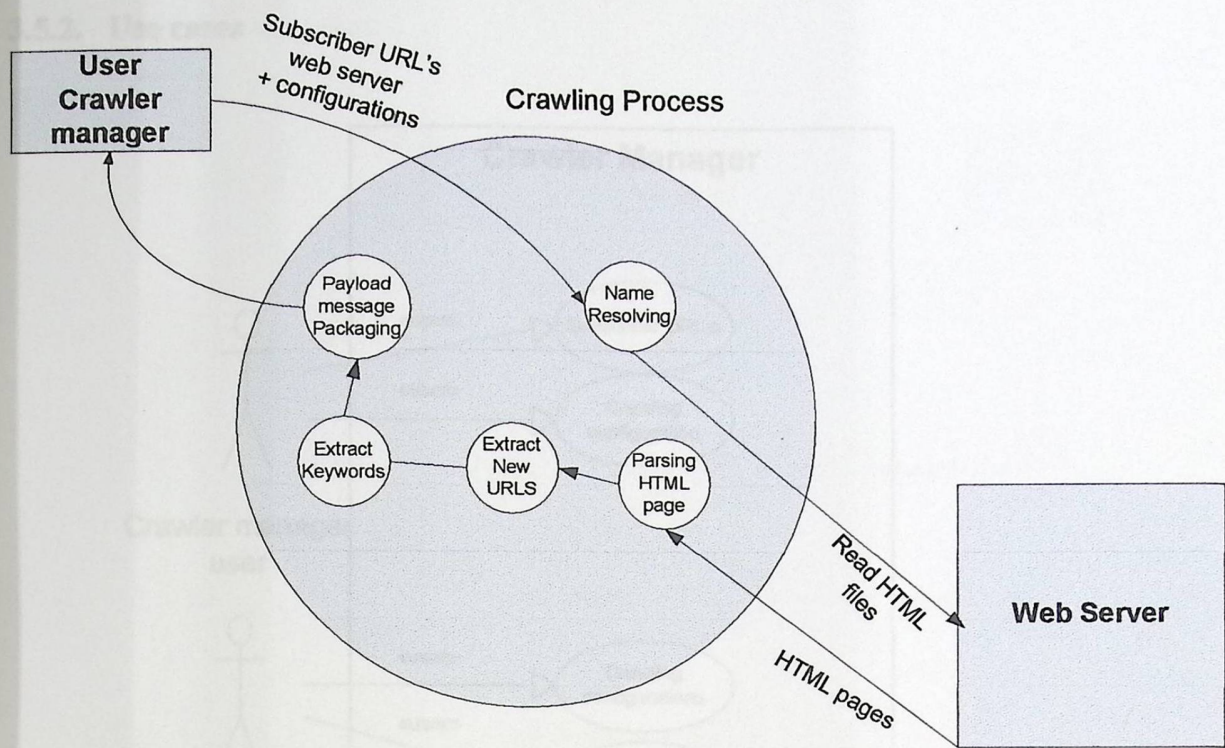
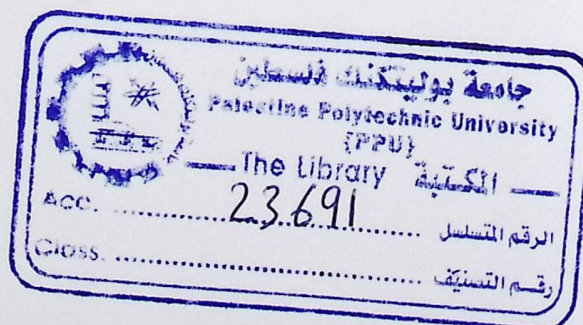


Figure 3.10 Data flow diagram- Level 1

Figure 3.10 show refined data flow – level one which gives details for the crawling process to explicitly describe each sub process .



3.5.2. Use cases

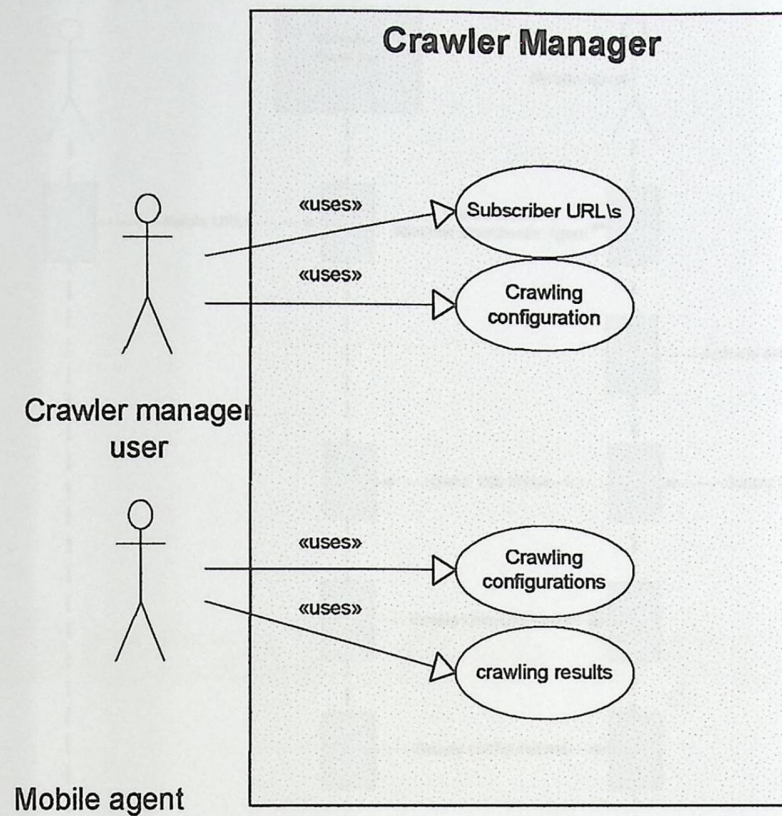


Figure 3.11 Use case at crawler manager

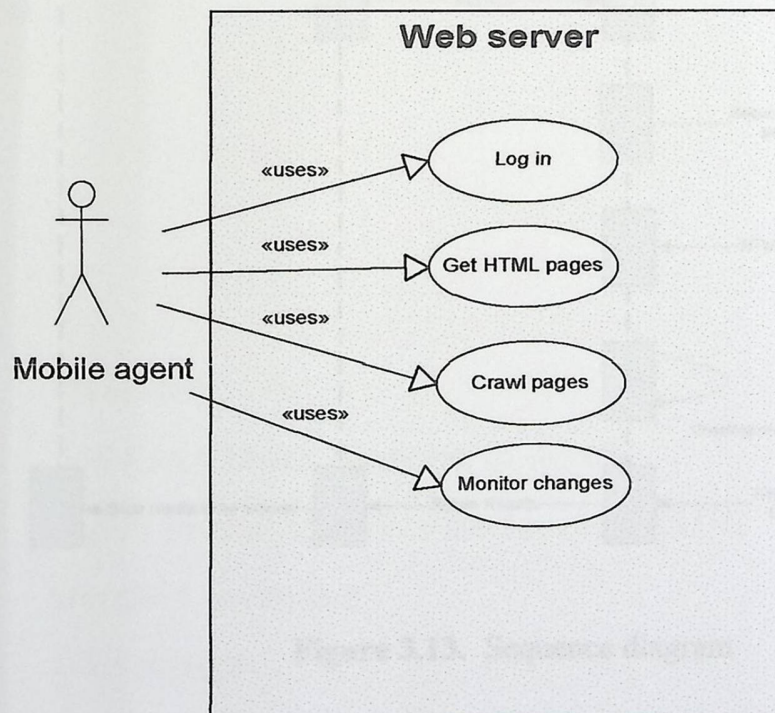


Figure 3.12. Use case at web server

3.5.3. Sequence diagram

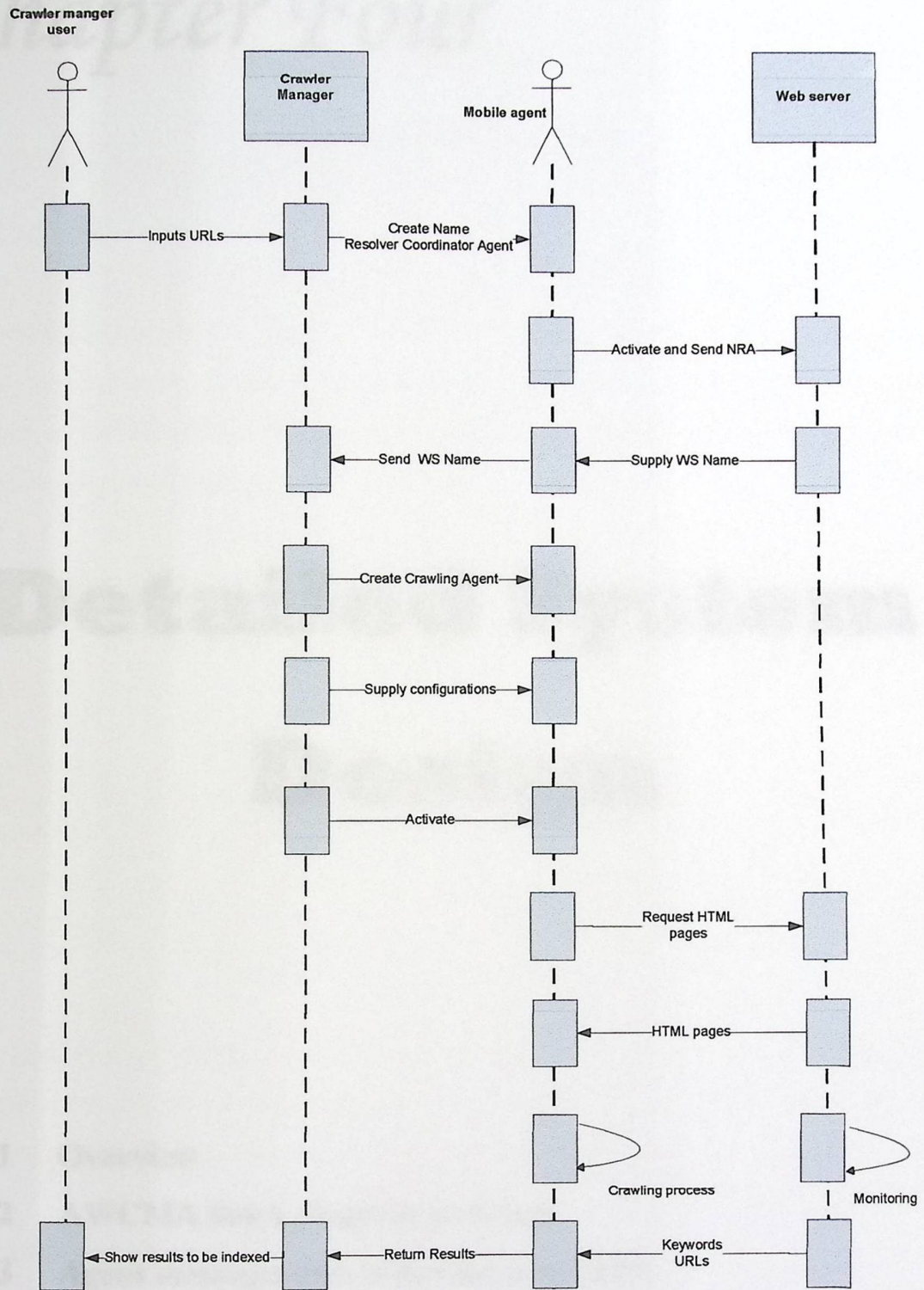


Figure 3.13. Sequence diagram

Chapter Four

Detailed System Design

4.1. Overview

In this chapter we are going to describe the whole design for the system, describe the project components in detail, the interface, and the flow charts for the system also we introduce in the last section the proposed design improvements to the system which were utilized due to some challenges will be discussed in details.

Detailed System Design

Design

Figure 4.1. Block diagram of the AWCMA

4.1.1. User (web developer): the user of the system is generally a web developer who is responsible for giving the requested configurations from participated web servers in the

4.1.2. Administrator: an administrator within a search engine

4.2 AWCMA block diagram in details

4.3 Agent messages and behaviours in JADE

4.4 Detailed modeling

4.5 GUI design for selecting the used URL(s) for the participated web

4.6 Challenges and improvements

Chapter Four

Detailed System Design

4.1. Overview

In this chapter we are going to describe the whole design for the system, describe the project components in detail, the interface, and the flow charts for the system also we introduce in the last section the proposed design improvements to the system which were initiated due to some challenges will be discussed in details.

4.2. AWCMA Block diagram in details

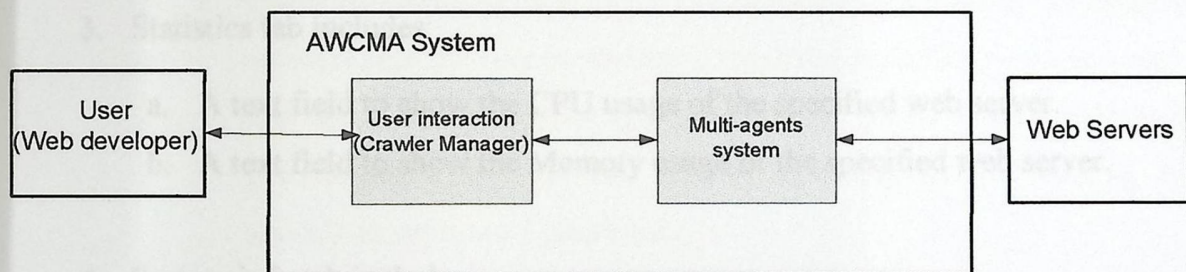


Figure 4.1. Block diagram of the AWCMA

4.2.1. User (web developer): the user of the system is generally a web developer who is responsible for giving the requested configurations from participated web servers in the system; mostly he will be an administrator within a search engine

4.2.2 User interaction (Crawler Manger): which consists of:

4.2.2.1 GUI (Graphical User Interface) unit :

1. Configuration tab includes:

- a. A combo box for selecting the seed URL(s) for the participated web server(s).

- b. A radio button group for selecting the strategy plan of the crawling mission (i.e. serial, parallel).
- c. A text box for entering the sleep time between two successive crawling process assured by a set time button.
- d. A text box for entering the maximum memory utilization allowed for a specified web server.
- e. A text box for displaying the crawling completion time.
- f. A button to start crawling.

2. Results tab includes:

- a. Two text areas to display both the extracted and absolute URLs with indicating labels.
- b. Another text area to display the titles and keywords of the web servers' pages.

3. Statistics tab includes:

- a. A text field to show the CPU usage of the specified web server.
- b. A text field to show the Memory usage of the specified web server.

4. System info tab includes:

- a. The link for downloading the JADE packages.
- b. The user manual which demonstrates the right installation and use of system.
- c. The ethics of use see Appendix A.

4.2.3 Multi-agent system consists of the following:

1. Crawler manger agents(home agents):

- a. Name Resolver Coordinator Agent (NRCA): this agent is responsible for creating and dispatching the Name Resolver Agents (NRA) according to the existence of participating web servers.

This agent uses the cyclic behavior to receive the names of the web servers read by the NRA, analyses the names and displays them within the Combo Box, it is valuable to say that this agent also interacts with agent management system to be aware of the existing web servers' containers.

- b. Crawler Manager Agent (CMA): this is responsible for receiving the data messages retrieved by the crawler agents and integrating these messages into the database, this agent uses the advantages of the cyclic behavior to be prepared for receiving the data messages cyclically whenever ready.
- c. Performance Receiver Agent (PRA): this agent acts as a receiver for the performance checking results examined by the Performance Tester Agent (PTA) which will be discussed later, then displays the results (CPU, Memory usage) within the GUI.

2. Mobile agents

- a. Name Resolver Agent (NRA): this mobile agent transfers to each web server, reads the root name there, sends the name (i.e. www.ppu.edu) to the NRCA to be displayed, the number of NRAs to be dispatched is predetermined through the NRCA according to the existing web servers as mentioned previously.
- b. The Crawling and Monitoring Agent (CMNA): which acts as the most important agent within the system that is responsible for doing the whole crawling process by extracting the relative and absolute URLs and extracting the page keywords and description from the following important places:
 - i. The URLs are extracted from the `<href="http\THE URL.EXTENSION">` tag.
 - ii. The header title of the pages is extracted from `<title>THE TITLE OF THE PAGE </title>` tag.
 - iii. The Meta keywords tag `<Meta content = "THE KEYWORDS ARE HERE" name=keywords>`

Figure 4.2. Web servers directory hierarchy

- iv. The Meta description tag `<Meta content= "THE DESCRIPTION IS HERE" name=description>`
- c. The Performance Tester Agent (PTA) : to get use of the Cocrawlers we have to test the main resources of the web servers indented to be crawled (i.e. CPU and Memory) so that this agent uses the NSClient4j API and its library which allows a full access to the machine resources and retrieve their values continuously.

3. Web servers

The web servers entity is a crucial component from which the information is retrieved to feed the search engines, in AWCMA there are some constrains on these web servers to be participated in the system:

- a. The HTML files (i.e. the contents) are located in a folder named www in turn this is within the web server name folder (i.e. ppu), under the domain name (i.e. com, org...etc) folder and finally all are under the root folder. This hierarchy is shown in the figure below.

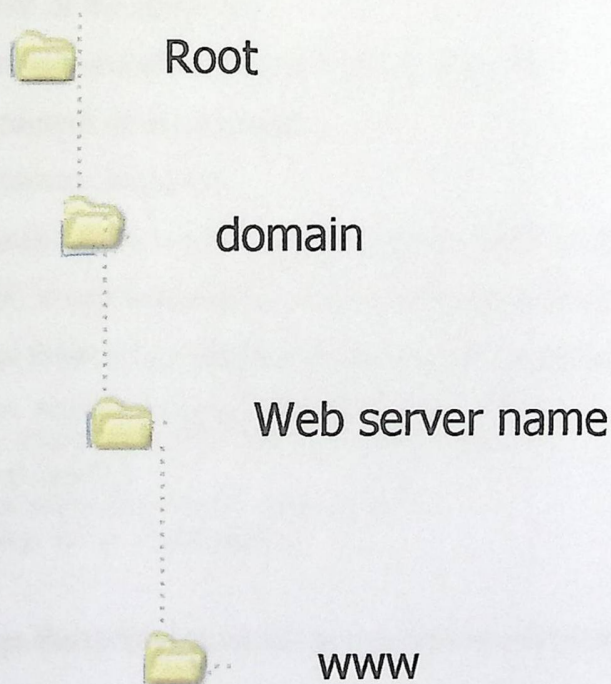


Figure 4.2. Web servers directory hierarchy

4.3. Agent messages and behaviours in JADE

4.3.1 Agent messages

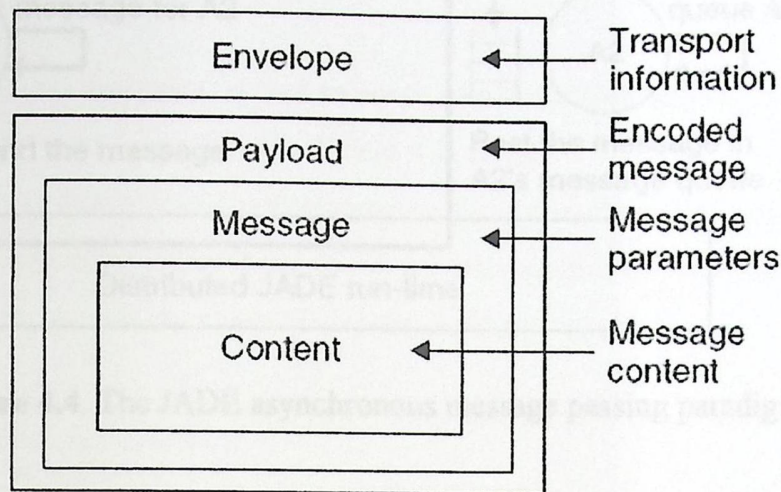


Figure 4.3. Message structure in JADE [15]

The figure above show the basic structure of Agent messages which is used for agent communication either within the same AP (Agent Platform) or different APs, the communication paradigm is based on asynchronous message passing.

The agent messages consist of these basic components:

- b. The sender of the message.
- c. The list of the receivers.
- d. The communicative act (i.e. inform, request).
- e. The content of the message.
- f. The content language.
- g. The ontology (i.e. vocabulary of the symbols used)

The message in JADE is implemented as objects of the `jade.lang.acl.ACLMessage` class to see what goes behind its implementation see this example.

```
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID("Peter", AID.ISLOCALNAME));
msg.setLanguage("English");
msg.setOntology("Weather-forecast-ontology");
msg.setContent("Today it's raining");
send(msg);
```

after sending the message the receiving agent has to receive and process the message as following:

```
ACLMessage msg = receive();
if (msg != null) {
    // Process the message
```

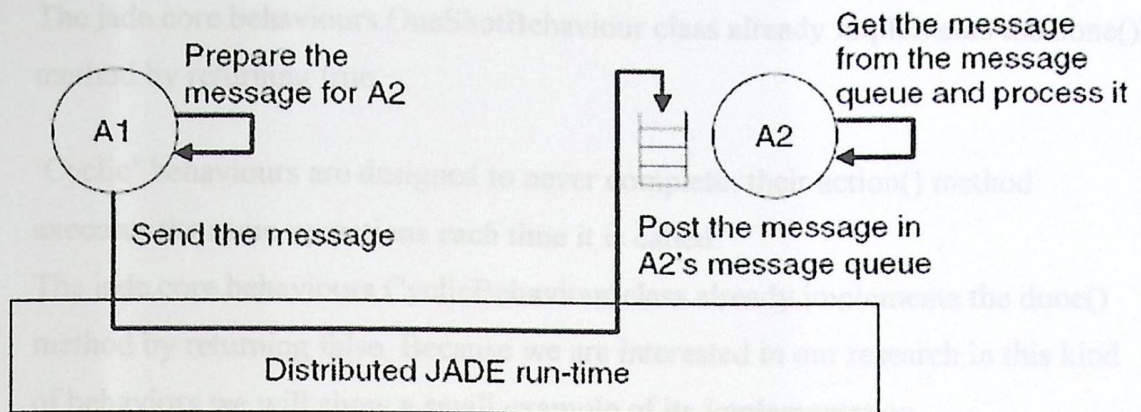



Figure 4.4. The JADE asynchronous message passing paradigm [12]

4.3.2 Primary Agent behaviors

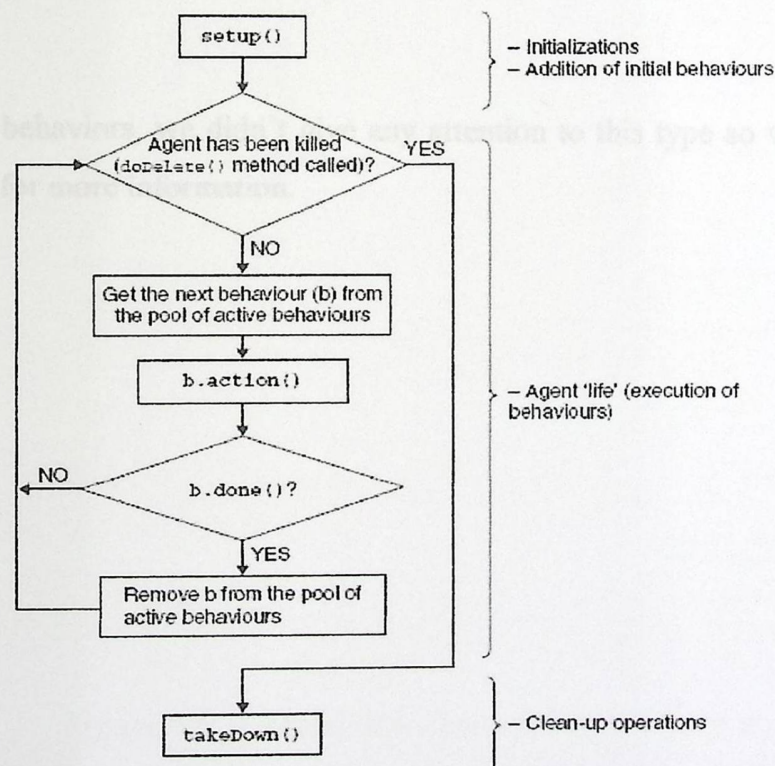


Figure 4.5. Agent thread path of execution [12]

Figure 4.5 will save time and effort in explaining the whole technology of agent behaviors which is very crucial in agent science, rather in few words we can summarize the primary types of these behaviors:

- ‘One-shot’ behaviours are designed to complete in one execution phase; their `action()` method is thus executed only once.

The `jade.core.behaviours.OneShotBehaviour` class already implements the `done()` method by returning `true`.

- ‘Cyclic’ behaviours are designed to never complete; their `action()` method executes the same operations each time it is called.

The `jade.core.behaviours.CyclicBehaviour` class already implements the `done()` method by returning `false`. Because we are interested in our research in this kind of behaviors we will show a small example of its implementation.

```
public class MyCyclicBehaviour extends CyclicBehaviour {
    public void action() {
        // perform operation Y
    }
}
```

Here Y executes repeatedly until the agent that executing the behavior terminates or die.

- Generic behaviors, we didn’t give any attention to this type so we advise you to see [12] for more information.

4.4. Detailed modeling

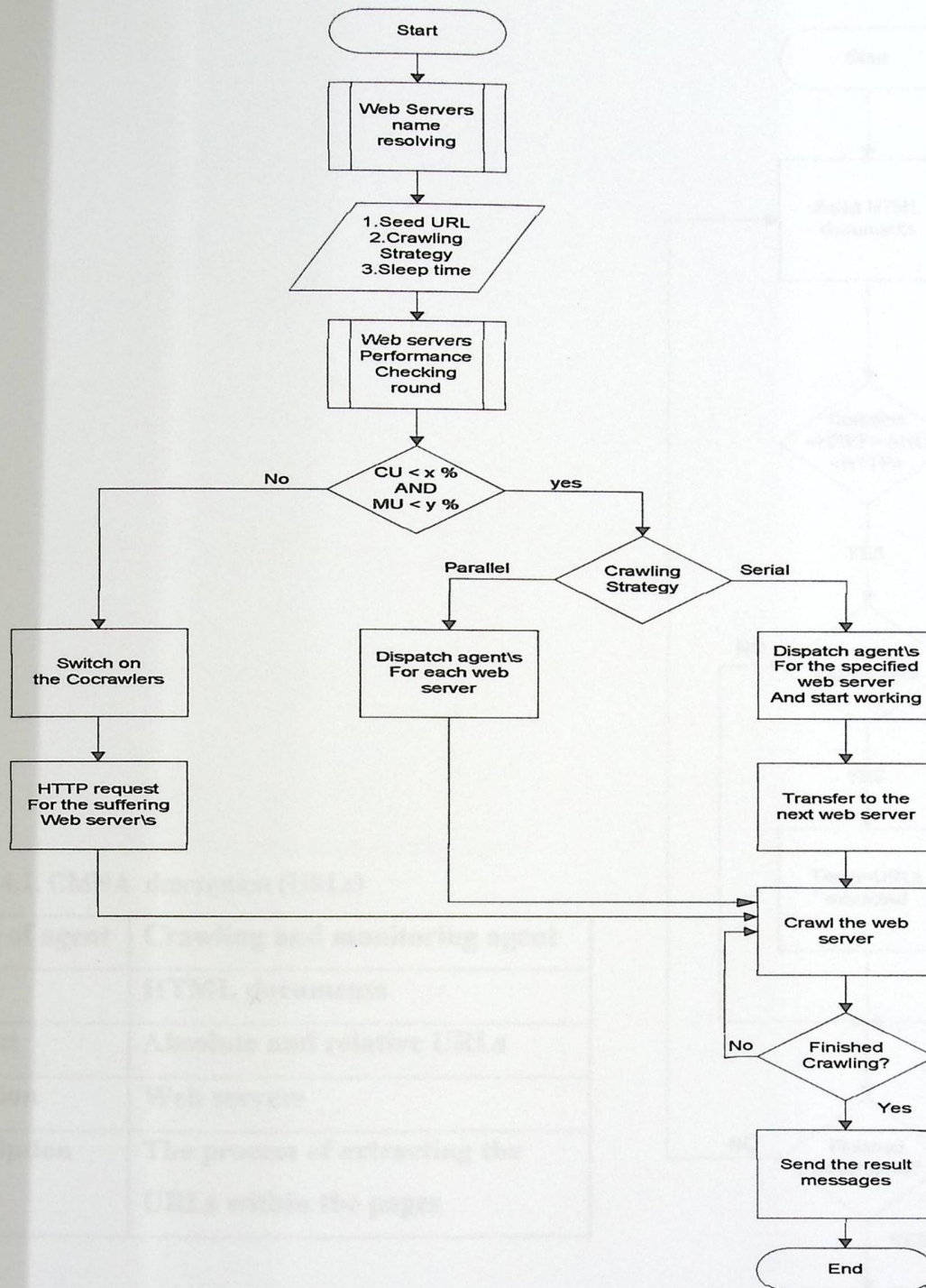


Figure 4.6. General flow chart of AWCMA system

Figure 4.6 describes the general algorithm of AWCMA and how the crawling process is done from the initial step of stating the application.

Table 4.1. CMNA description (URLs)

Name of agent	Crawling and monitoring agent
Input	HTML documents
Output	Absolute and relative URLs
Location	Web servers
Description	The process of extracting the URLs within the pages

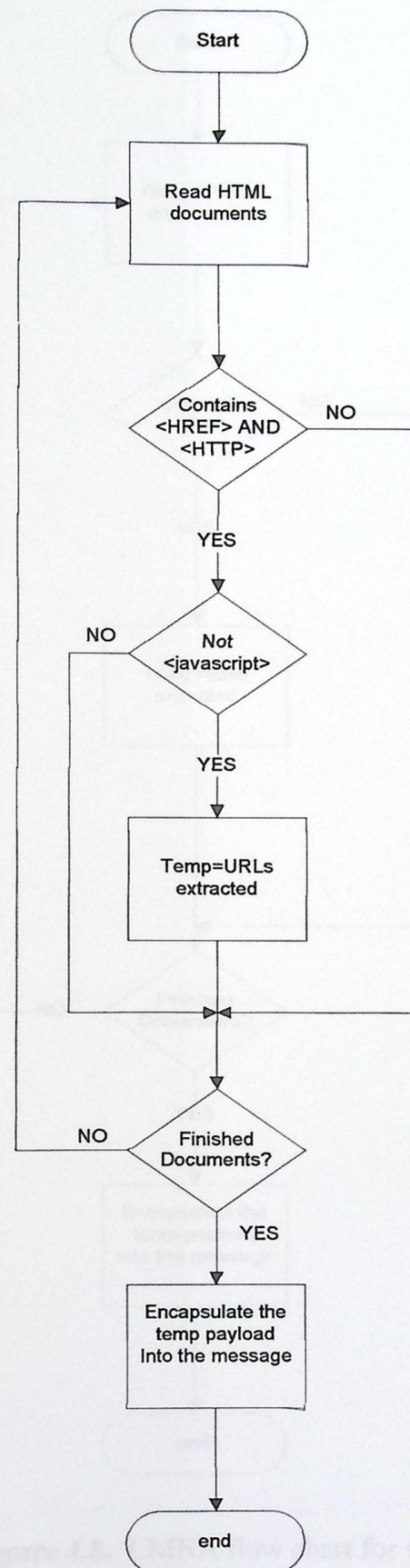


Figure 4.7. CMNA flow chart for URLs

Table 4.2. CMNA (Titles)

Name of agent	Crawling and monitoring agent
Input	HTML documents
Output	Web pages Titles
Location	Web servers
Description	The process of extracting the Titles of the pages

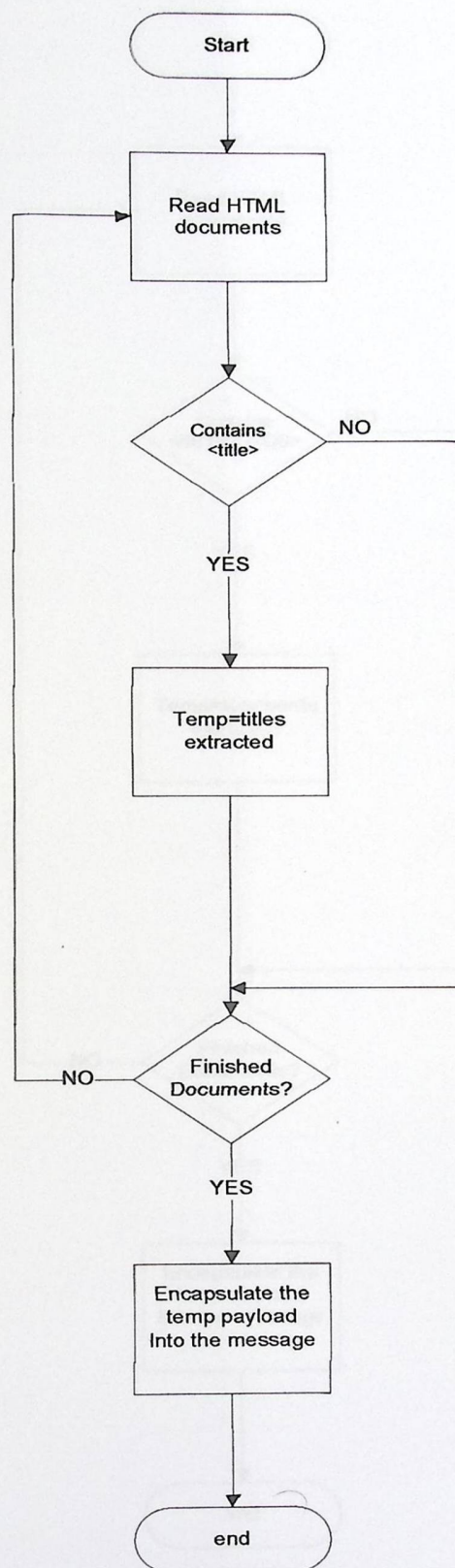


Figure 4.8. CMNA flow chart for web pages titles

Table 4.3. CMNA description (Keywords)

Name of agent	Crawling and monitoring agent
Input	HTML documents
Output	Web pages Keywords
Location	Web servers
Description	The process of extracting the Keywords in the pages

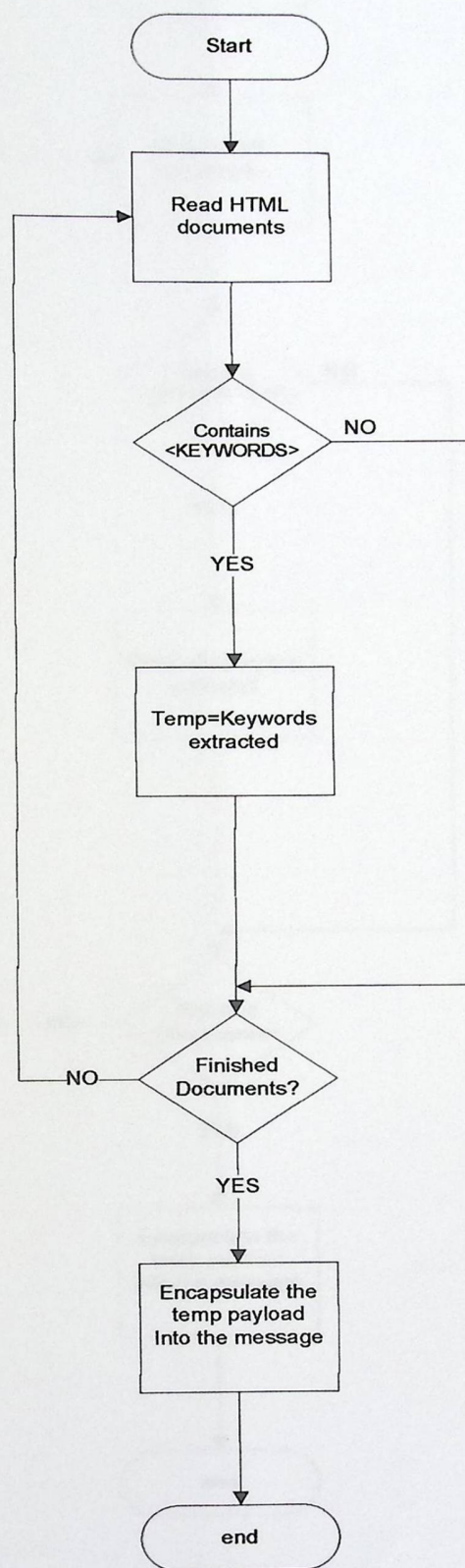


Figure 4.9. CMNA flow chart for web pages Keywords

Table 4.4. CMNA (Description)

Name of agent	Crawling and monitoring agent
Input	HTML documents
Output	Web pages Description
Location	Web servers
Description	The process of extracting the Description of the pages

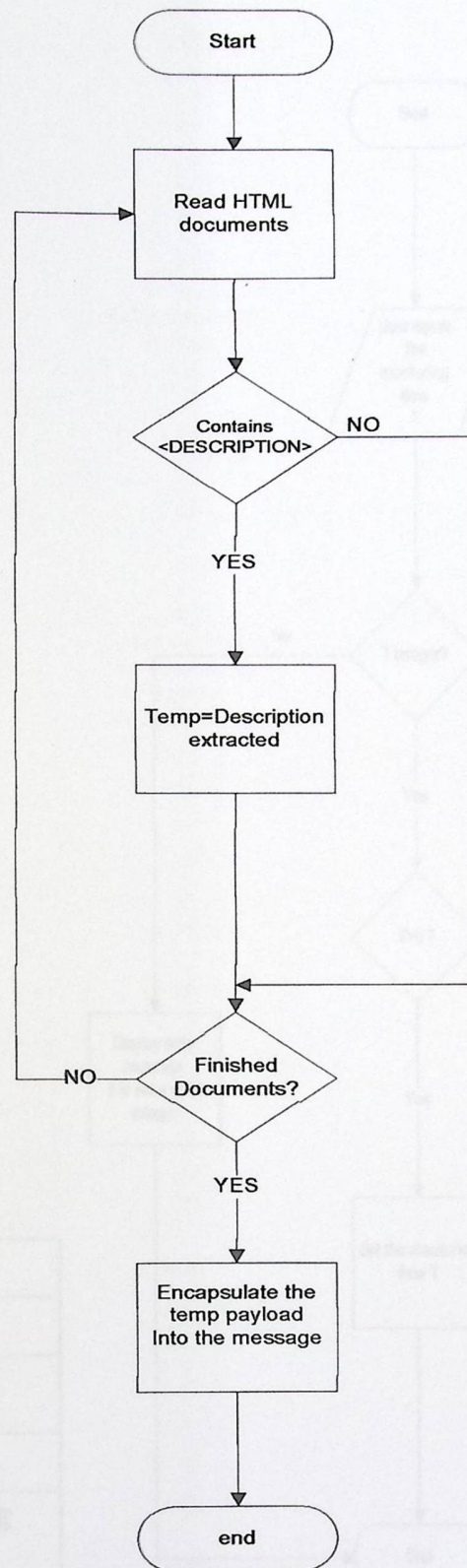


Figure 4.10. CMNA flow chart for web pages Description

Table 4.5. Monitoring time description

Name	Monitoring time set
Input	Monitoring time T
Output	The time between successive crawling.
Location	Crawler manager
Description	Repeat the process of crawling after T seconds

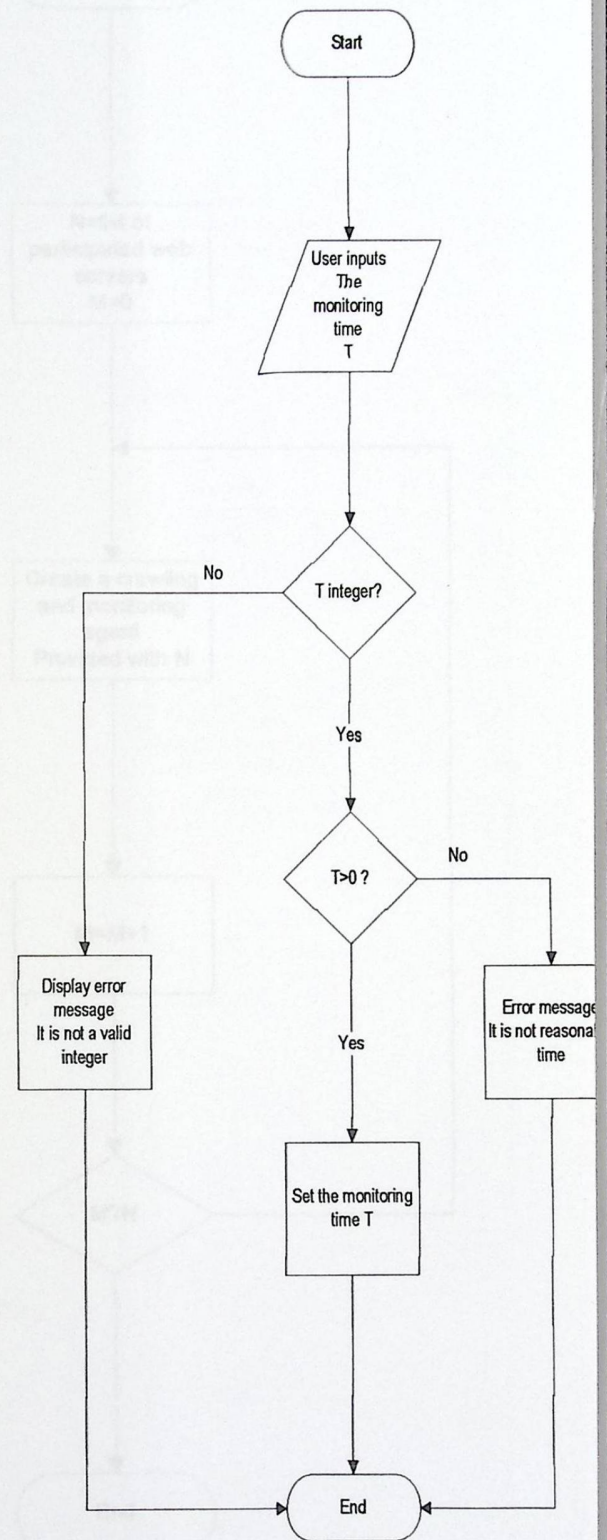


Figure 4.11. Monitoring time flow chart

Table 4.6. Serial crawling description

Name	Serial crawling strategy
Input	participated web servers
Output	Result messages
Location	From home to Web servers
Description	One agent sent for all web servers starting from seed URL

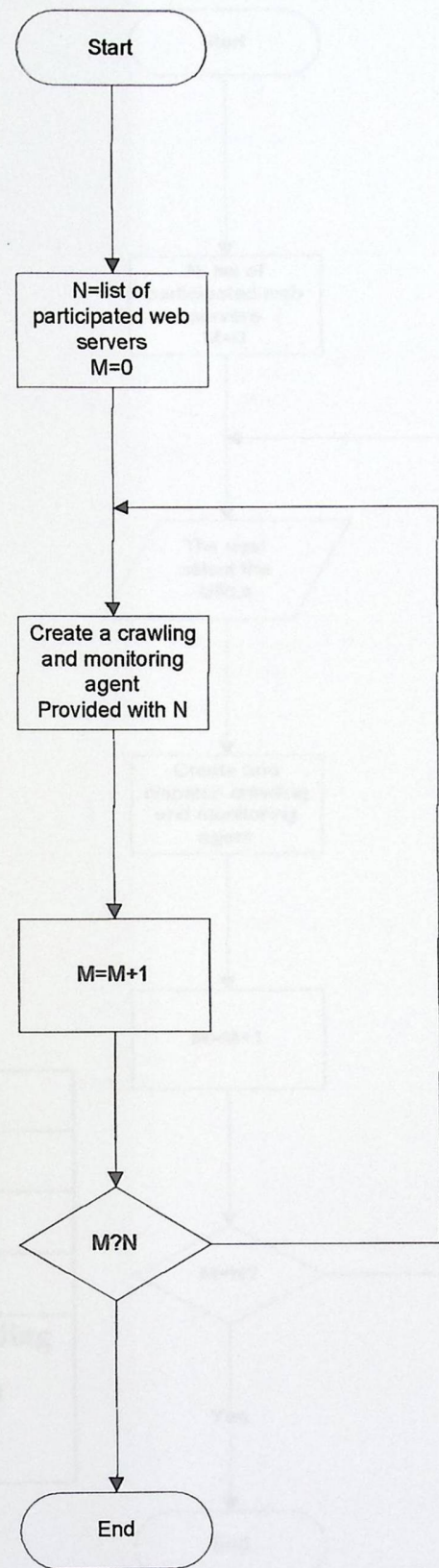


Figure 4.12. Serial crawling flow chart

Table 4.7. Parallel crawling description

Name	Parallel strategy
Input	Participated web servers
Output	Message results
Location	From home to web servers
Description	What we get is what we need , according to the user selection of N web servers dispatch N agents

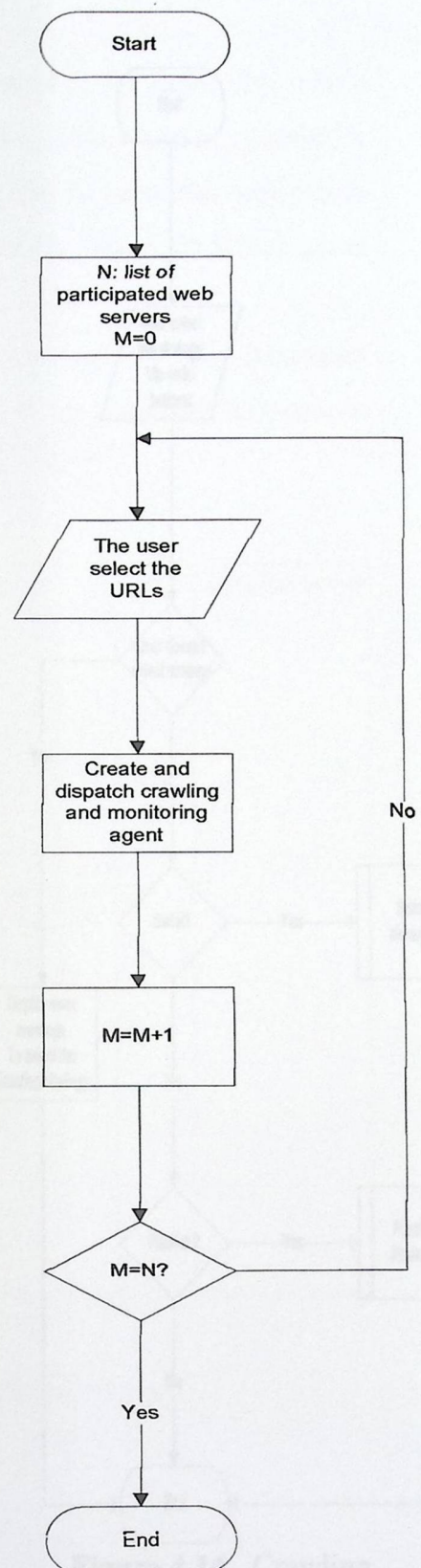


Figure 4.13. Parallel crawling flow chart

4.4.1. Crawling and monitoring agent

As we explained in section 4.2 that the CMNA is responsible for the whole process of crawling the web pages in servers and index their URLs and keywords, it can start after two requirements that should be available 1) the name of the destination web server supplied by the name resolver agent 2) the monitoring time which gives the time between two successive crawling.

After reaching the web server it starts extracting and return on the important information of the page then it leaves this web server and sends a message to the home (i.e. crawler manager).

Table 4.8. Crawling strategy description

Name	Selecting crawling strategy
Input	Crawling strategy
Output	Serial , Parallel paradigm
Location	Crawler manager ,GUI
Description	Allow the user to select serial or parallel

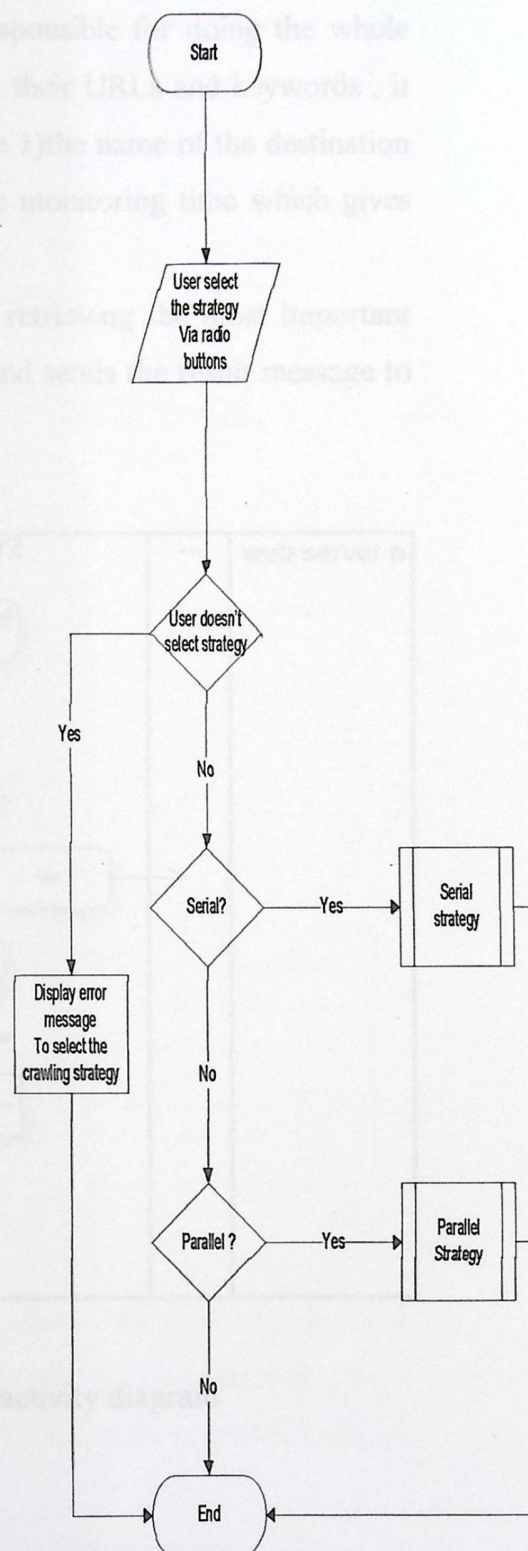


Figure 4.14. Crawling strategy flow chart

4.4.1. Crawling and monitoring agent

As we explained in section 4.2 that this CMNA is responsible for doing the whole process of crawling the web pages in servers and index their URLs and keywords , it can start after two requirements that should be available 1)the name of the destination web server supplied by the name resolver agent 2) the monitoring time which gives the time between two successive crawling.

After reaching the web server it starts extracting and retrieving the most important information of the page then it leaves this web server and sends the result message to the home (i.e. crawler manager).

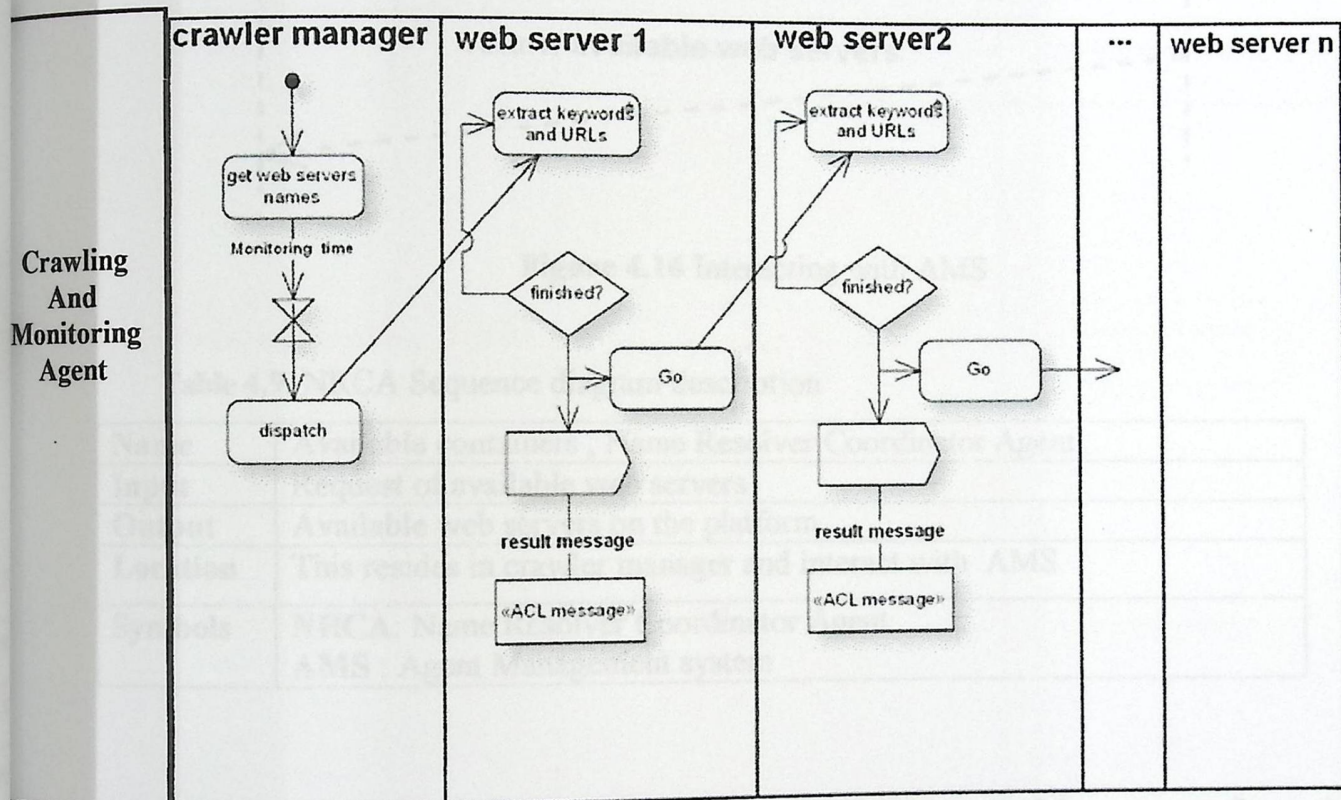


Figure 4.15 Crawling and monitoring agent activity diagram

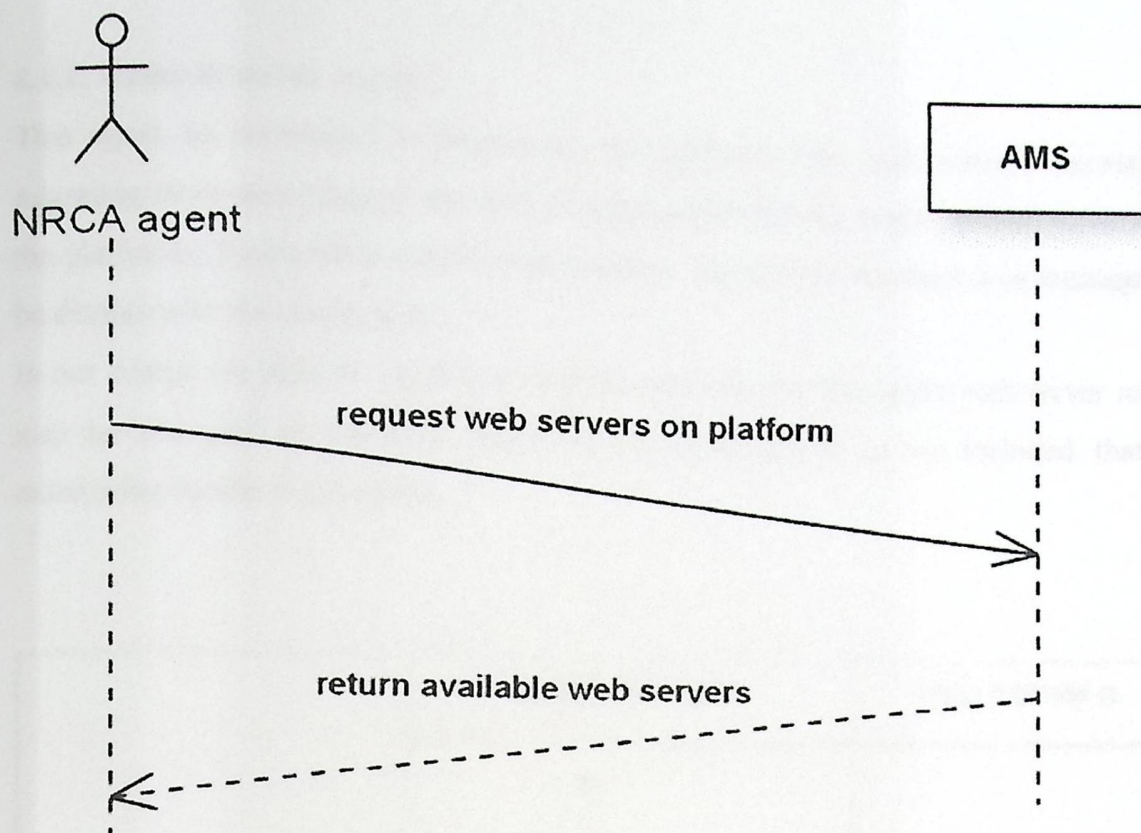


Figure 4.16 Interacting with AMS

Table 4.9. NRCA Sequence diagram description

Name	Available containers , Name Resolver Coordinator Agent
Input	Request of available web servers
Output	Available web servers on the platform.
Location	This resides in crawler manager and interact with AMS
Symbols	NRCA: Name Resolver Coordinator Agent AMS : Agent Management system

The system needs strongly to interact with AMS (Agent Management System) through the NRCA which send request to know the locations and containers (i.e. web servers) available on the platform (i.e. remote platform) and then migrates their to have the names of these web servers.

Fortunately FIPA specifications allow us to do that by implementing the AMS as an agent with capabilities of sending and receiving messages [15].

4.4.2. Name Resolver Agent

This agent as mentioned is responsible for gathering the web servers' names by migrating there according to the NRCA supplements for the available web servers on the platforms; finally NRA sends these names to the crawler manager as a message to be displayed in the combo box.

In our design we took an important issue in consideration that is the web server name may be changed or modified based on its administrator so we included that in monitoring beside the contents.

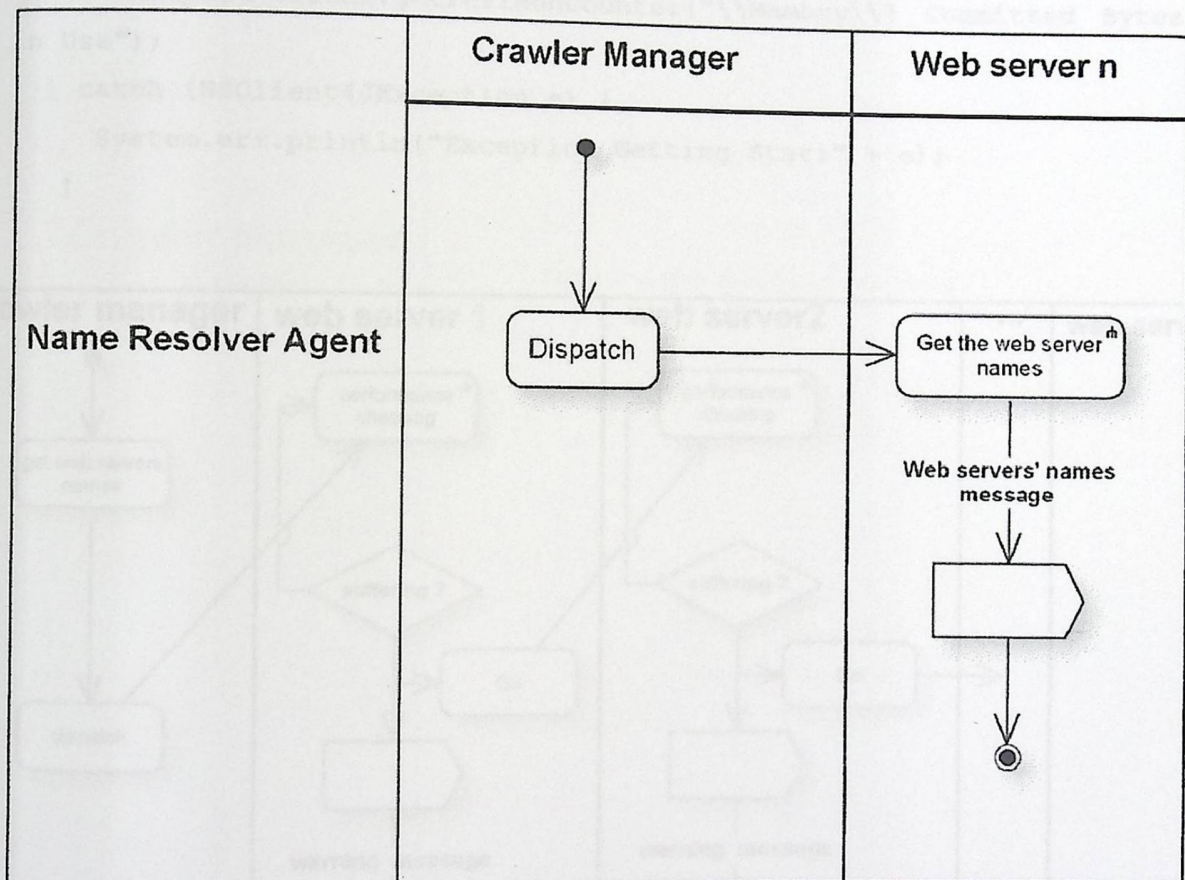


Figure 4.17 NRA activity diagram

4.4.3. Performance Tester Agent

As the user of the program clicked on the test performance button this agent is created to make a test round among all participated web servers supplied with NSClient4j library as its diagnostic tool which have the capacity to go through operating system resources (i.e. CPU, Memory, Hard Disk, ...etc) and then encapsulate the CPU and the Memory Usage at a specified time range. We will demonstrate here a small example of its implementation and powerful simplicity.

```
try {
    NSClient4j client = new NSClient4j("127.0.0.1");
    CReport=client.getPerfMonCounter("CPU");
    MReport=client.getPerfMonCounter("\\Memory\\% Committed Bytes
in Use");
} catch (NSClient4JException e) {
    System.err.println("Exception Getting Stat:" + e);
}
```

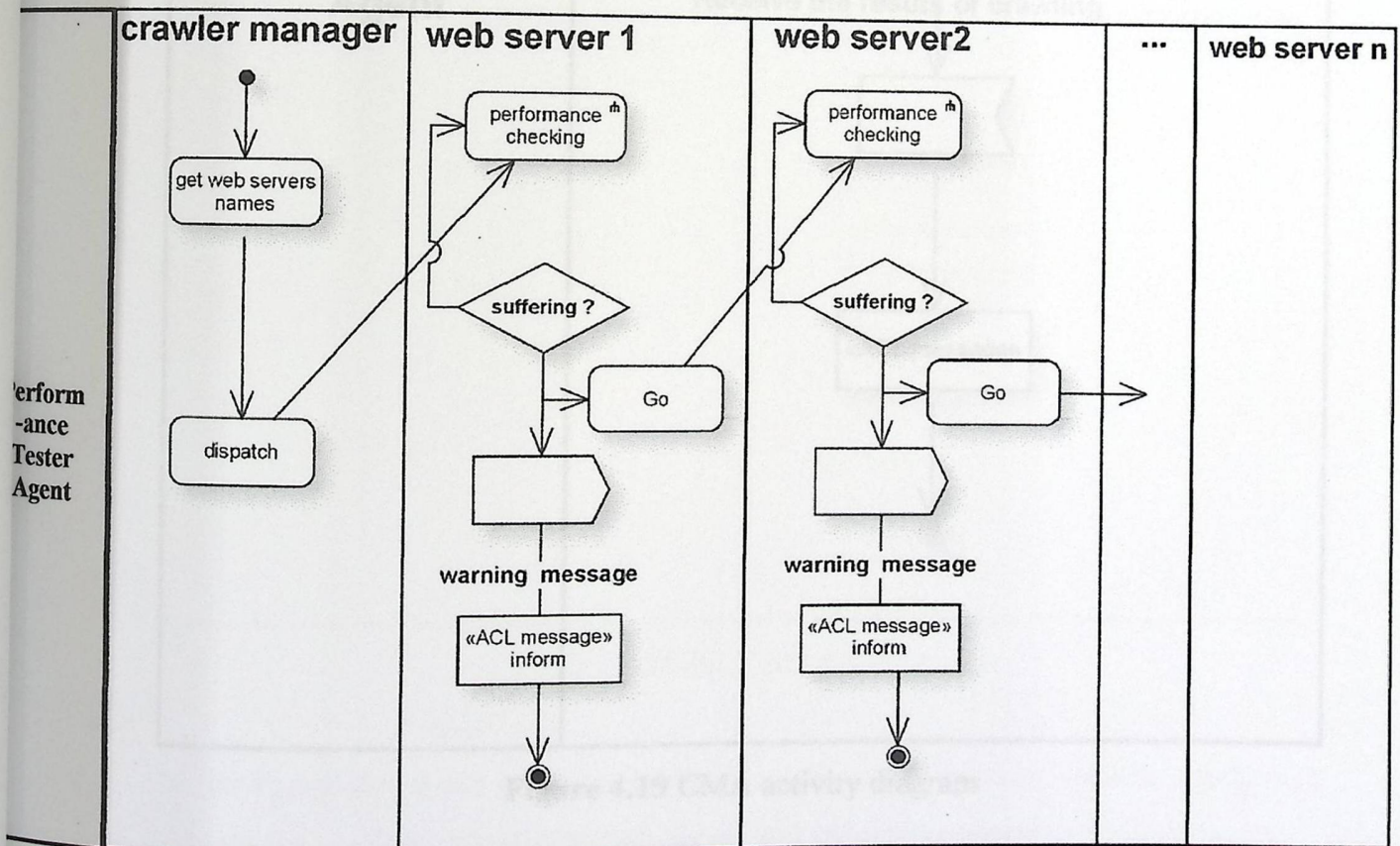


Figure 4.18 PTA activity diagram

4.4.4. Crawler Manager Agent (CMA)

This agent resides in the crawler manager (Home). It acts as a receiver for the messages created by the crawling and monitoring agent aided by its cyclic behavior which enables it to be aware of every update sent.

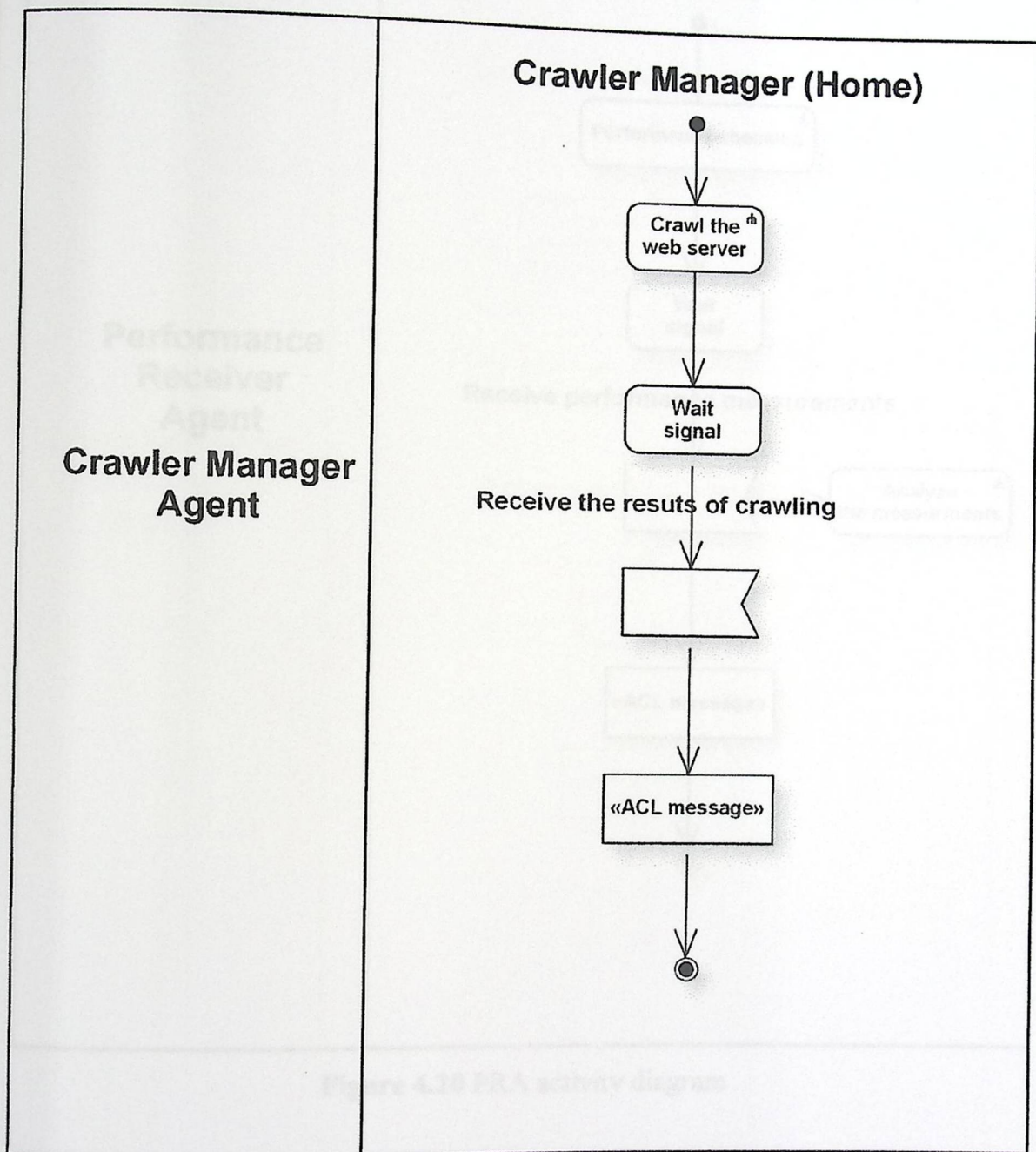


Figure 4.19 CMA activity diagram

4.4.5. Performance Receiver Agent (PRA)

From its name it indicates the rule on which it works, also it resides in the crawler manager (Home) and waits for the measurements sent by the performance Tester Agent.

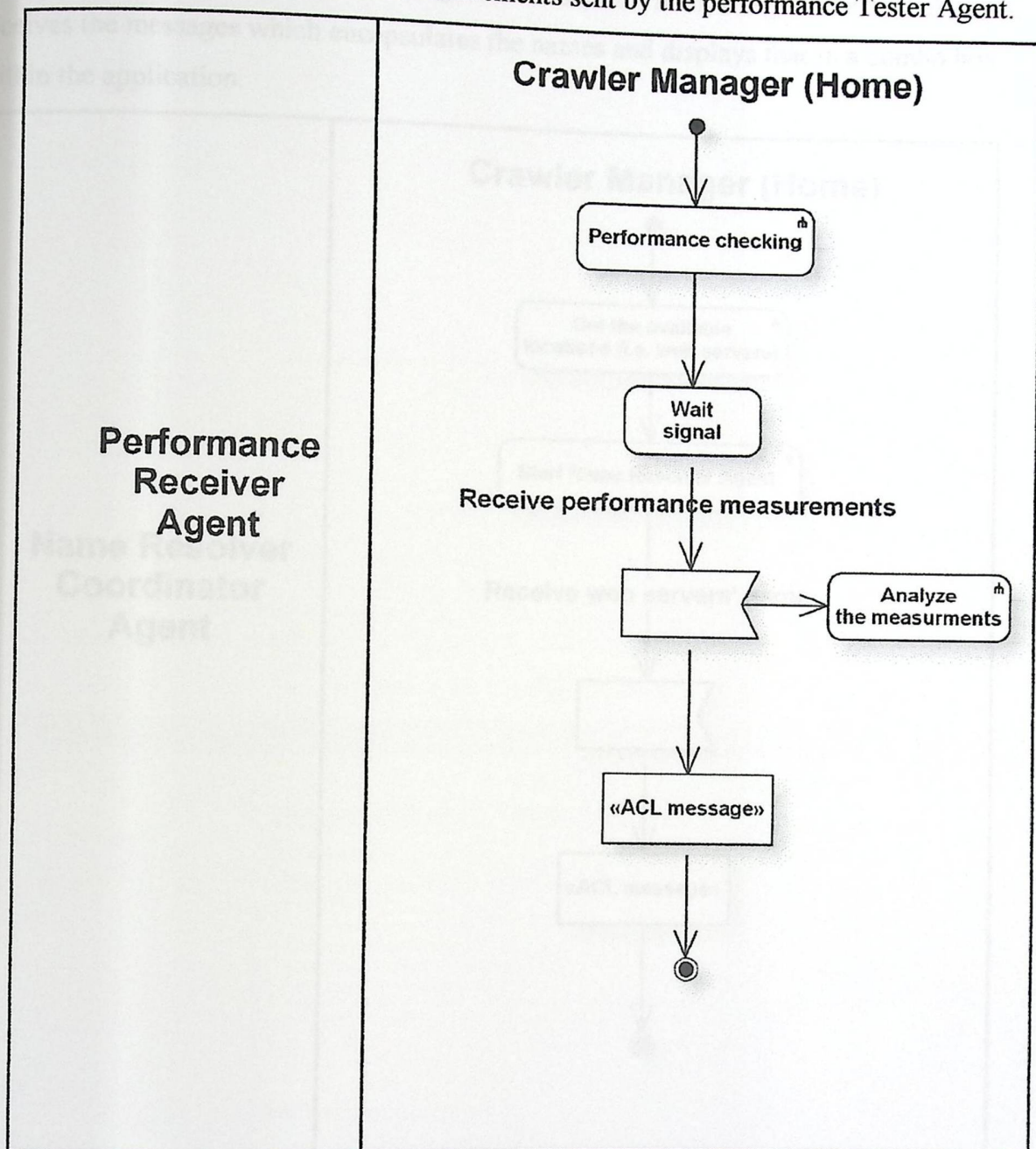


Figure 4.20 PRA activity diagram

In the figure above the sub activity "analyze the measurements" include a judgment processes to indicate a suffering web server and request a coocrawlers management.

4.4.6. Name Resolver Coordinator Agent

This lives in the crawler manager and guarantees dispatching the NRAs which are responsible for collecting and resolving the web servers' names. Again the NRCA waits, receives the messages which encapsulates the names and displays that in a combo box within the application.

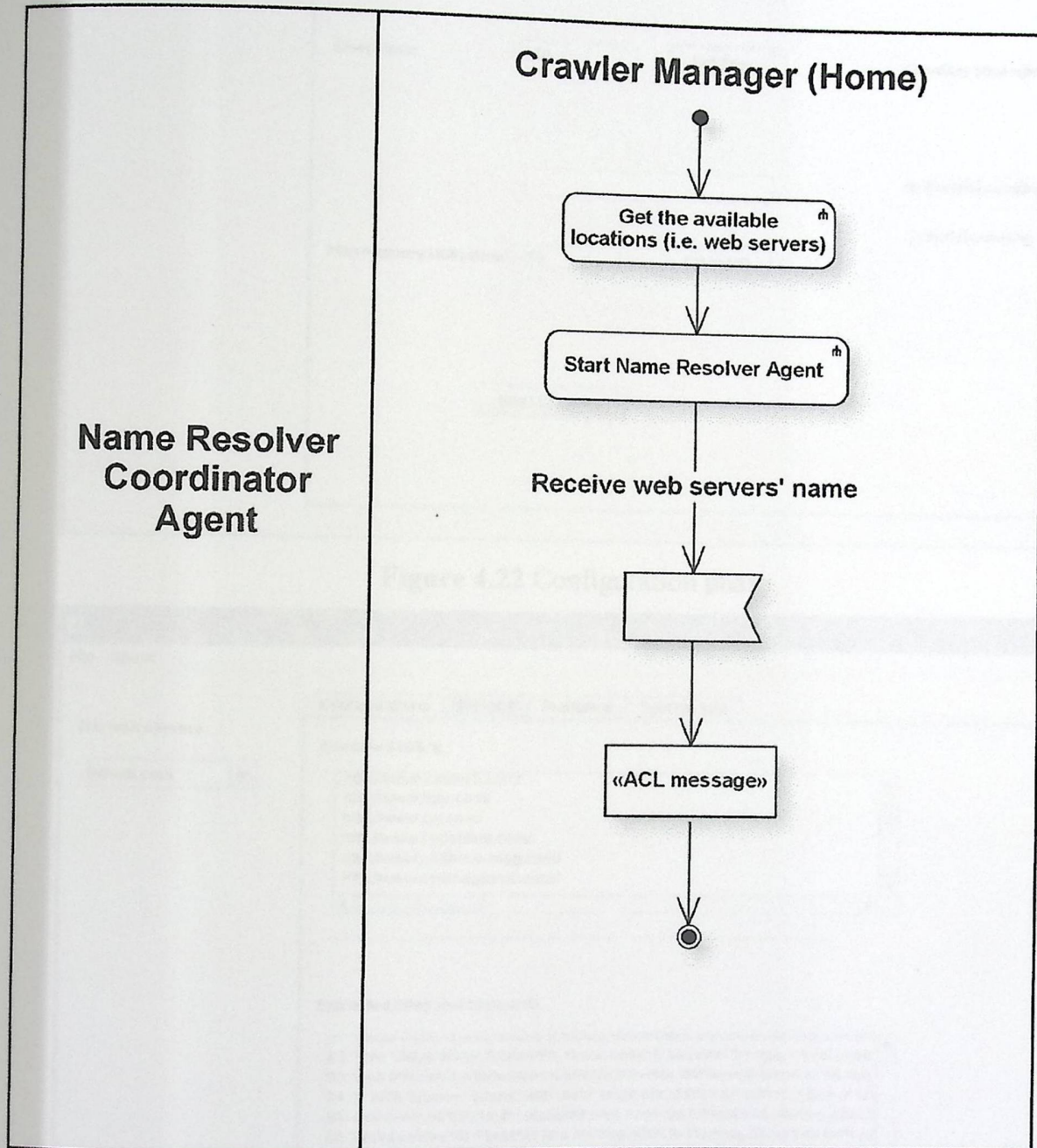


Figure 4.21 PRA activity diagram

4.5. GUI design

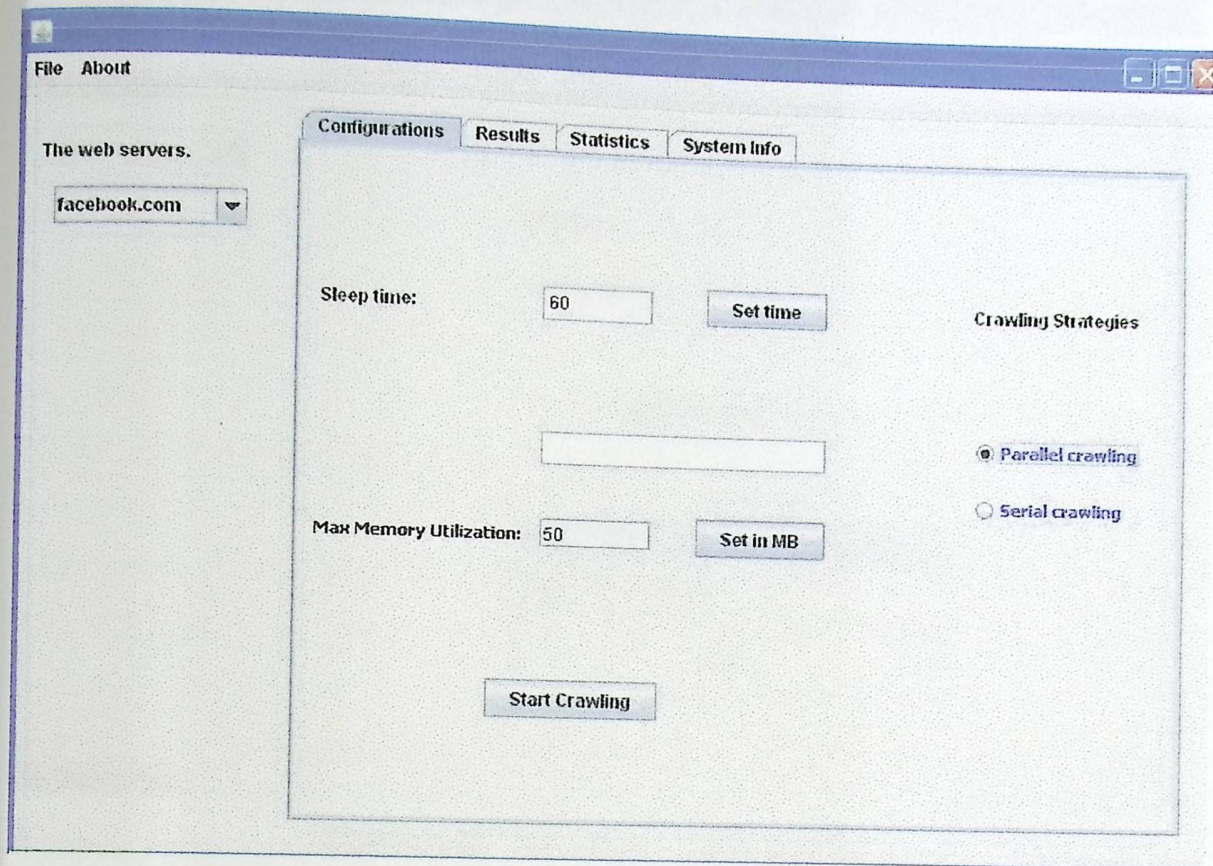


Figure 4.22 Configuration phase

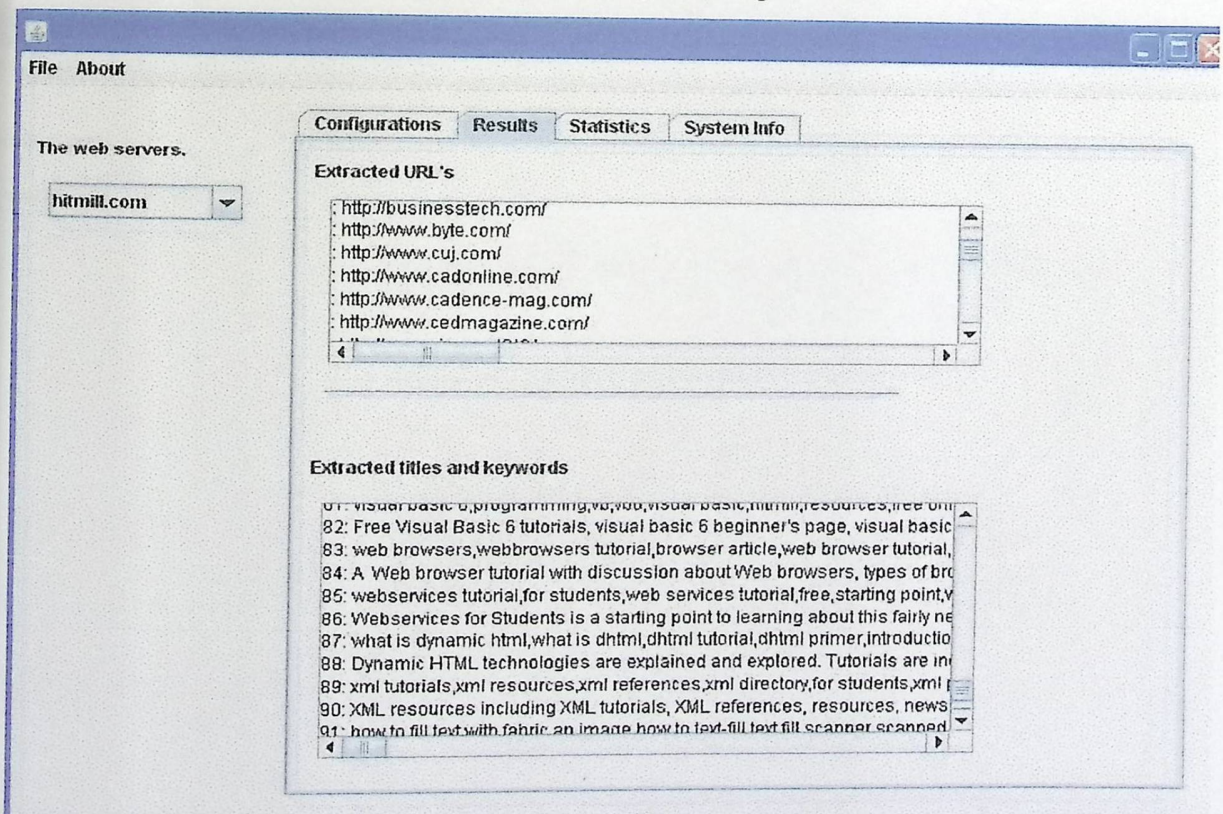


Figure 4.23 Results phase

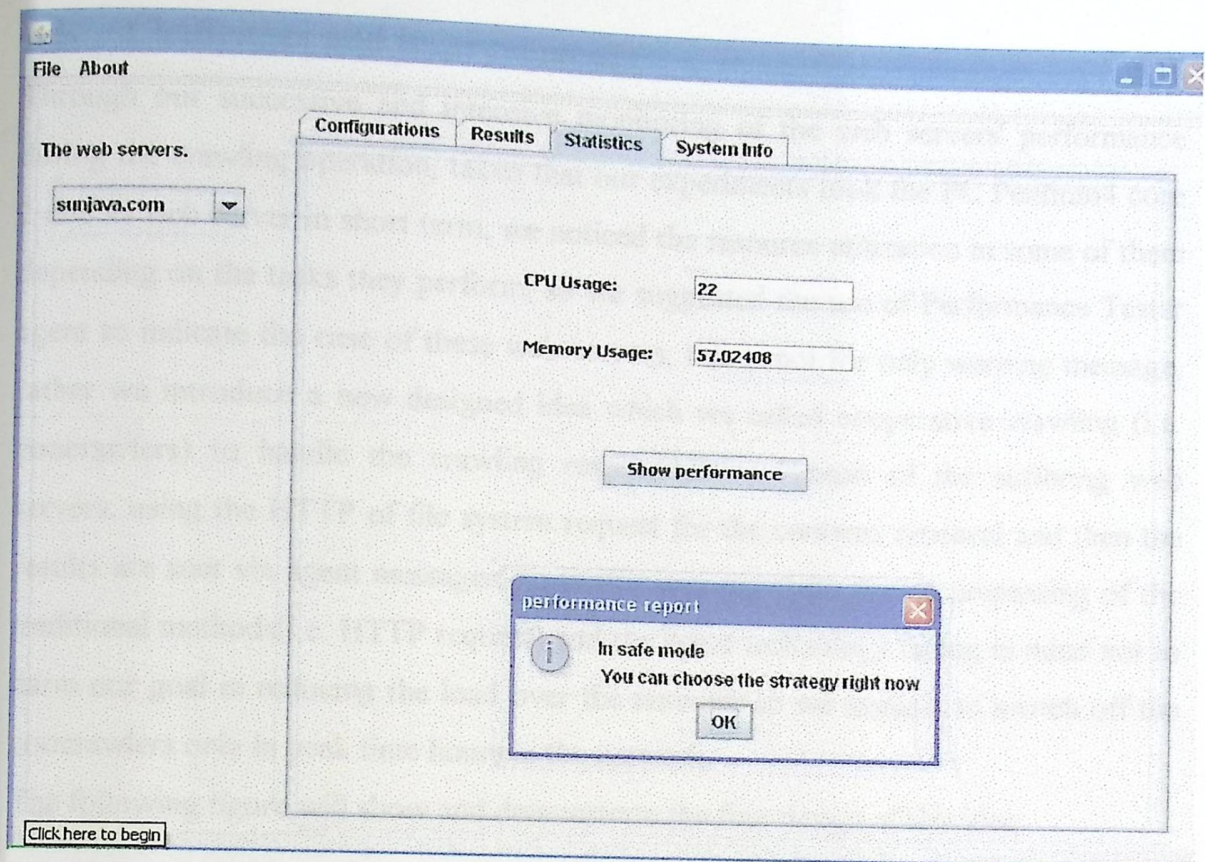


Figure 4.24 Performance statistics phase

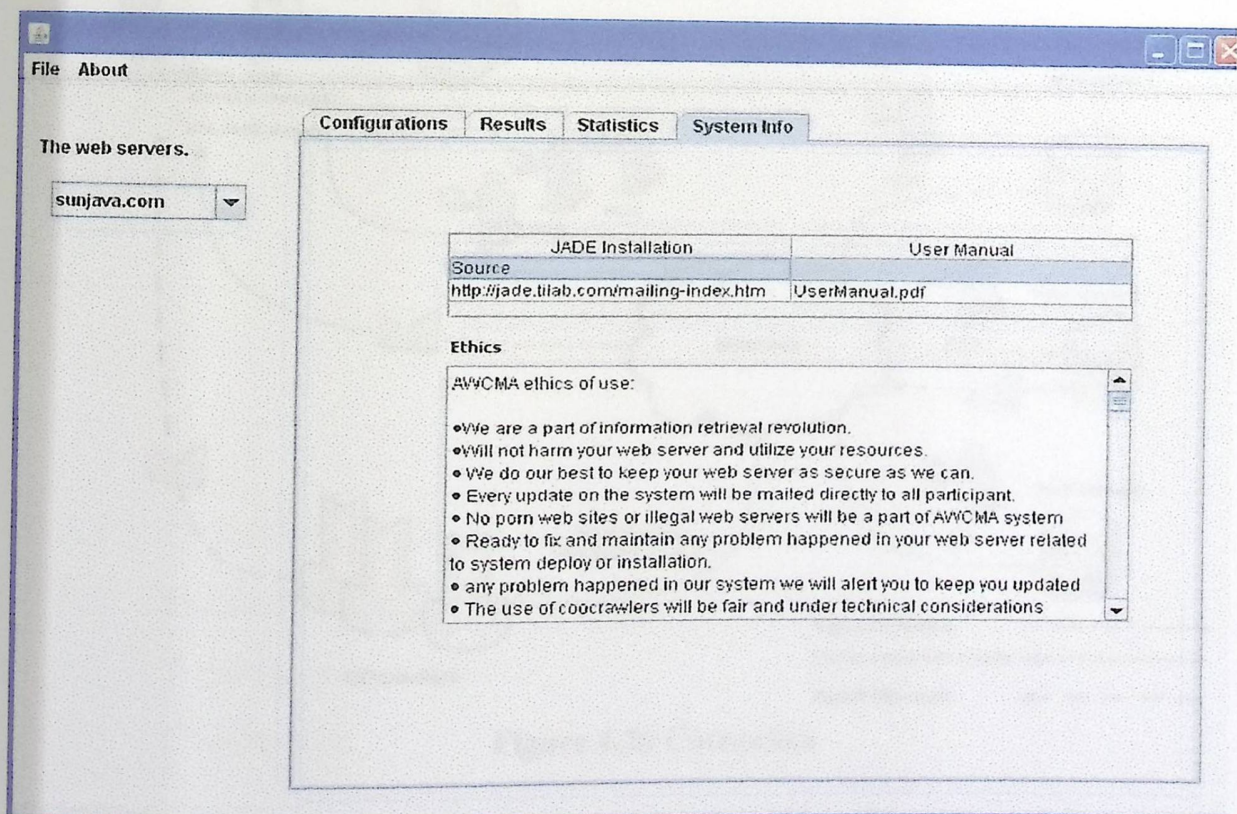


Figure 4.25 System info and ethics view

4.6. Challenges and improvements

Through our successive and intensive monitoring of the web servers' performance during the crawling operation, taken that our experiments took the PC Pentium4 core 2 due as web server in short term, we noticed the resource utilization in some of them depending on the tasks they perform; so we suggested the use of Performance Tester agent to indicate the case of these web servers, this is not for only warning message, rather we introduce a new designed idea which we called cooperative crawling (i.e. coocrawlers) to handle the crawling responsibilities instead of the suffering web servers, using the HTTP of file system request for the contents retrieval and then the results are sent via agent messages, so in this way we claim the encompassing of the traditional methods(i.e. HTTP request) and the agent technology taking in mind not to harm our goal in reducing the load over the network so we suggest to switch off the coocrawlers only in peak time hours in the network.

The following figure will show and demonstrate the first design of this idea.

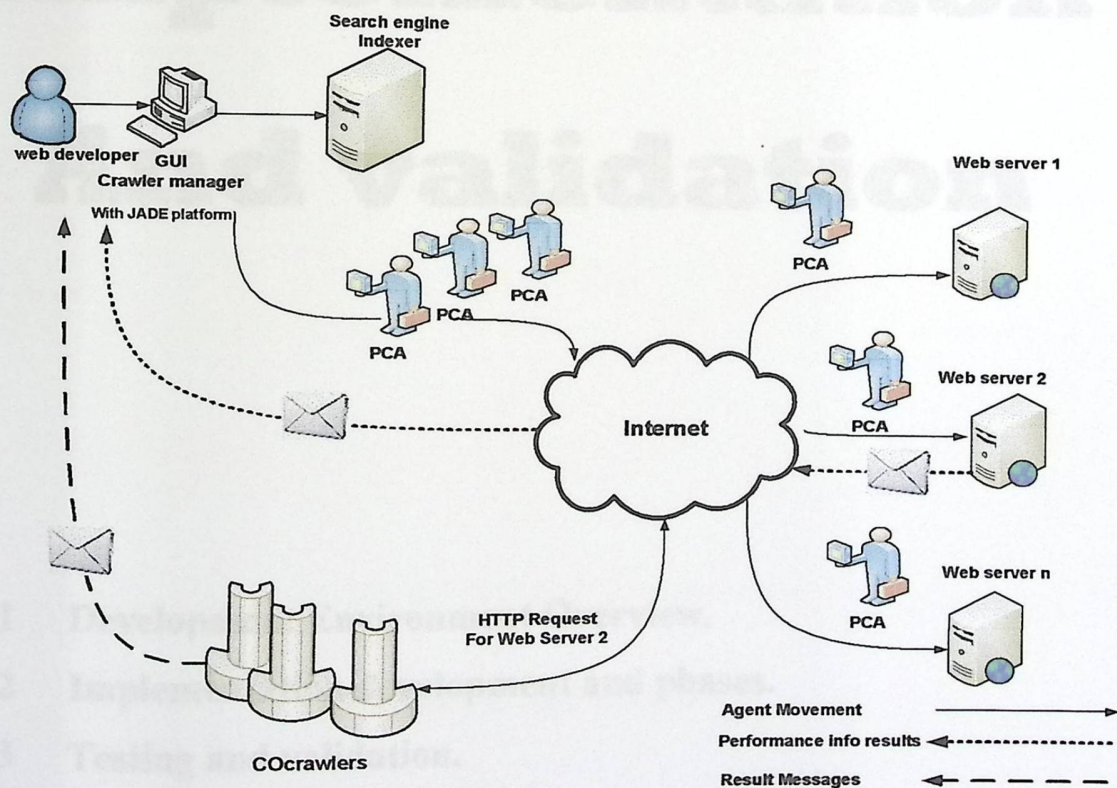


Figure 4.26 Cocrawlers

Chapter Five

Implementation And validation

- 5.1 Development Environment Overview.**
- 5.2 Implementation development and phases.**
- 5.3 Testing and validation.**
- 5.4 Challenges.**

Chapter five

In this chapter we demonstrate the implementation phase of developing the system and the testing conditions upon which the validation was done.

5.1. Development Environment overview

The lower case tools which we used in developing the system can be explained as the following:

- NeatBeans IDE 6.5 : which we found flexible and free, we used it for both developing the GUI of the application and naturally for writing the source code

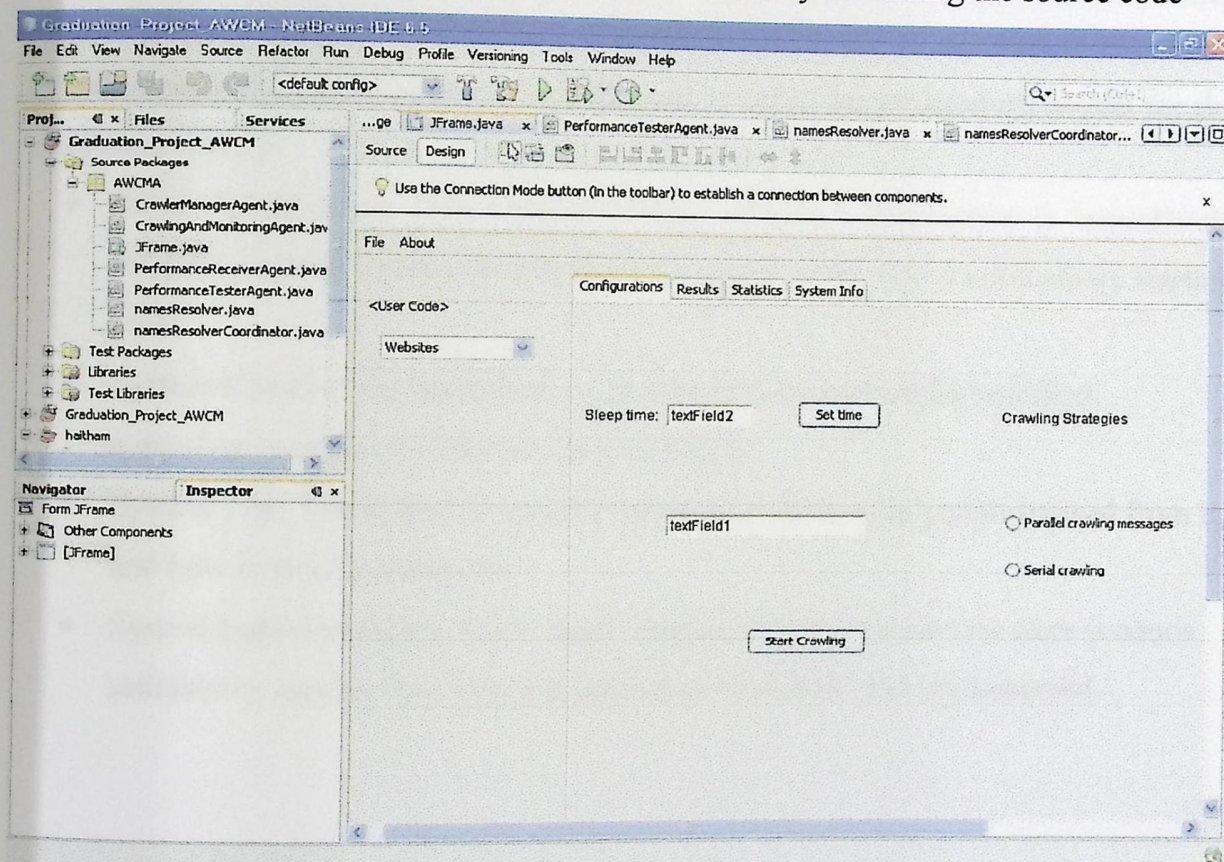


Figure 5.1. NetBeans IDE

- JADE packages: installed on both the crawler manager and web servers (i.e. testing computers) to enable an agent platform there and containers to accept the migrating agents to carry out the whole process.
- JDK: to compile java source code into byte code then to machine code to be executable file. In ours we used JDK 6 update 11 which we found suitable.

5.2. Implementation development and phases.

5.2.1. Implementation development

Through the survey of the crawler applications –either traditional or agent based- we noticed that they differ hugely based on requirements and specifications; symmetrically when we started to gather the requirements of our system there were subtle requirements in hand so we suggested to adapt the **exploratory model** for the coming reasons:

- Initial Specification Development. Using whatever information is immediately available, a brief System Specification is created to provide a rudimentary starting point.
- System Construction/Modification. A system is created and/or modified according to whatever information is available.
- System Test. The system is tested to see what it does, what can be learned from it, and how it may be improved.
- System Implementation. After much iteration of the previous two steps produce satisfactory results, the system is dubbed as “finished” and implemented.

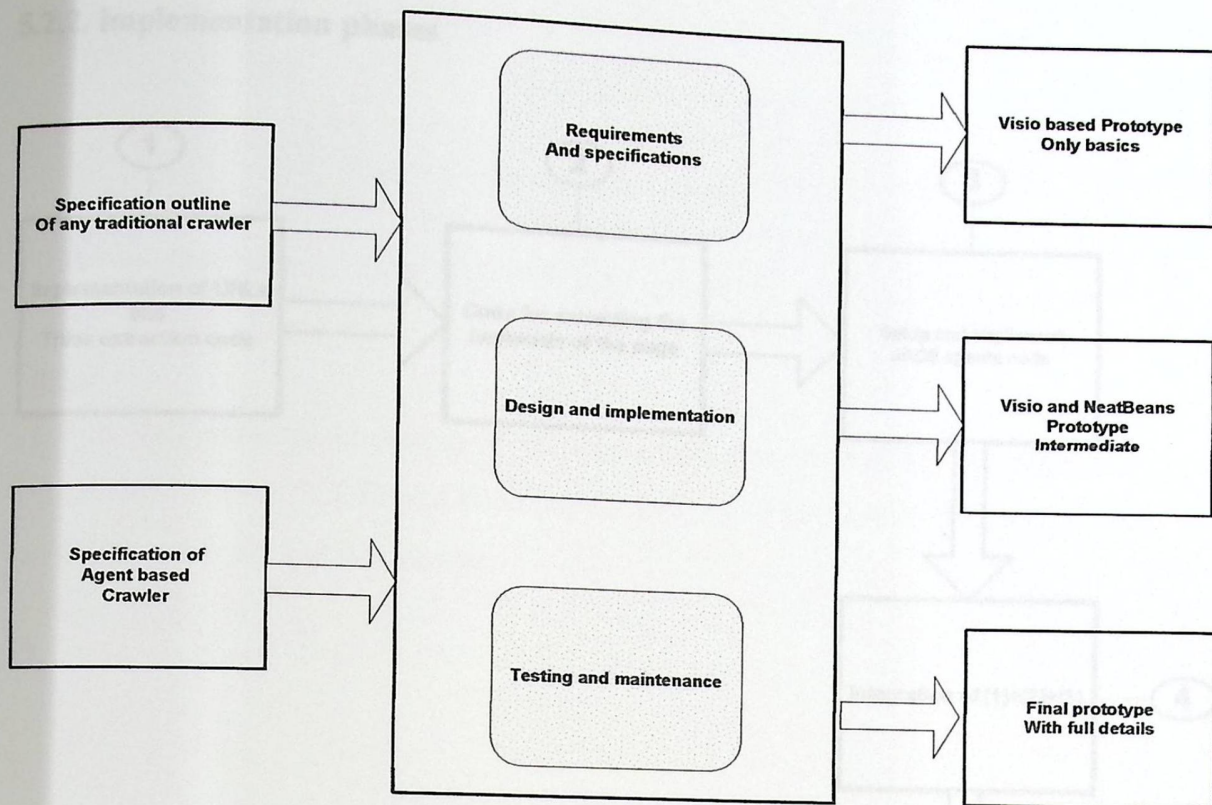


Figure 5.2. Prototype modeling

Beside that prototype models were generated as shown in the figure (5.2) to allow system improvements and maintenance coming to the final prototype which generally carries the full details and requirements desired.

Figure 5.3. Implementation phases

The figure (5.3) summarizes the implementation phases by which we brought the system to reality.

Step 1:

Implementation of URLs and titles extraction code that reads a set of HTML pages and save the extracted URLs and titles to a specific file stream.

```

while (true) {
    try {
        // Read the next line from the input stream
        String line = br.readLine();
        // If the line contains a URL, extract it
        if (line.contains("http://")) {
            // Extract the URL
            String url = line.substring(line.indexOf("http://"), line.indexOf("\n"));
            // Add the URL to the list of extracted URLs
            urls.add(url);
        }
    } catch (IOException e) {
        // Handle the exception
        e.printStackTrace();
    }
}
  
```


5.2.2. implementation phases

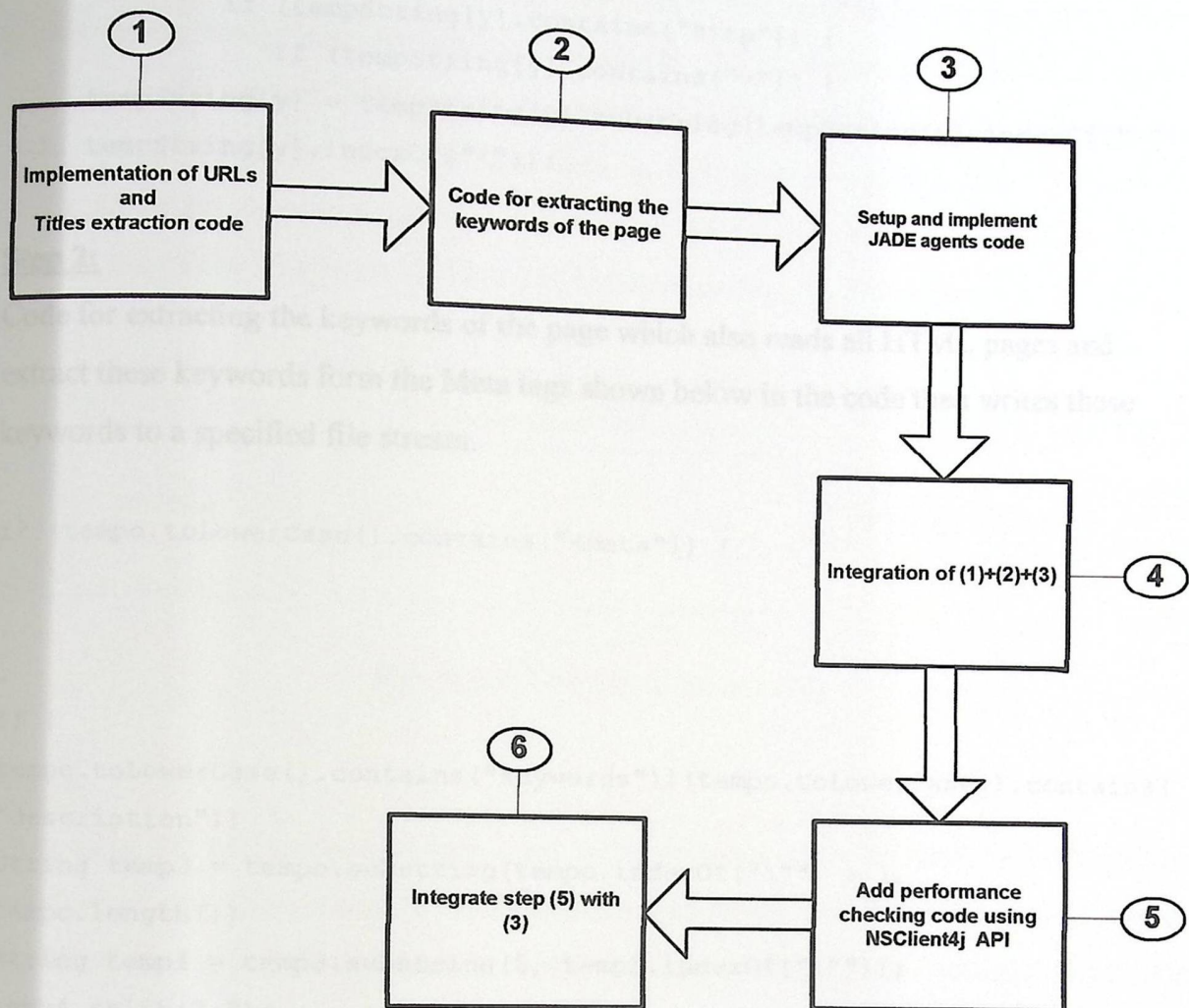


Figure 5.3. Implementation phases

The figure (5.3) summarizes the implementation phases by which we brought the system to reality.

Step 1:

Implementation of URLs and titles extraction code that reads a set of HTML pages and write the extracted URLs and titles to a specific file stream.

```
while ((temp = br.readLine()) != null) {  
    if (temp.contains("href") && temp.contains("http") &&  
    !temp.contains("javascript")) { //javascript will add additional load  
        String[] tempString;
```



```
tempString = temp.split("\\");
for (int y = 0; y < tempString.length; ++y) {
    if (tempString[y].contains("http")) {
        if (tempString[y].contains(">")) {
            tempString[y] = tempString[y].substring(tempString[y].indexOf(">")
+ 1, tempString[y].indexOf("<"));
        }
    }
}
```

Step 2:

Code for extracting the keywords of the page which also reads all HTML pages and extract these keywords from the Meta tags shown below in the code then writes these keywords to a specified file stream.

```
if (tempo.toLowerCase().contains("<meta")) {
    .
    .
    .
    If {
        tempo.toLowerCase().contains("keywords") || tempo.toLowerCase().contains(
        "description")) {
            String temp3 = tempo.substring(tempo.indexOf("\\") + 1,
            tempo.length())
            String temp4 = temp3.substring(0, temp3.indexOf("\\"));
            temp4.split(",");
        }
    }
}
```

Step 3, 4:

Setup and implement JADE agents' code to carry out a simple data extraction from local platforms within the same host, encapsulates these data into a message and sends it to the other platform (i.e. crawler manager). The integration of the this code with agent's code formed the clear face for our system as specially as the crawling process has now to be done in the remote web servers without the need to download their contents to be processed.

Step 5:

Add performance checking code using NSClient4j API (org.nsclient4j package) which allows going freely into the system's resources (CPU, Memory, Hard Disk ...etc), entering to the area of Windows Performance Monitor (WPM) statistics and retrieving their measures relative to a specified time period, and can also access to low-level performance or operating statistics that can be useful in low-level code, such as the current rate of GETS against IIS (Internet Information Server) Web server.

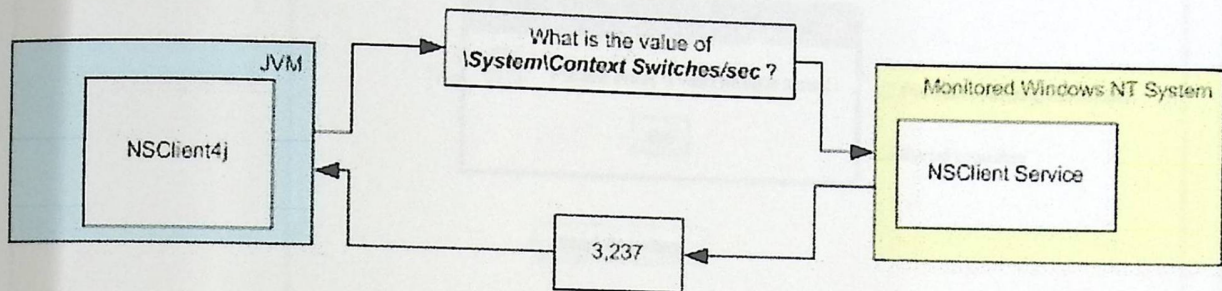


Figure 5.4. NSClient4j service [33]

Step 6:

Integrating the performance testing code with the agent code so that the performance tester agent can simply travel to the destination web server, do all the checking processes explained above, and finally sends a report message to the crawler manager to provide full measurements of participated web servers' resource utilization and in turn allow the cocrawlers to judge whether to start working or to continue sleeping.

5.3. Testing and validating

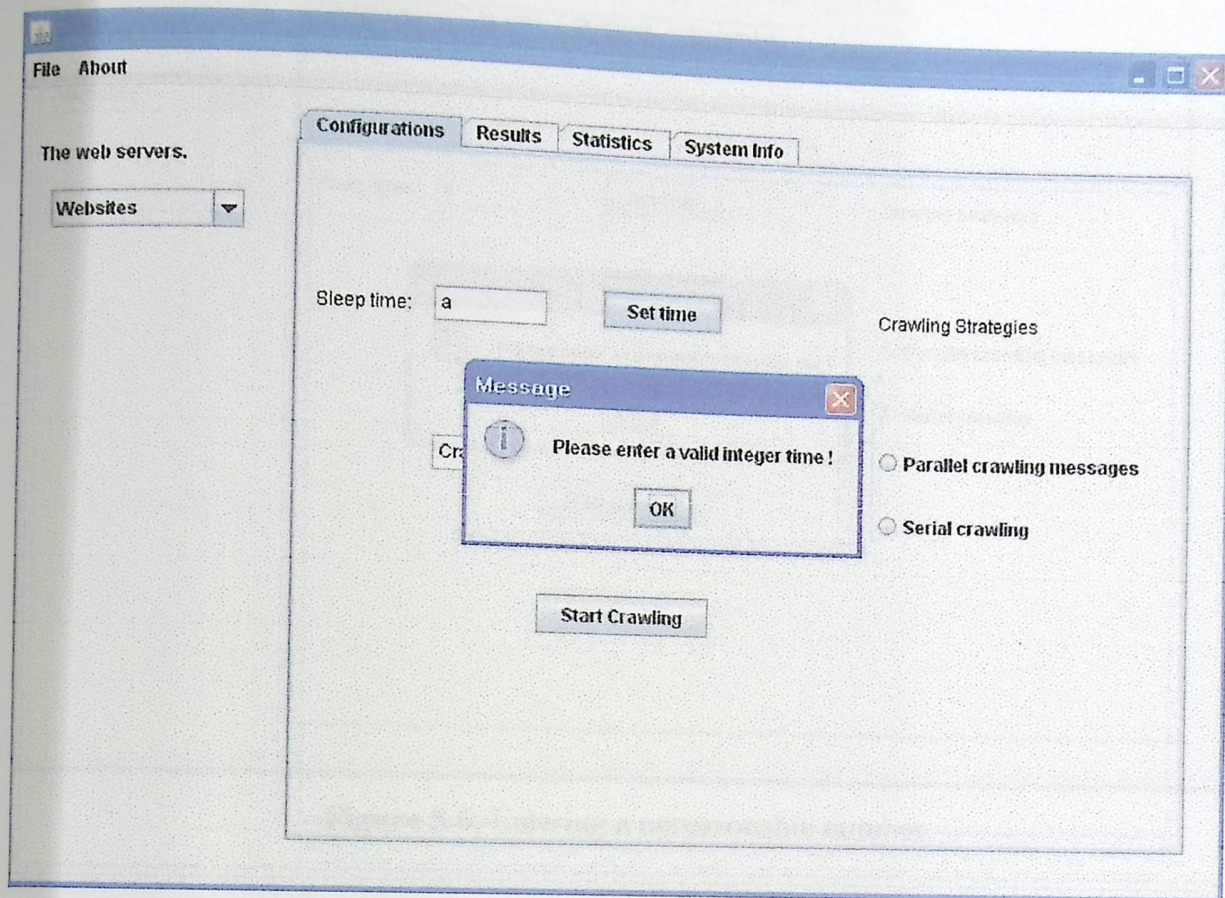


Figure 5.5. Entering invalid number

Generally the user may mistake when entering the time avoiding that we suggested to warn him by this message window.

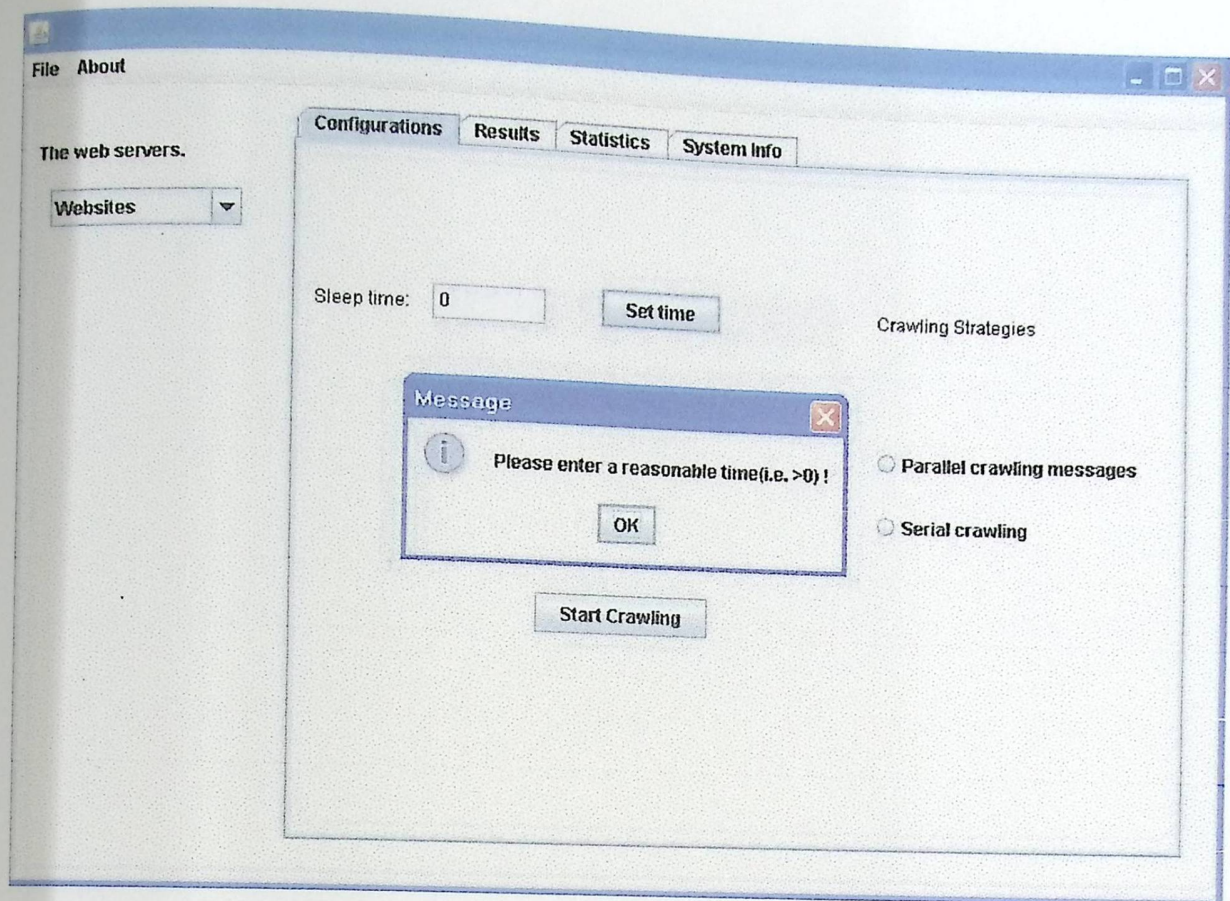


Figure 5.6. Entering a unreasonable number

The user may also make a mistake during the monitoring time determination, so another warning message forbids him from entering time less than zero.

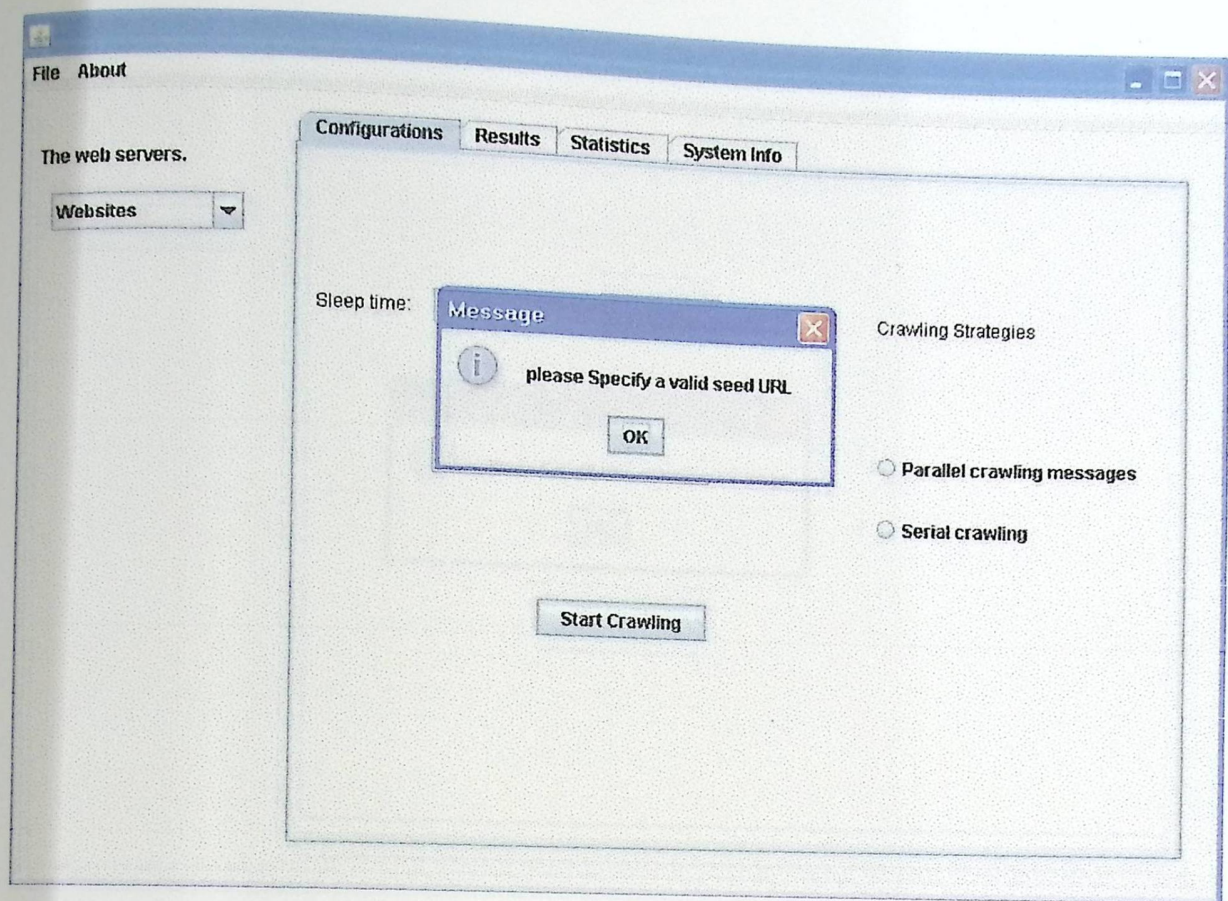


Figure 5.7. Selecting a invalid URL

Always you have to decide the seed URL as a starting point itinerary for the crawling process.

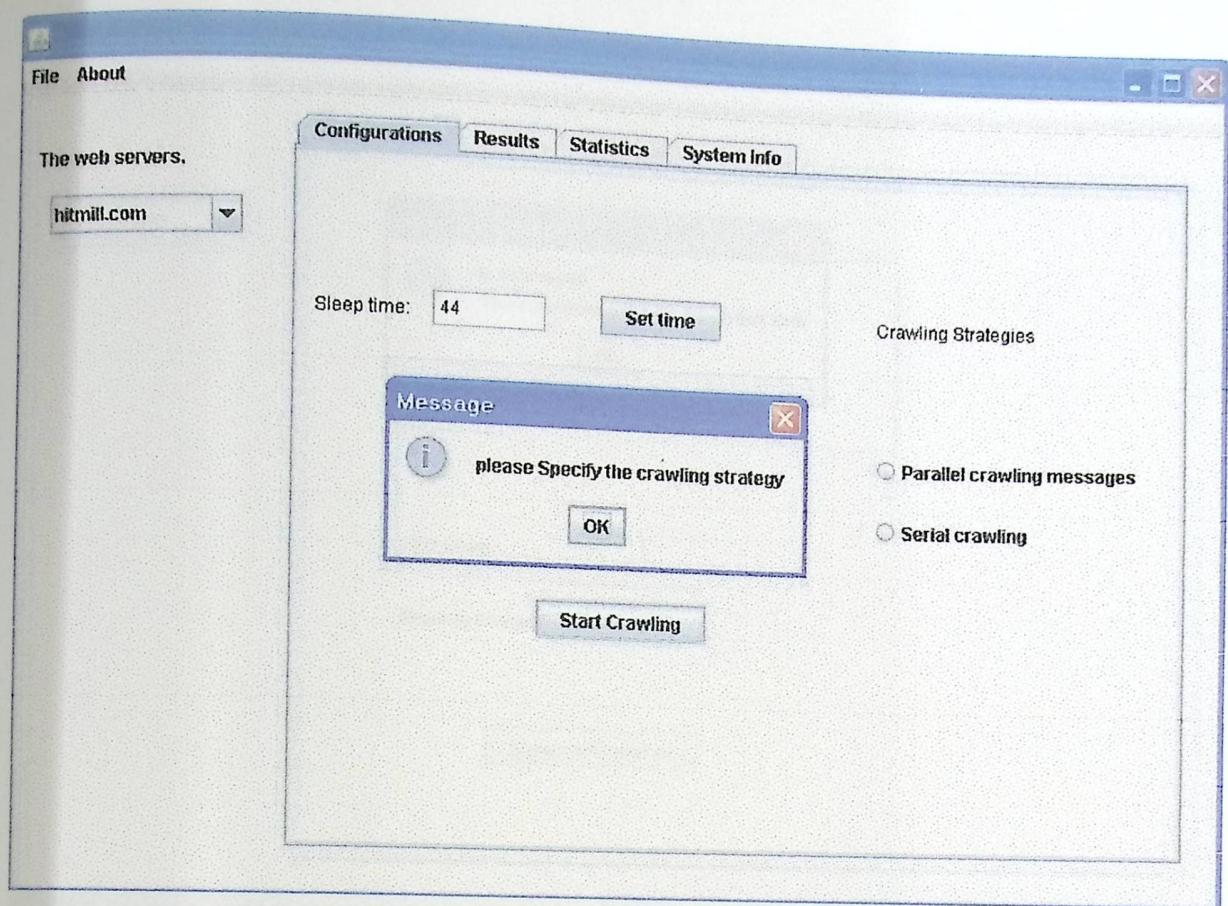


Figure 5.8. Forgetting to select the strategy

Also it is a strong requirement to select the crawling strategy (serial or parallel) to complete the trip plan for the CMNA.

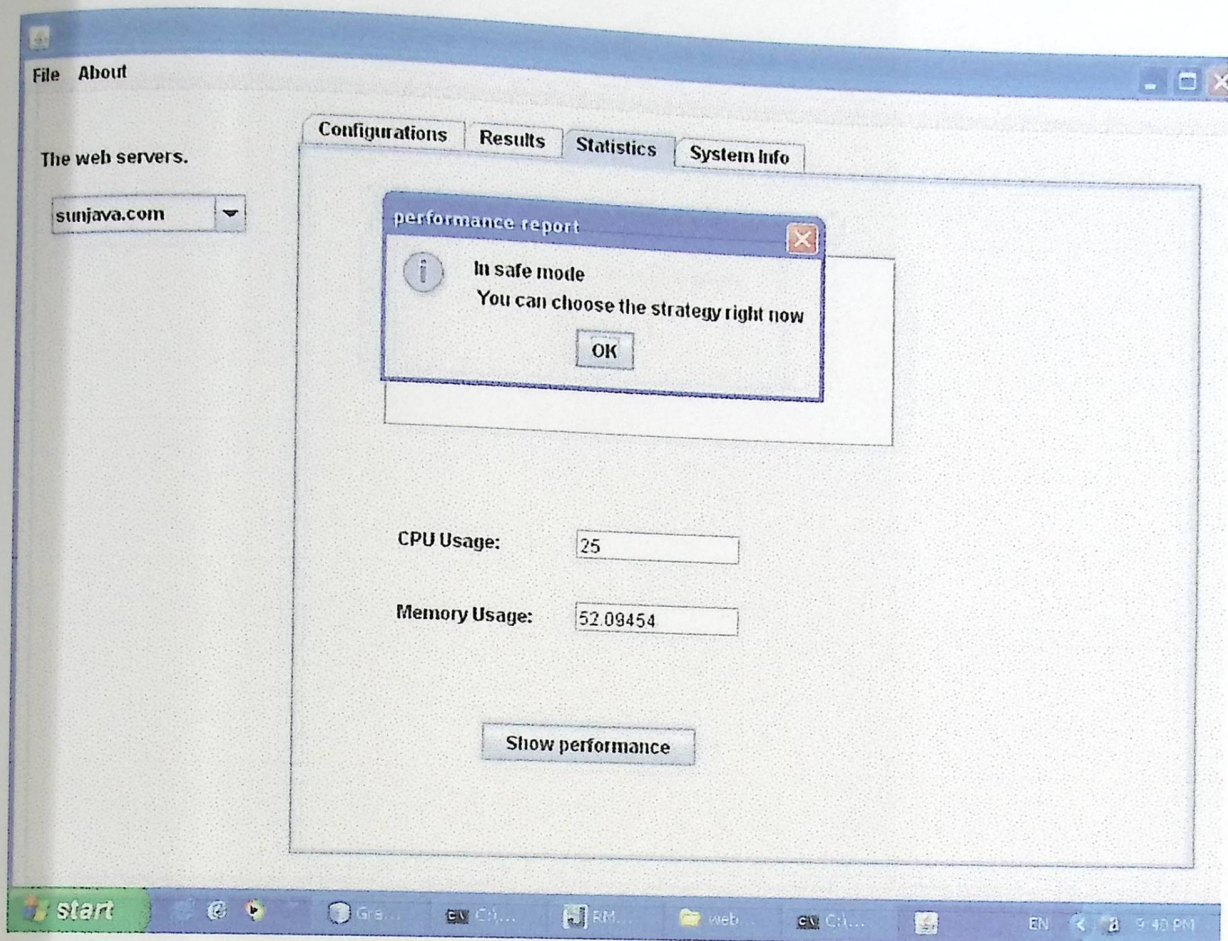


Figure 5.9. Safe mode

In this case the performance Tester agent returns a report message that says everything is fine the resources are not utilized to the maximum specified, we can select the crawling strategy to begin the process. In our test for the program you notice in the figure above that we keep the Start bar shown to simulate the case of unexhausted resources (i.e. small number of opened applications).

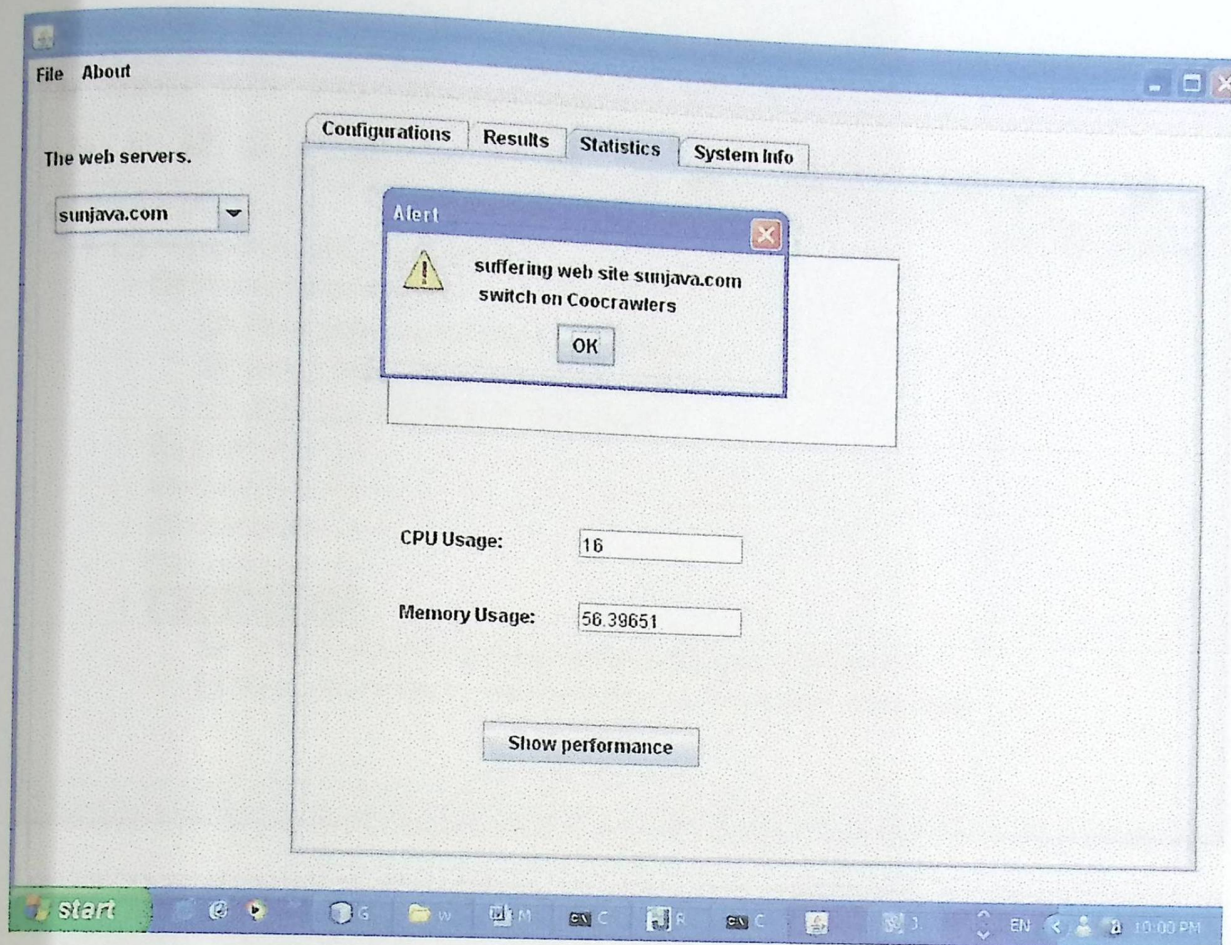


Figure 5.10. Unsafe mode (Switch to the Coocrawlers)

In this case the performance Tester agent returns a report message that warns the administrator for the suffering web server name and tell him to switch on the coocrawlers. Again we kept the Start bar to show multiple applications running (i.e. highly utilized resources relatively).

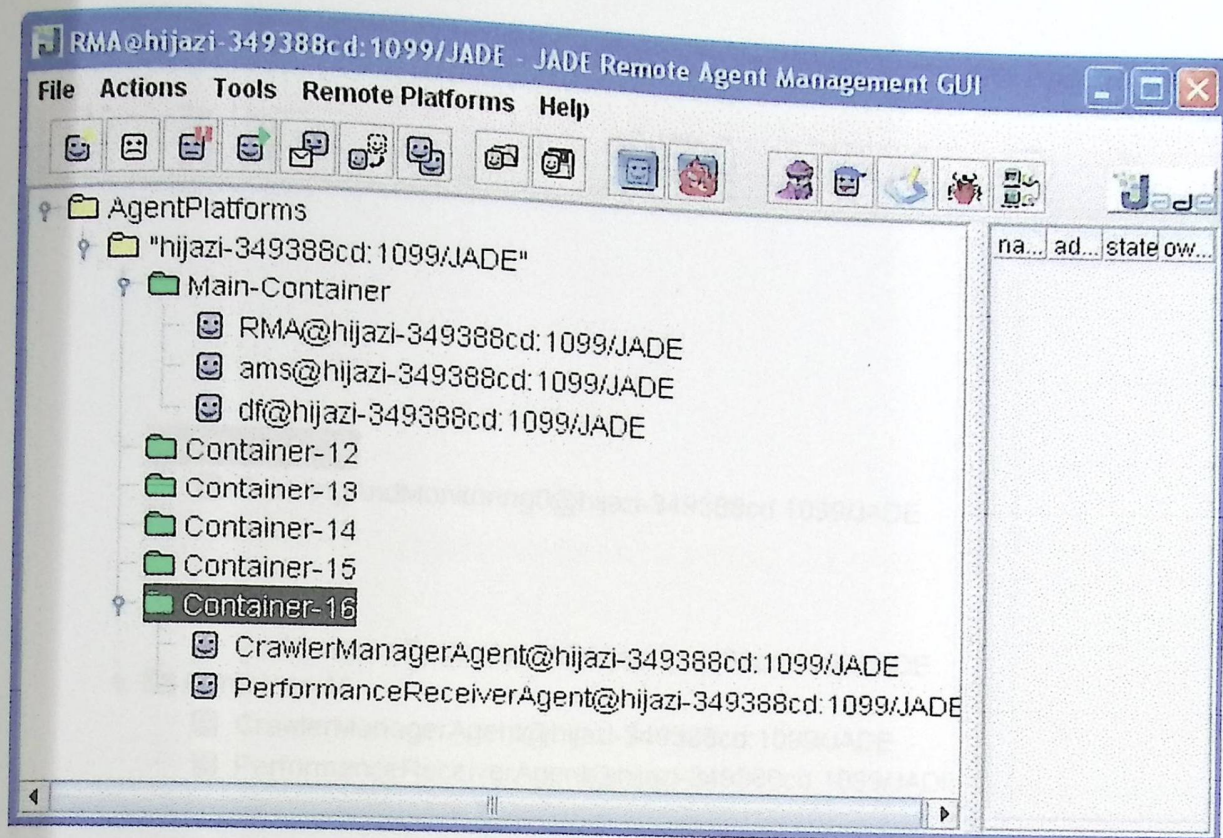


Figure 5.11. RMA at program startup

We can notice that when starting the program, two agents are already in the crawler manager (i.e. Container-16) the crawler manager agent and the performance receiver agent.

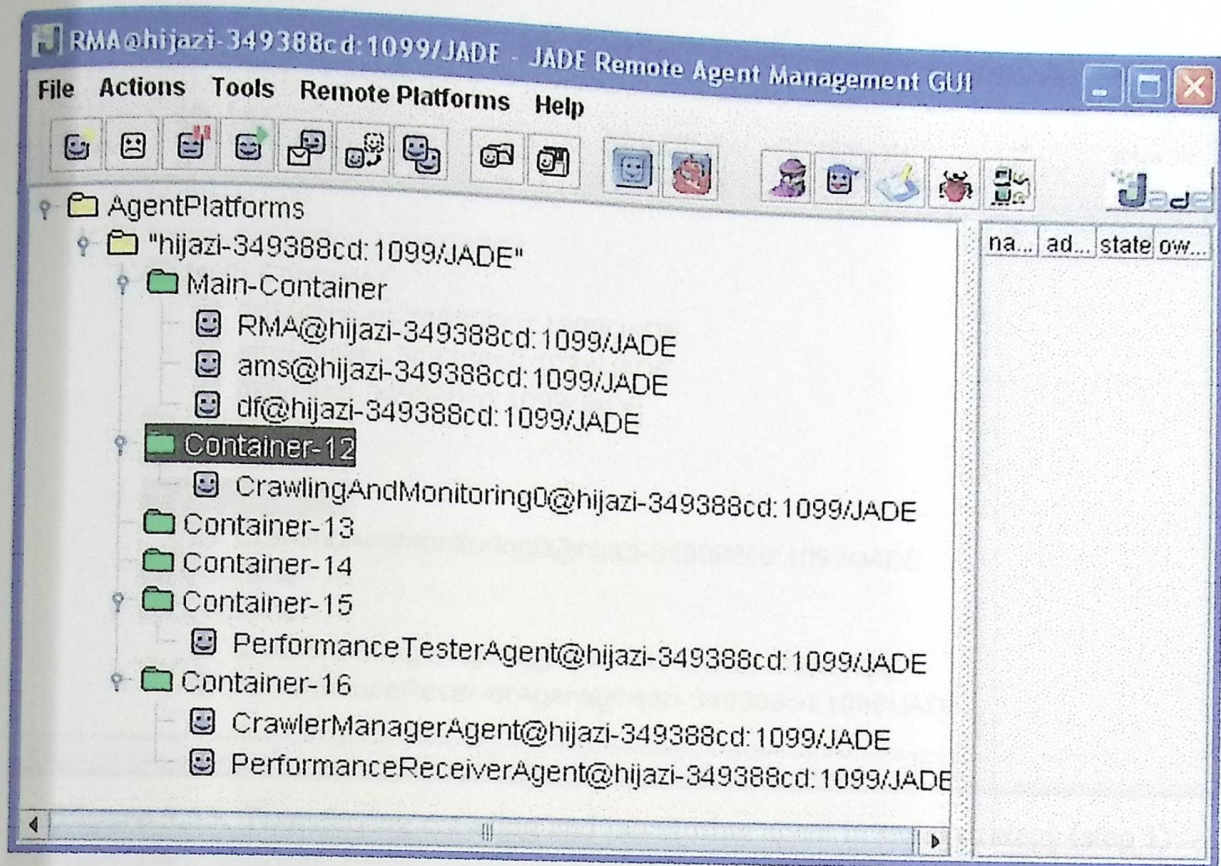


Figure 5.12. Transferring crawling and monitoring agent

Before transferring crawling and monitoring agent to container-15 the performance Tester agent transfers there to test the resources.

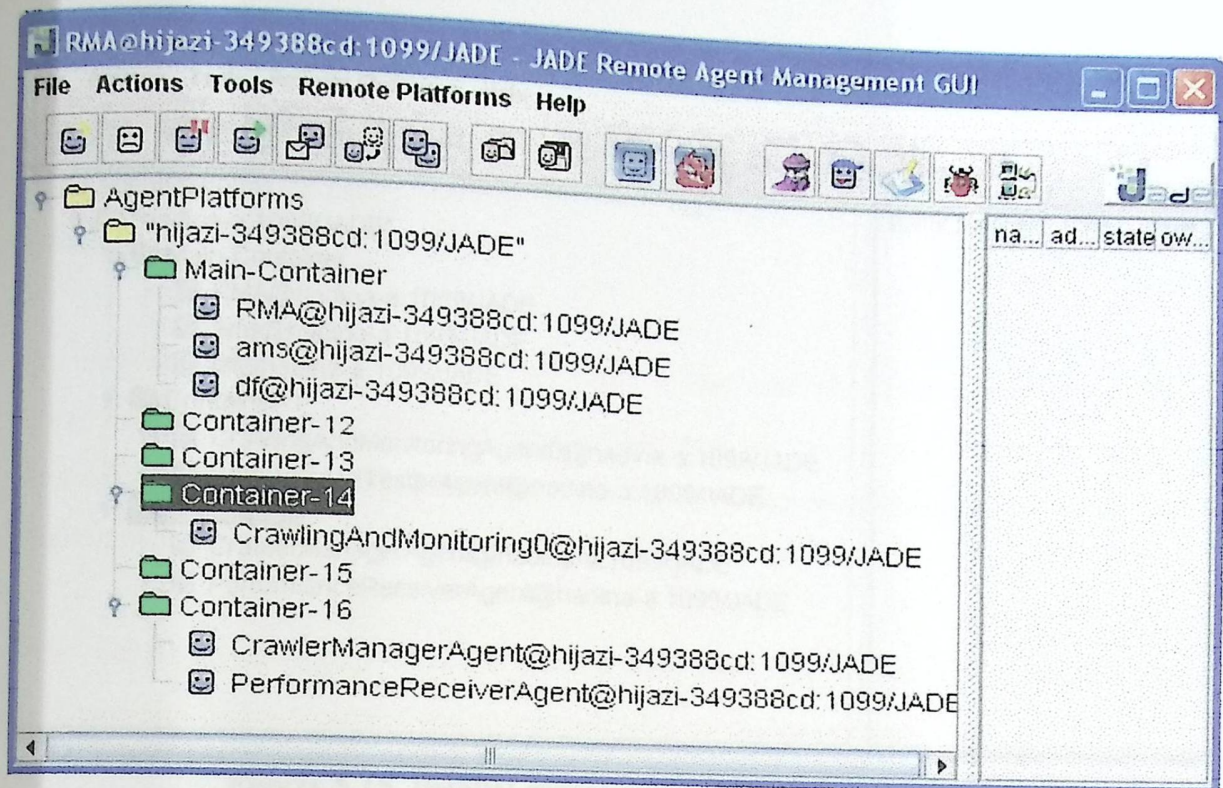


Figure 5.13. Transferring crawling and monitoring agent in serial strategy (step 1)

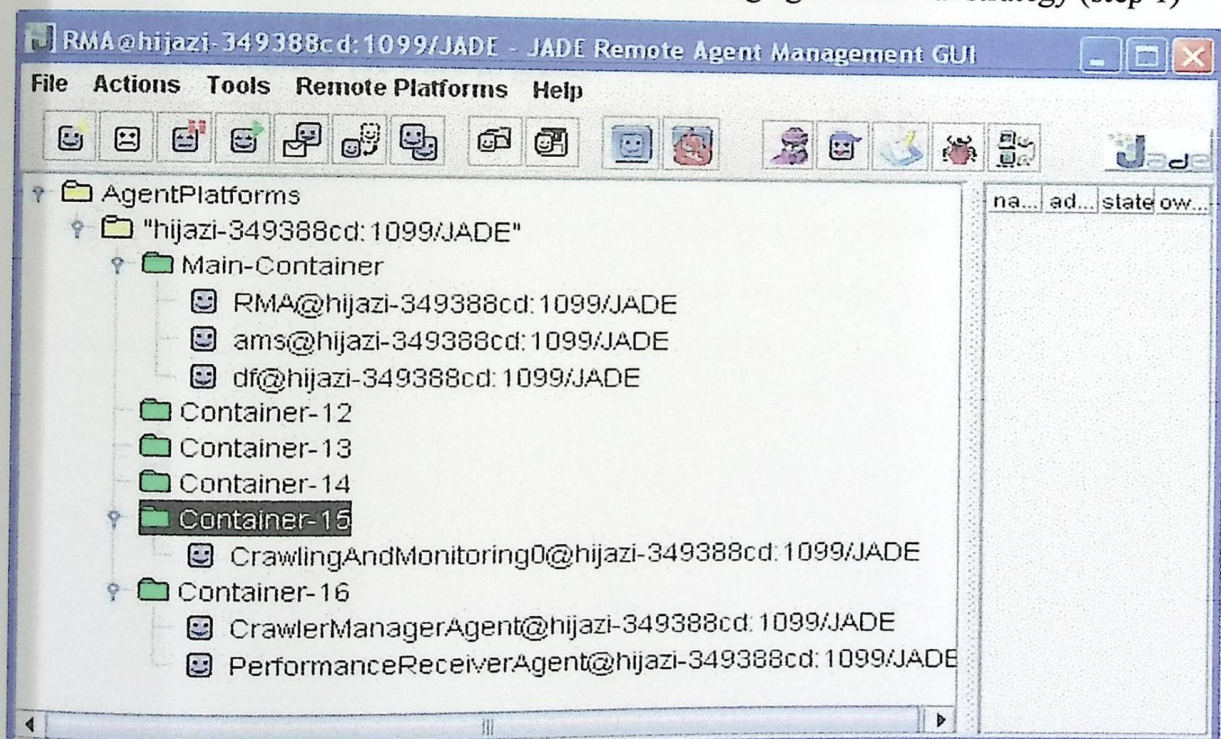


Figure 5.14. Transferring crawling and monitoring agent in serial strategy (step 2)

In figure 5.14. and 5.15. the serial strategy for example starts here at Container-14 and after that moves to container -15 and container -n.

5.4. challenges

Through our implementation of the system we faced various challenges solved via consulting our supervisor, reading external materials, connecting with professional friends.

The most important challenge when we came to implement the performance tester agent, at the beginning we read through Java forums and tutorials that there are no available Java APIs that could test the performance of the operating system and access its resources freely and simply.

After intensive searching through forums and sites, we found the great NSClient4j packages and API which allowed us successfully to implement the PTA.

Other minor challenges related to implementation were acted in extracting the HTML tags with various formatting depend on the designer of the page.

Finally we faced another challenge which is the heap size of JVM because of the large size of data to be processed, this was solved by changing some parameters in netbeans.conf file and project proprieties inside the IDE.

6.1 Experiments and mathematical description.

6.2 Conclusion.

6.3 Future work.

6.4 Challenges.

Chapter six

In this chapter we demonstrate the experiments done to test the efficiency and reliability of the system showing how much we reached the goals previously assigned, coming to the conclusions after these experiments, and finally speak freely about the possible enhancements that could be done as a future work.

6.1 Experiments and mathematical description.

From the point of view that the experimental metrics which could be measured are strongly related to the goals assigned to the system, we chose the data size (i.e. for bandwidth saving) and time as two measurable values in our experiments.

We noticed that over 80% of related works have used mathematical description

Results, conclusions

And

Future work

- 6.1 Experiments and mathematical description.
- 6.2 Conclusions.
- 6.3 Future work.
- 6.4 Challenges.

Chapter six

In this chapter we demonstrate the experiments done to test the efficiency and reliability of the system showing how much we reached the goals previously assigned , coming to the conclusions after these experiments, and finally speak freely about the possible enhancements that could be done as a future work.

6.1 Experiments and mathematical description.

From the point of view that the experimental metrics which could be measured are strongly related to the goals assigned to the system, we chose the data size (i.e. for bandwidth saving) and time as two measurable values in our experiments.

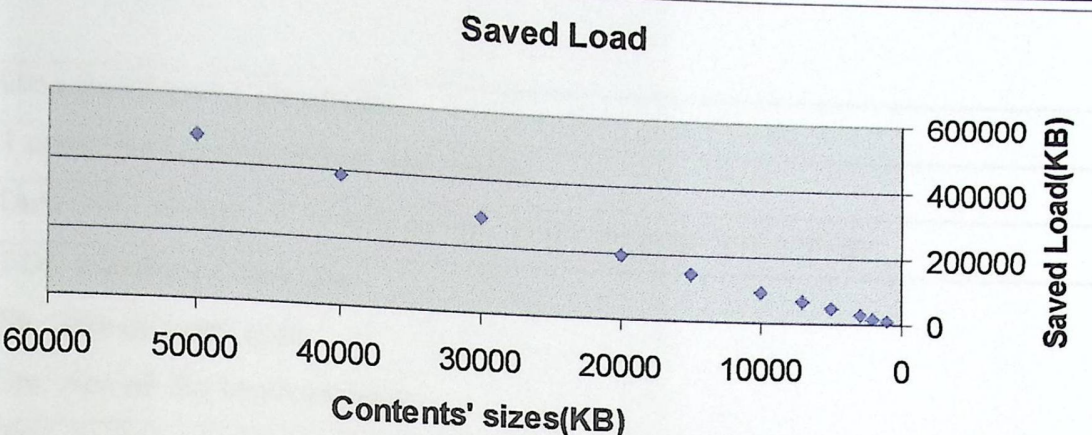
We noticed that over 80% of related works have used mathematical description (equations and statistics) to show their work under real lens and hence we decided to make such description to have a reasonable, fair and justified comparison.

6.1.1. The first experiment.

The save in Load relative to the network over which the search engines live is an essential purpose in AWCMA system, so in this experiment we approached the proportionality between the web servers' contents and the save in load as a function of the content size, the form below demonstrates the experiment in an organized way.

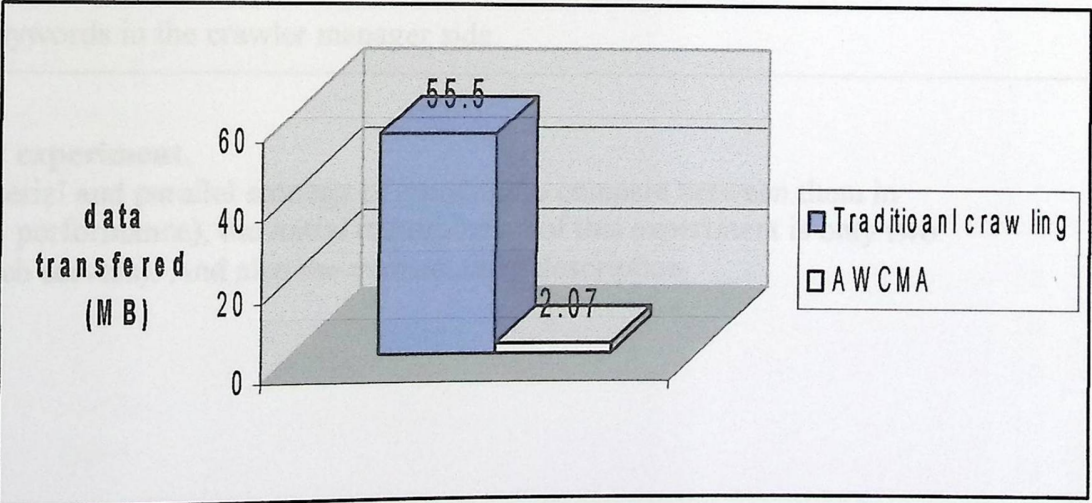
Notice here that we used real web pages from different web sites with various size ranges.

Name	Saved load
Tools	Equations and 2 computers
Input	Different sizes of web servers' contents, number of web servers to be crawled N In this experiment N=10
Symbols	Sc: size of page contents L(Sc): the load over the network S(sc) the saved load for different sizes Sr: size of HTTP request Sp: size of HTTP response Sa : size of agent code Sm: size of the result message
Output	Saved Load S(sc)
Theory	We claim that the use of mobile crawlers will reduce the load over the network in an efficient manner with the following parameters and equations we can prove $L(Sc)_{\text{traditional}} = N(Sr + Sp + Sc)$ $L(Sc)_{\text{AWCMA}} = N(2 * Sa + Sm)$ **We considered that the code of the name resolver agent plus the monitoring and crawling agent will be transferred initially so $2 * Sa$. By examining the relation between Sc and Sm we concludes that as follows : $Sm = .014 * Sc$ $S(sc) = L(Sc)_{\text{traditional}} - L(Sc)_{\text{AWCMA}} = N * (Sr + Sp + Sc) - N * (2 * Sa + Sm)$ Let $Sr, Sp = 0.1 \text{ KB}$ $Sa = 10 \text{ KB}$ $S(Sc) = 9.86 * Sc - 198$

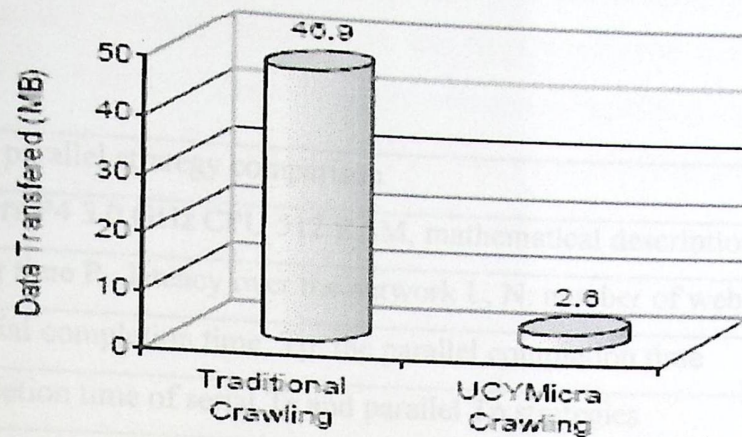
The result Figure 6.1.	 <table border="1" data-bbox="495 183 1587 619"> <caption>Data points estimated from Figure 6.1</caption> <thead> <tr> <th>Contents' sizes(KB)</th> <th>Saved Load(KB)</th> </tr> </thead> <tbody> <tr><td>50000</td><td>550000</td></tr> <tr><td>40000</td><td>450000</td></tr> <tr><td>30000</td><td>350000</td></tr> <tr><td>20000</td><td>250000</td></tr> <tr><td>15000</td><td>180000</td></tr> <tr><td>10000</td><td>120000</td></tr> <tr><td>8000</td><td>100000</td></tr> <tr><td>6000</td><td>80000</td></tr> <tr><td>4000</td><td>60000</td></tr> <tr><td>2000</td><td>40000</td></tr> </tbody> </table>	Contents' sizes(KB)	Saved Load(KB)	50000	550000	40000	450000	30000	350000	20000	250000	15000	180000	10000	120000	8000	100000	6000	80000	4000	60000	2000	40000
Contents' sizes(KB)	Saved Load(KB)																						
50000	550000																						
40000	450000																						
30000	350000																						
20000	250000																						
15000	180000																						
10000	120000																						
8000	100000																						
6000	80000																						
4000	60000																						
2000	40000																						
conclusion	We can conclude that the relation between the saved Load over the network and the different sizes of the web servers' contents is approximately linear so the mobile crawler will save the bandwidth in an efficient way in proportional with the size of contents and that's what we desire as specially as with the growing nature of the web. We conclude that the difference between the mathematical description and real estimates via simulation is neglected deviation.																						

6.1.2. The second experiment.

Is a comparison oriented experiment which takes two approaches of crawling in a view of two mirrors, in other words the two approaches are the traditional and multi agent strategies while the two mirrors are UCYmicra (i.e. related work used mobile agent) and AWCMA system respectively.

Name	Data transferred Comparison						
Tools	4 computers, mathematical description.						
Input	Data size average of 10 web servers, result message size average.						
Symbols	TDs: transferred data size. Sa : size of agent code. Sm: size of the result message.						
Output	TDs, comparison figure.						
Theory	While the use of mobile agents avoid the transfer of all the web contents to the search engines side to be indexed, the problem of heavy load over the network still exist in some mobile crawlers as they carry additional data they can compensate at home with self intelligent description of the content. Again the transferred data in AWCMA is: $TDs = N * (3 * Sa + 2 * Sm)$ Notice here we included the size of performance Tester agent and its result message which holds the performance measurements. ** we took the average of 10 web servers to give 55.5 MB and 2.07 MB						
The result Figure 6.2.	 A 3D bar chart comparing data transferred (MB) for two methods: Traditional crawling and AWCMA. The vertical axis is labeled 'data transfered (MB)' and ranges from 0 to 60. The horizontal axis shows two categories. The bar for 'Traditional crawling' is blue and has a value of 55.5 MB. The bar for 'AWCMA' is white with a black outline and has a value of 2.07 MB. A legend on the right identifies the bars: a blue square for 'Traditioanl craw ling' and a white square for 'AWCMA'. <table border="1"> <thead> <tr> <th>Method</th> <th>Data Transfered (MB)</th> </tr> </thead> <tbody> <tr> <td>Traditional crawling</td> <td>55.5</td> </tr> <tr> <td>AWCMA</td> <td>2.07</td> </tr> </tbody> </table>	Method	Data Transfered (MB)	Traditional crawling	55.5	AWCMA	2.07
Method	Data Transfered (MB)						
Traditional crawling	55.5						
AWCMA	2.07						

The result
Figure 6.3.



conclusion

The UCYmicro crawling comparison shows that they reduced the data transferred 18 times over traditional crawling methods. Well we made the same comparison and showed better result which is approximately 27 times over traditional crawling methods.

You may ask how and why? We can answer you simply that we extracted the most important information within the HTML documents in each web server (i.e. meta) and encapsulated these data in a light agent message rather than a carrier agent (i.e. UCYmicro). we introduced the idea of "*Transfer the Seed and Generate the Trees*" *TSGT*, which enables us to extract only the most important and process the retrieved data to generate a self intelligent description with synonyms for all keywords in the crawler manager side.

6.1.3. The third experiment.

Which took the serial and parallel strategy of crawling to compare between them in terms of time (i.e. performance), the initial requirements of this experiment is only two computers (i.e. web servers). And also the mathematical description.

Name	Serial and parallel strategy comparison																					
Tools	2 computers P4 3.0 GHz CPU 512 RAM, mathematical description.																					
Input	Processing time P , latency over the network L, N: number of web servers.																					
Symbols	Ts: the serial completion time, Tp: the parallel completion time																					
Output	The completion time of serial Ts and parallel Tp strategies																					
Theory	The serial strategy is claimed to have a less load on the network as it employs only one agent for crawling while in parallel we need N agents for N web servers, on the other hand the time needed for the former is more than the later, the following equation computes the completion time of serial strategy: $T_s = \sum_i P_i + \sum_j L_j : i \in \{0-N\} , j \in \{0, N+1\} \dots (1)$ $T_p = 2 * (L)_{\max} + (P)_{\max} \dots (2)$ Coming to the processing time evaluation which can be derived as follow: $P \propto D_s \rightarrow P = C * D_s \text{ where } R: \text{ constant depends on the process power of computers (i.e. CPU performance).}$ Substituting P in equation 1 $T_s = R * \sum_i D_{s_i} + \sum_j L_j$ And in equation 2 $T_p = 2 * (L)_{\max} + R * (D_s)_{\max}$																					
The result Figure 6.4.	<table><thead><tr><th>Content Size (MB)</th><th>Parallel Completion Time (sec)</th><th>Serial Completion Time (sec)</th></tr></thead><tbody><tr><td>60</td><td>15</td><td>25</td></tr><tr><td>50</td><td>10</td><td>20</td></tr><tr><td>40</td><td>8</td><td>15</td></tr><tr><td>30</td><td>6</td><td>10</td></tr><tr><td>20</td><td>4</td><td>7</td></tr><tr><td>10</td><td>3</td><td>4</td></tr></tbody></table>	Content Size (MB)	Parallel Completion Time (sec)	Serial Completion Time (sec)	60	15	25	50	10	20	40	8	15	30	6	10	20	4	7	10	3	4
Content Size (MB)	Parallel Completion Time (sec)	Serial Completion Time (sec)																				
60	15	25																				
50	10	20																				
40	8	15																				
30	6	10																				
20	4	7																				
10	3	4																				

conclusion	We conclude that the parallel strategy took less completion time than the serial which (i.e. the former) depends on the largest content size to have already completed while the later depends on the cumulative content sizes. We conclude also that R is constant relative to a specific host performance in other words R will differ depending on the host upon which it is being crawled or the performance case at a specific time range. You will say that the serial is more efficient in terms of load over the network because it uses only one agent while the parallel is much less, this is true in case that we dispatch N agents for N web servers, while in our system parallel strategy took a new direction that is the administrator (i.e. the user of the application) will choose uniquely the web servers to be crawled; so we don't have to dispatch N agents for N web servers.
-------------------	--

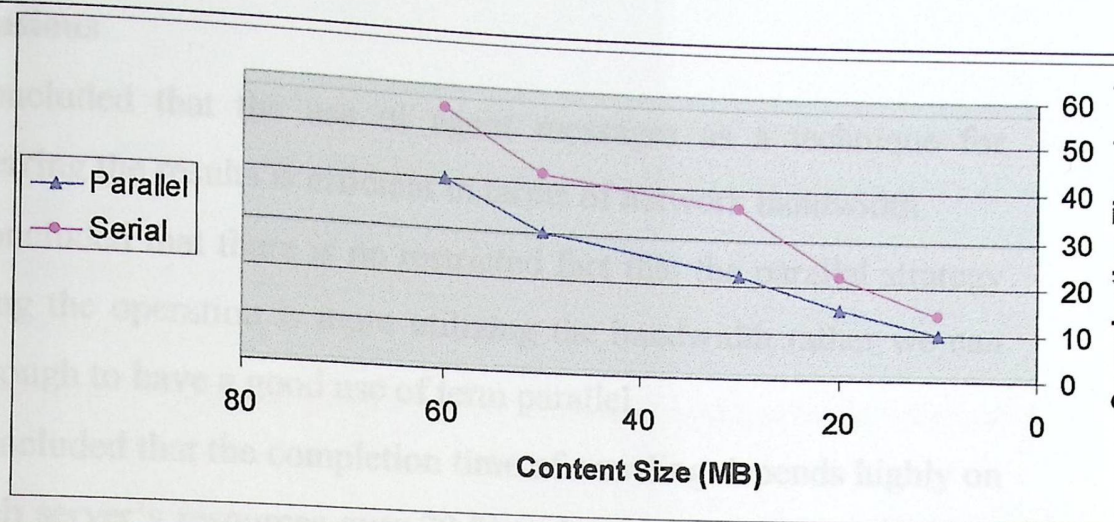
** we didn't include the network latency as it can be neglected within local area network.

6.1.4. The fourth experiment.

We decided to investigate the performance of both strategies on different computers with different properties in terms of CPU and Memory. The differences were applied on both the crawler manager and web servers respectively the following table will demonstrate in more details.

Name	Serial and parallel strategy comparison
Tools	2 computers 1.5 GHz CPU 256 RAM, mathematical description.
Input	Processing time P , latency over the network L, N: number of web servers.
Symbols	Ts: the serial completion time, Tp: the parallel completion time
Output	The completion time of serial Ts and parallel Tp strategies
Theory	The serial strategy is claimed to have a less load on the network as it employs only one agent for crawling while in parallel we need N agents for N web servers, on the other hand the time needed for the former is more than the later, the following equation computes the completion time of serial strategy: $T_s = \sum_i P_i + \sum_j L_j : i \in \{0-N\} , j \in \{0, N+1\} \dots (1)$ $T_p = 2 * (L)_{\max} + (P)_{\max} \dots (2)$ Coming to the processing time evaluation which can be derived as follow: $P \propto D_s \rightarrow P = C * D_s \text{ where } C: \text{ constant depends on the process power of computers (i.e. CPU performance).}$ Substituting P in equation 1 $T_s = R * \sum_i D_{s_i} + \sum_j L_j$ And in equation 2 $T_p = 2 * (L)_{\max} + R * (D_s)_{\max}$

The result
Figure 6.5.



conclusion

We concluded that the performance which is acted in the completion time in our experiments is affected directly based on the computers used for both the crawler manager and web servers.

We concluded also that when changing the power capacity of web servers, the size of effects is more than when changing the power capacity of crawler manager that is because the whole process done at the web servers.

6.2. Conclusions

- we concluded that the use of agent messages as a technique for transferring the results is efficient in terms of network bandwidth.
- We concluded that there is no restricted fact that the parallel strategy in doing the operation is more utilizing the bandwidth rather we can get through to have a good use of term parallel.
- We concluded that the completion time of crawling depends highly on the web server's resources over 70 % and crawler manager resources over 25 %.
- We concluded that parallel strategy took less completion time than serial with a tangible difference.

6.3. Future work

We are impassioned to analyze and process the Arabic web servers' contents arrived to the index and use the Arabic Morphology skills to have a smart Arabic index toward an enhanced the existence of Arabic contents on the web.

Also we are interested in upgrading a new more efficient algorithm in crawling to minimize the centralized paradigm effects.

We are interested also to design and implement two advanced modules for security and fault tolerance to have a universally adopted system that can be easily used instead of traditional crawling methods.

6.4. Challenges.

In this period of project which was prepared for gaining results to show how much we achieved our goals, it was very difficult and challenging.

When we used JADE we knew in advance that if we want to use it among different networks we have to use add-ons packages beside having a real IP for the crawler manager computer and thanks to *Eng. Ghnam al Gaabari* who granted us the IP 195.31.188.19 for the crawler manager.

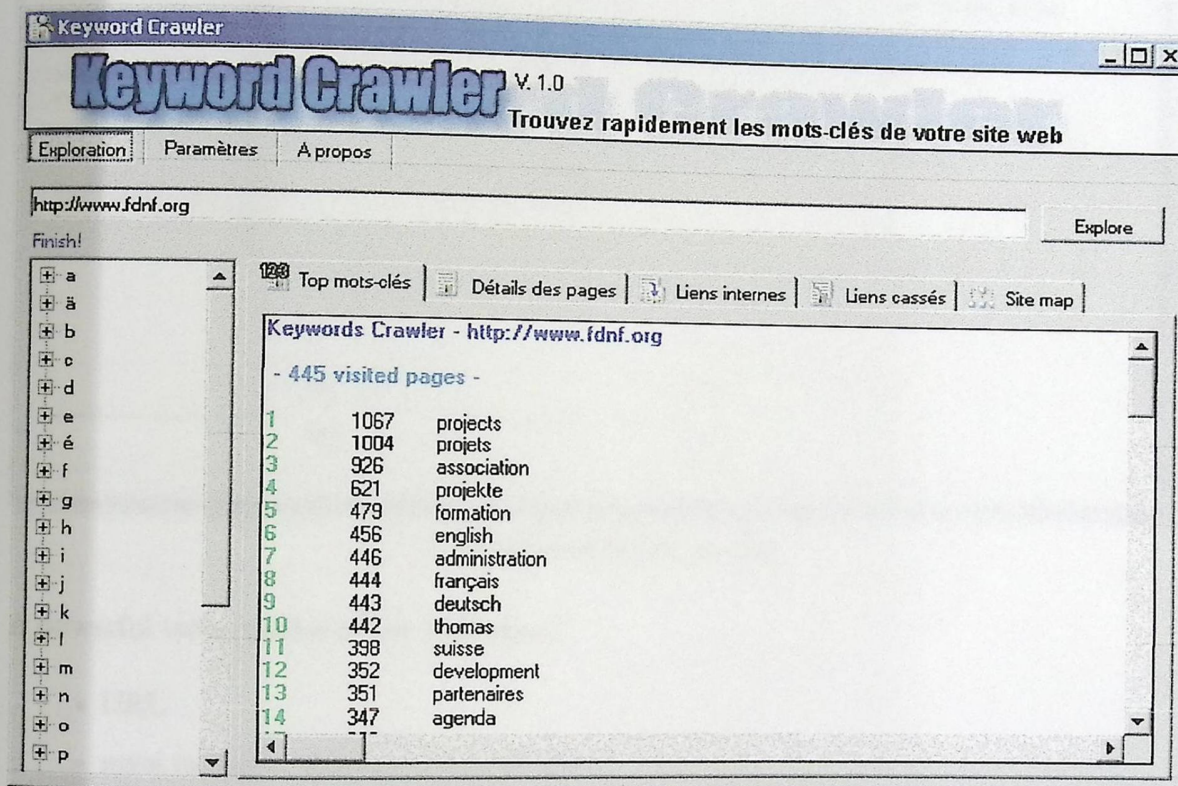
Another challenge faced us which is the difficulty of testing the freshness of the web page in real values that was because we don't have real and commercial web servers to have updated contents.

APPENDIX A

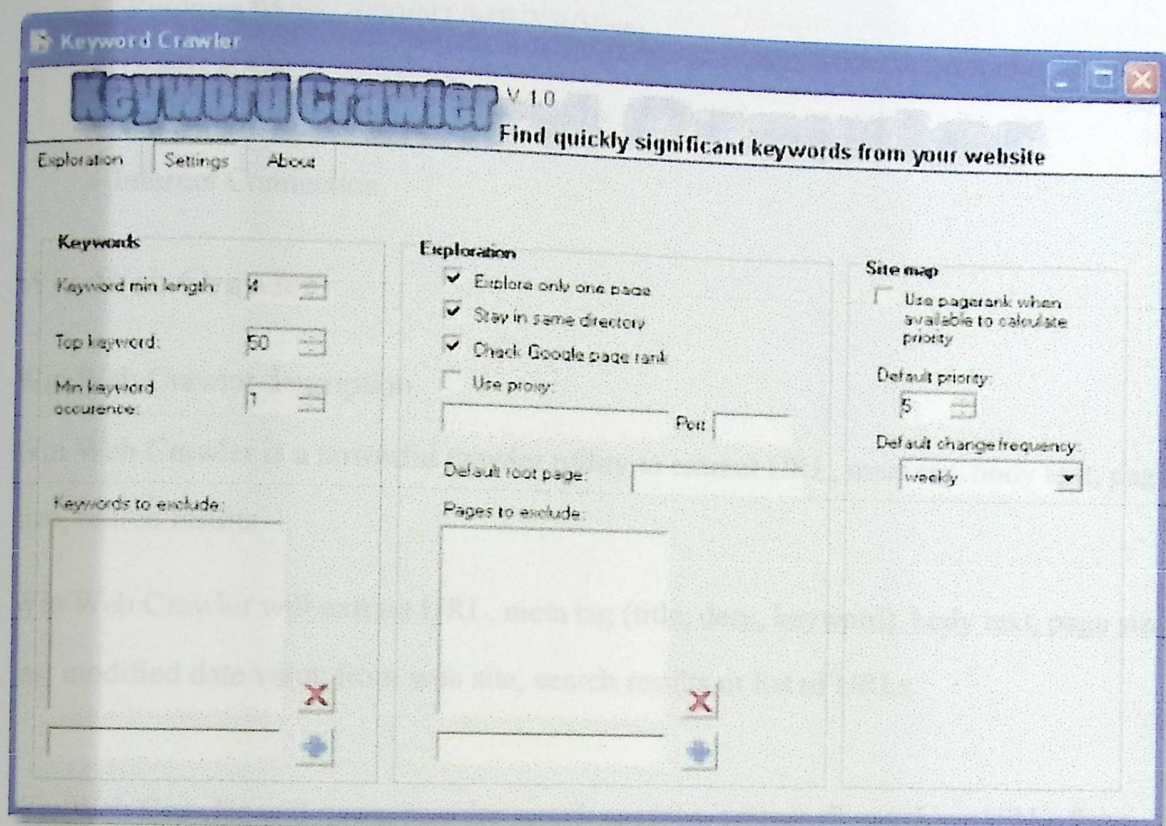
Survey and Ethics of Use

Keyword Crawler

It is an Online website optimization tool which enables you to explore and see your site like webengines. Count and display broken and internal links. Get Google PageRank for each file and generate a SiteMaps file in XML format. Keyword Crawler could also calculate word density in pages and display the metas and observes the directives of the Robots.txt and 'robot' meta tags



A.1 a snap shot from keyword crawler during process



A.2 configuring the crawler

A powerful web crawler utility to Extract:

- URL
- meta tag (title, description, keyword)
- plain text between <body> to </body> tag
- page size
- last modified date value

High speed, multi-threaded, accurate extraction - directly saves data to disk file. Program has numerous filters to restrict session, like - URL filter, text filter, data filter, domain filter, date modified, etc. It allows user-selectable recursion levels, retrieval threads, timeout, proxy support and many other options. A must have tool for webmasters who build Search Directories.

System Requirements:

- Windows 95/98/2000/NT/ME/XP/Vista
- 32 MB RAM
- 1 MB Hard Disk Space
- Internet Connection

Win Web Crawler

Win Web Crawler description

Win Web Crawler is a powerful crawler utility to extract URL, meta tag, body text, page size, search results

Win Web Crawler will extract URL, meta tag (title, desc, keyword), body text, page size, last modified date value from web site, search results or list of URLs.

Win Web Crawler will query popular search engines, extract all matching URLs from search results, remove duplicate URLs and finally visits those websites and extract data from there.

You can tell Win Web Crawler how many search engines to use. Click "Engines" button and uncheck

listing that you do not want to use. You can add other engine sources as well.

Win Web Crawler send queries to search engines to get matching website URLs. Next it visits those matching websites for data extraction. How many deep it spiders in the marching websites depends on Depth setting of "External Site" tab.

High speed, multi-threaded, accurate extraction - directly saves data to disk file. Program has numerous filters to restrict session, like - URL filter, text filter, data filter, domain filter, date modified, etc.

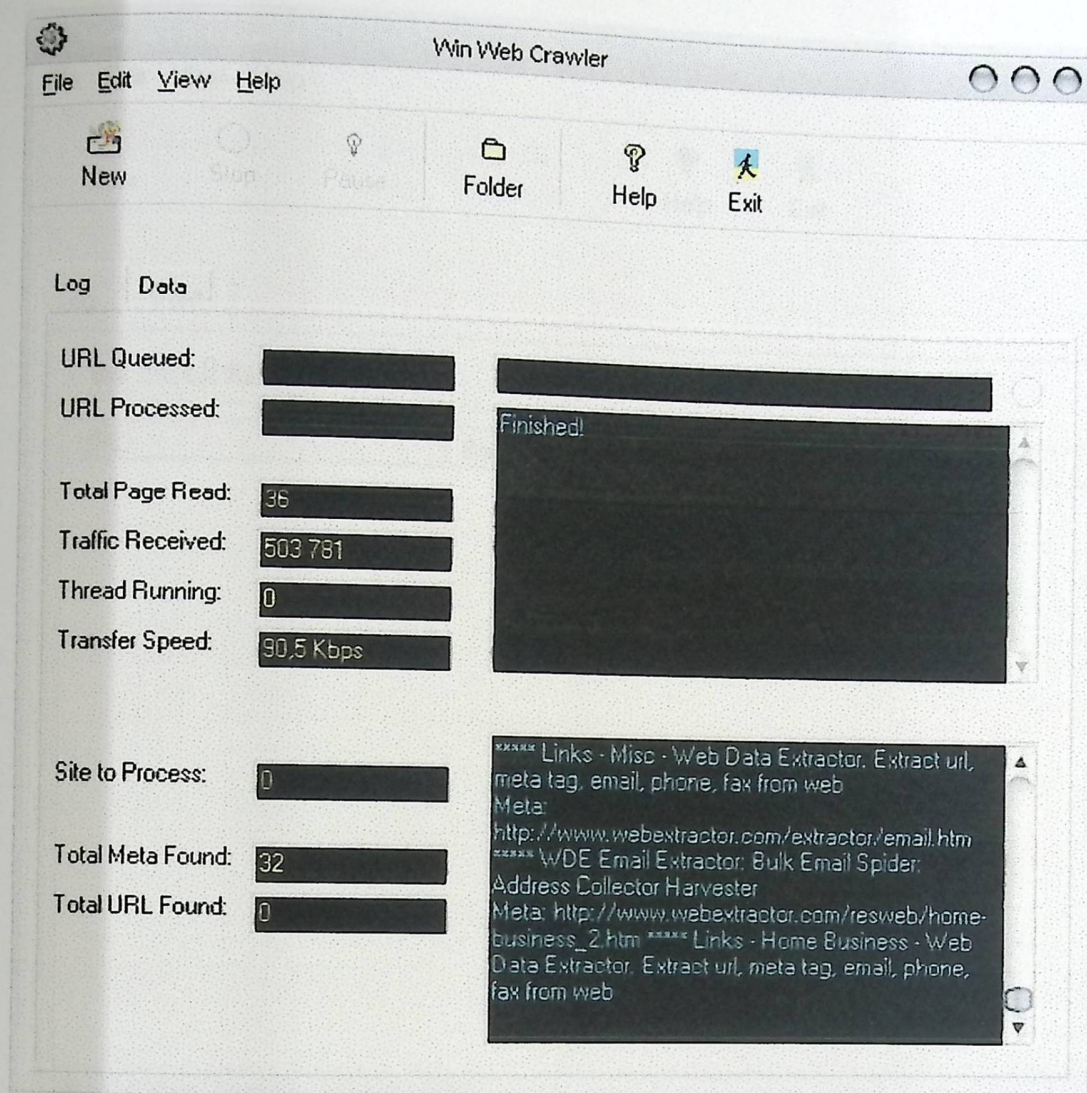
Win Web Crawler allows user-selectable recursion levels, retrieval threads, timeout, proxy support and many other options.

Requirements:

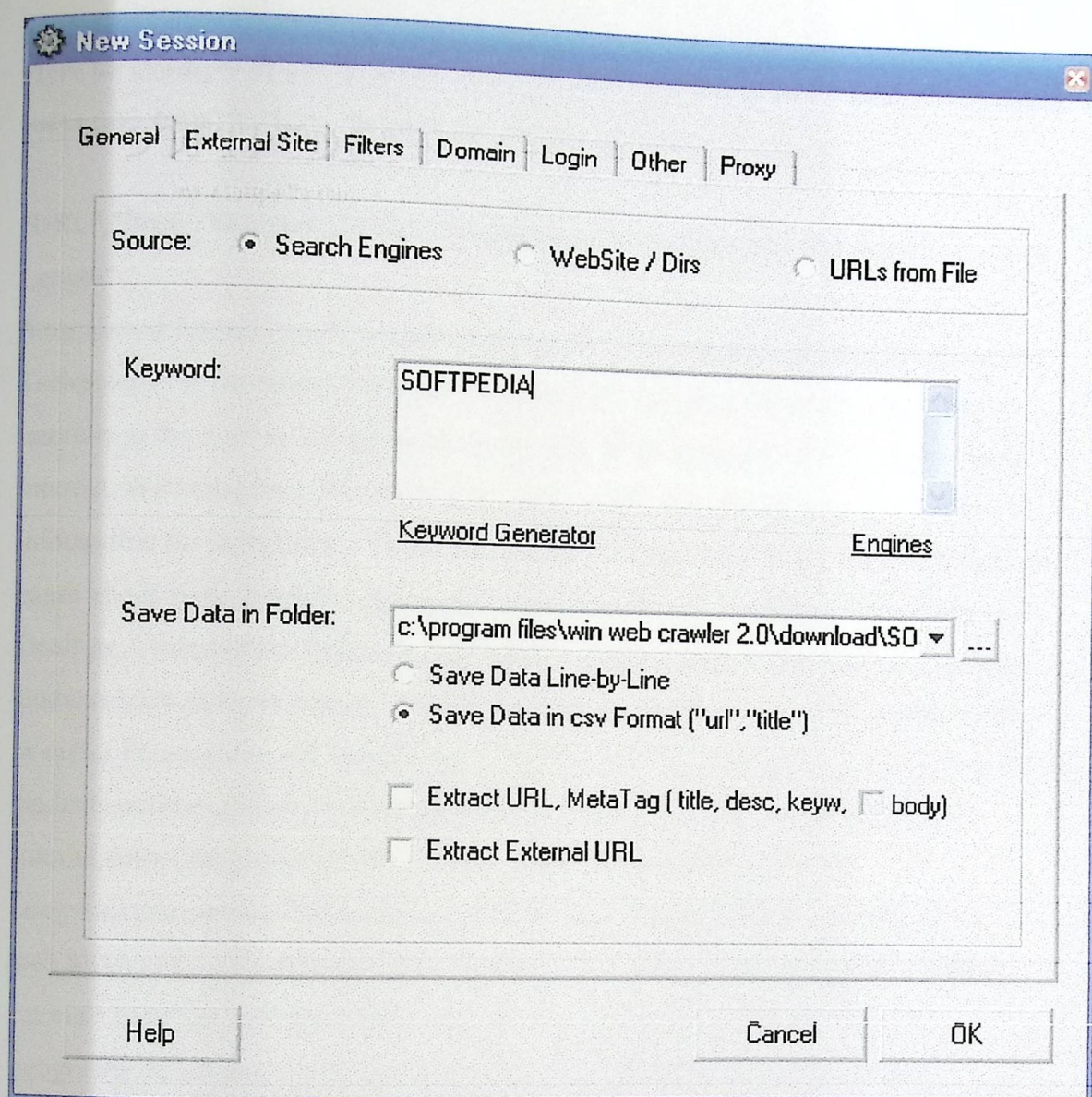
- 32 MB RAM
- 1 MB Hard Disk Space
- Internet Connection

Limitations:

- 15 days trial



A.3 Win Web crawler ..running



A.5 Win Web crawler ..session configurations

Here an example of win web crawler on www.hitmill.com and how they extracted meta tags from the web site then we put our output to compare

"URL","Base","Domain","Title","Description","Keyword","Last Modified","Content Length"

"http://www.hitmill.com/","hitmill.com",".com","Hitmill.com - Internet Training and Technology for Students","Internet Technology Training for Students has original tutorials in the field of computer programming, Web pages and HTML, networking, Internet, Web Graphics, Web Services, XML, ASP, Web technologies as well as information for developers. There is a college courses resource site now with links to pages about math, English language, California History, California Geography and Geology, Comparative Religions, and more... There are pages interesting to college students such as news, sports, weather and page of interest to all, such as the Severe Weather Preparation and Safety Tips.", "internet training, technology, students, internet for students, internet primer, internet tutorial, beginning programming, programming tutorial, computer programming, kerberos, network security, macintosh computer, mac, networking, computer networking tutorials, browsers, history of the web, history of computers, simula language, simula, qbasic programming, photoshop tutorials, html tutorials, math links, english links, california geology, california geography, seismology, sports, css, javascript, asp, web services tutorial, apache server, linux, introduction, primer", "", "12264"

"http://www.hitmill.com/web_graphics/fabricText.html","hitmill.com",".com","How To Fill Text With A Fabric","Filling text with a scanned fabric", "how, to, fill, text, with, fabric, an, image, how to, text-fill, text fill, scanner, scanned, fabric, image, photoshop, images, picture, tutorial, primer, lesson", "", "5398"

"http://www.hitmill.com/web_graphics/letterfills.html","hitmill.com",".com","PhotoShop Tutorial: Pattern Fills for Lettering","PhotoShop Tutorial: Pattern-fills for lettering explains how to fill letters with texture and pattern-fills made with Adobe

PhotoShop.", "photoshop tutorial, pattern-fill, free, online, tutorial, pattern fills for lettering, hitmill", "", "3217"

"http://www.hitmill.com/about.html", "hitmill.com", ".com", "Hitmill.com - About HiTMILL.com", "Internet technology training and tutorials for students and developers; computer networking topics, programming resources and tutoring; web technology tutoring, training and resources for students; html training, tutoring and resources and much more. See home page at www.hitmill.com.", "about hitmill, it training, students, developers, computer, internet, technology, training, students, web technology, programing, networking, news, weather, california history, free, online, hitmill, about", "", "6658"

"http://www.hitmill.com/computers/hoax.html", "hitmill.com", ".com", "Hitmill.com - Computer Hoax Tutorial", "This computer hoax tutorial will help you to not perpetuate hoaxes by forwarding them in e-mail to anyone else without investigating the hoax threat to find out if it is on the hoax list.", "e-mail, hoax, investigation, primer, tutorial, computer hoax tutorial, hoax list, computer hoaxes, web hoax, e-mail hoax, hitmill", "", "6306"

"http://www.hitmill.com/computers/index.html", "hitmill.com", ".com", "Hitmill.com - Computers, Computer Services", "Information for students about computers, computer processing and where to obtain driver updates, device driver or software downloads, operating systems, Apple and Macintosh computer information, Microsoft links, computer-related magazines, and more. Easy to navigate and full of information technology topics, history of computers, history of the Web, links to other Hitmill tutorials about Programming, Networking, WLANs, HTML, DHTML, CSS and additional Web technologies.", "computer services, where to find device drives, computer tutorials, computer operating systems, apache server resources, history of the web, history of computers, computers, introduction to computers, device drivers, find, computer hoax tutorial, computer hoax, e-mail hoax, virus hoax, hoax vs virus, stop hoax now, internet hoax, apple computer, macintosh computer, apple and mac, os x, bandwidth test, linux, monitor resolution, download software, tu cows, nonags, maps, news, travel, driver, update, networking, email, browsers, hitmill", "", "9985"

"http://www.hitmill.com/computers/intro.html", "hitmill.com", ".com", "Introduction to Computers", "", "introduction to computers, operations of a computer, computer introduction tutorial, hitmill, central processing unit, computer operations", "", "8504"
"http://www.hitmill.com/internet/index.html", "hitmill.com", ".com", "Hitmill.com - Introduction to the Internet", "This is a free online Internet tutorial and introduction to the Internet for students. This page introduces students to the components of the Internet, such as e-mail, browsers, the Web also known as the World Wide Web, how to search the Web, and a link to the section about HTML or Hypertext Markup Language used to make Web pages. The Internet is a world-wide network of computer networks joined together by communication lines, wireless connections and satellite connections. The Internet is not owned by any government nor individual nor corporation.", "introduction to the internet, internet primer, internet tutorial, hitmill, introduction to email, introduction to browsers, search tutorial, history of the web, history of computers, history of the internet, what is the internet", "", "5544"

"http://www.hitmill.com/internet/browsers.html", "hitmill.com", ".com", "Hitmill.com - Web Browser Tutorial for Students", "A Web browser tutorial with discussion about Web browsers, types of browsers, Google's new Chrome browser, Mozilla.com's Firefox3 browser, Microsoft's Internet Explorer 8 Beta, Mosaic, Opera, Lynx, and the Amaya editor/browser of the World Wide Web Consortium... and related links.", "web browsers, web browsers tutorial, browser article, web browser tutorial, about browsers, chrome browser, chrome, google chrome, firefox, firefox3, lynx viewer, ncsa mosaic, mosaic browser, internet browser, web accessibility, browser compatibility, speech enabled browser, amaya, w3c, lynx browser, msie, microsoft internet explorer, netscape navigator, ncsa mosaic, hitmill, opera browser, add-ons, ie, msie, browser plug-ins, opera, ie7, ie8, web browser", "", "8501"

"http://www.hitmill.com/html/promote/search.htm", "hitmill.com", ".com", "Hitmill.com - Search Engine Tutorial", "This is a free online search engine tutorial for students and includes what is a search engine, where to find search information, how to search, specialized searches, usenet news searches, meta search, Web search, gopher search, music search, how to use search tools, find music and mp3 files, how search engines work, how they find your Web page, how SEs rank pages, and even search engine ratings

and search engine information for Web masters. In no time at all you will be searching the Web, the Internet and learning how to find specific search information on the Web.", "search engine tutorial, what is a search engine, use search tools, search the internet, web, search, internet, where to find search information, education search, music search, game search, gopher search, searching the web, web search, search, usenet news, meta tag, tutorial, hitmill, understanding internet, search tools, music, mp3, napster, find, searching, tools, gopher, software, rating, page rank", "", "8504"

"http://www.hitmill.com/web_graphics/patternFills.html", "hitmill.com", ".com", "Hitmill.com - PhotoShop Tutorial: Fill Text With Pattern-Fill", "PhotoShop Tutorial: Filling text with a pattern-fill.", "photoshop tutorial, fill text with pattern-fill, hitmill, pattern-fill lettering, adobe creative suite 4", "", "5784"

"http://www.hitmill.com/web_graphics/lettering.html", "hitmill.com", ".com", "Hitmill.com - PhotoShop Tutorial: Lettering and Gradient-Fills", "Free PhotoShop Tutorial: Discussion of PhotoShop lettering, and filling text with an image or with a gradient fill. PhotoShop 6 was used.", "photoshop tutorial, lettering and gradient fills, photoshop primer, lettering tutorial, photoshop 6, gradient-fill, lettering, text, image-fill, create text-fills from an image using photoshop 6, lettering tutorial, hitmill", "", "6889"

"http://www.hitmill.com/internet/url.html", "hitmill.com", ".com", "Uniform Resource Locator Tutorial: What is a URL?", "Uniform Resource Locator tutorial: What is a URL?", "what is a url, uniform resource locator tutorial, uniform resource indicator, hitmill, ip address, url, url tutorial, uri, internet address, web address", "", "9222"

"http://www.hitmill.com/college/eq2.htm", "hitmill.com", ".com", "Hitmill.com - Seismology - Earthquakes", "Seismology-Earthquake quick reference", "tectonics, earthquake, magnitude, map, shake map, california earthquake, seismograph, seismogram, earthquakes, report earthquake, canada, canadian, japan earthquake, world wide earthquakes, temblor, earth tremor, usgs, pacific rim, ring of fire, hitmill, geological hazards, fault zones, earthquake faults, seismicity, seismology, safety tips, earthquake preparation.", "", "13881"

"http://www.hitmill.com/internet/magazines.html", "hitmill.com", ".com", "Computer Magazines and E-zines", "...compiled directory of computer and network magazines include topics such as computer technology, networking, processors and hardware,

programming languages, robotics, visual basic, .NET technology, c++ programming, c# programming, xml magazines and more..", "online computer and internet magazines,online,internet,ezone,magazine,e-zine,directory,computer e-zine,computer ezine,it e-zine,it magazine,computer technology e-zine,networking magazine,processor magazine,robotics magazine,vb user magazine,c++ user magazine,.net magazine,hitmill,computers,programming languages,ieee,robotics,processor,networking,wan,wireless,wyfy,bluetooth,macintosh,mac c,apple,os x,ipod,imac,mac g5,online news,travel","", "25715"

"http://www.hitmill.com/internet/ftp.html", "hitmill.com", ".com", "Hitmill.com - FTP New User: An FTP Guide, Tutorial and Directory
for File Transfer Protocol", "This is an FTP guide and tutorial for new FTP users about File Transfer Protocol (FTP) and an FTP tutorial about how to move files such as HTML documents between one computer to another computer over the Internet, using WS_FTP Software. There is an FTP directory included of FTP-related links.", "ftp,file transfer protocol,ftp guide,ftp new user,ftp tutorial,ftp directory,what is ftp,introduction to file transfer protocol,introduction to ftp,how to ftp,download ftp,hitmill,ftp search,ftp for students", "", "18558"

"http://www.hitmill.com/internet/web_history.html", "hitmill.com", ".com", "History of the Web Beginning at CERN", "The History of the Web Beginning at CERN is an original research article by Cheryl Gribble with Bibliography and Related Links.", "history of the web beginning at cern,berners-lee,xanadu,as we were,hypertext,http,mosaic,robert calilliau,html,mark andreesen,jim clark,netscape corp,web statistics,browser history,history of browsers,web demographics,web history,history of the web,hitmill", "", "17747"

"http://www.hitmill.com/computers/apache.htm", "hitmill.com", ".com", "Apache Server Resources", "This is an Apache Server guide for users, developers, webmasters, and students as we include user/developer information, Apache forums, Apache FAQ, PHP server-side includes, HTACCESS file, Apache Server books, and link to the Apache Project, additional Apache HTTP Server resources, and a UNIX resource list.", "apache,apache resources,apache links,apache directory,apache http server,php,mysql,free,tutorial,apache server resources,hitmill,server-side

include,files,htaccess file,php/mysql,apache books,file,apache developer,http project,apache project,ssi file,unix reference","","15910"

"http://www.hitmill.com/computers/kerberos.html","hitmill.com",".com","Hitmill.com - Tutorial About Kerberos","This is a free, online tutorial about Kerberos, and reference for Computer Networking students, about the Kerberos authentication protocol which uses secret-key cryptography..this is a free kerbos tutorial for students new to computer networking classes.", "tutorial about kerberos,kerberos tutorial,what is kerberos,kerberos authentication,tutorial,computer networking class,primer,kerberos,reference,hitmill,free,online,secret-key,kerberos,single sign on technology,network,authentication,free,cryptography,computer networking,computer security,kerberos v5,ticket,management,log on,password,applications,viewing tickets,protocol","","14533"

"http://www.hitmill.com/web_graphics/moreFabrics.html","hitmill.com",".com","PhotoShop Tutorial: More Fabric Fills","PhotoShop Tutorial: Filling text with a scanned fabric","photoshop tutorial,filling text with a scanned fabric,photoshop primer,more fabric fills,hitmill,how to fill text with scanned image","","3751"

"http://www.hitmill.com/phot/cgribble.html","hitmill.com",".com","Hitmill.com - Photography of Cheryl Gribble","Introduction to Photography, or Photo120 at Napa Valley College is an introductory level photography course about digital cameras, digital photography, and how to use a Digital SLR Camera.", "introduction to photography,digital cameras,history of photography,hitmill,intro phot,photo120,napa valley links","","5250"

"http://www.hitmill.com/web_graphics/califPoppies.html","hitmill.com",".com","Hitmill.com - Create An Image-Fill for Text Using PhotoShop","This is a free online PhotoShop tutorial about how to use image fills for text, putting an image into lettering, text image fill, font fills, text with image.", "image fill,lettering,font,text,photoShop tutorial,hitmill,put image into lettering","","10735"

"http://www.hitmill.com/college/college_index.html","hitmill.com",".com","College Course Resources and References","College course resources, college Preparatory classes, resources for math courses, Comparative Religion, PhotoShop Graphics, Ancient Civilizations, English, Computer Programming, Visual Basic and C++ Programming,

History of Computers, California History, California Indians, California Geography, and more...","college course resources and references,comparative religion,hitmill,online,free,computer programming,seismology,english,math,math links,math directory,california history,california geography,computer programming,visual basic,vb,c++,6,ancient history and civilizations,sports,travel,weather,california indians,photoshop,california geography,california history,history of computers,computer history,math courses","","6091"

"http://www.hitmill.com/internet/news.html","hitmill.com",".com","Hitmill.com - Find News","Internet search tool for finding local news, state news, national and international news, sports, weather and travel information.", "news resources,find news,local news,national news,world news,drudge report,local news, state newspapers, state news,international news,state,national,world,international,news,weather,sports,hitmill","","8897"

"http://www.hitmill.com/computers/computerhxl.html","hitmill.com",".com","History of Computers","The development of the modern day computer was the result of advances in technologies and man's need to quantify. The abacus was one of the first counting machines. Papyrus helped early man to record language and numbers. Some of the earlier counting machines lacked the technology to make the design work. For instance, some had parts made of wood prior to metal manipulation and manufacturing. Imagine the wear on wooden gears. This history of computers site includes the names of early pioneers of math and computing and links to relating sites about the History of Computers for further study. This site would be a good Web adjunct to accompany any book on the History of Computers or Introduction to Computers. The ","history of computers,history of computing,computer historytimeline,history of the web,hitmill,pioneers,pioneers of computing,pascal,babbage,computer pioneers,an open letter to hobbyists,boole,al gore,lovelace,gottfried leibnitz,george boole,konrad zuse,john l. von neumann,abacus, colusus,alan turing,grace hopper,robert e. kahn,John vincent atanasoff,atanasoff-berry,vannevar bush,edsgar wybe dijkstra,robert m. metcalf,shortest path algorithm,j presper eckert,john mauchly,univac,as we may think,bill hewlett,dave packard,stibitz,ted nelson,steven wozniac,edgar codd,steve job,william h. gates,bill

gates,j presper eckert,john mauchley,internet explorer,j.c.r. licklider,leonard kleinrock,dennis ritchie,xanadu,tcp/ip,kristen nygaard,blaise pascal,pdp computer,dennis ritchie,george stibitz,system r,an wang","", "37924"

"http://www.hitmill.com/college/english.html","hitmill.com",".com","Hitmill.com - English Course Resources and References","Free online resources and references for English and Literature courses including English Dictionary, English Thesaurus, English links, and English as a second language. See tutorials, newsgroups and more...","english,reference,resource,class,course,online,free,english references,literature reference,english course resources and references,english dictionary,english thesaurus,english grammar,english style,english reading,english literature,english as second language,esl,mla style,hitmill,references for english course,english course links,english class,directory of english links,english resources,english tutorial,mla style,hitmill","", "14405"

"http://www.hitmill.com/internet/maps.html","hitmill.com",".com","Hitmill.com - Maps and Atlases","A categorical directory of maps,navigation, globes, and other travel aids, including route maps, aviation maps, weather maps, topographical maps, and navigation maps. Maps of countries are included and links for street maps.","globe,world,map,map index,route maps,street map,atlas,maps and atlases,us map,canada,australia,japan,earthquake map,great britain,asia,china,switzerland,railroad,river,ocean,aviation map,weather map,navigation,trip,road map,uk,europe,spain,mexico,south america,north america,historical,hitmill,old map","", "14428"

"http://www.hitmill.com/college/math.html","hitmill.com",".com","Chalk Board Math Resources for Students","Chalk Board is a Mathematics directory site with resources for students and educators.","hitmill,mathematics,research,resources,women mathematicians,history of mathematics,algebra tutorials,geometry tutorials,pre-calculus,tutorial,precalculus,calculus tutorial,abstract algebra tutorials,miscellaneous math links,math links for students and educations,teaching math,algebra,pre-algebra,geometry theorems,geometry,calculus,teaching calculus,careers in mathematics,chalk board mathematics,women in mathematics,women in science,math resources,math homework help","", "12914"

"http://www.hitmill.com/internet/sports.html", "hitmill.com", ".com", "Chalk Line", "Chalk Line is a directory of sports news, scores and resources ... for polo, gymnastics, NFL, equestrian, baseball, football, fantasy football, baseball scores, and more...", "chalk line, fantasy football, nfl football, football, afc, nfc, football scores, news, sports scores, chalk line, football

scores, sporting, sport, sports, score, scoreboard, polo, gymnastics, nfl, butkus, equestrian, equestrian polo, baseball scores, baseball, hitmill, football news, football injuries, baseball news", "", "10028"

"http://www.hitmill.com/html/rgbcolorvalues.html", "hitmill.com", ".com", "Hitmill.com - Hexadecimal Color Values Reference", "Here are six pages of color reference for hexadecimal color values and a colour scheme selector tool where you can select one color or see a whole color scheme and how the colors would look together. Each page has the hexadecimal color value number with each color and there are several tables of colors and corresponding hexadecimal values. Follow the links to get the full benefit of this color reference. Use the scheme selector tool to convert hexadecimal values to rgb numbers.", "hexadecimal color code reference, hexadecimal color values reference, color reference, colour reference, rgb color values, rgb values list, hex

color, hitmill, colour, scheme selector, convert hexadecimal to rgb", "", "5143"

"http://www.hitmill.com/internet/weather.html", "hitmill.com", ".com", "Hitmill.com - A Weather Watch", "A free compiled online directory of weather links for the United States, Canada and North America, including marine, aviation, and noaa weather links, road and weather conditions, state weather, ski conditions, marine weather, aviation weather, road and weather current conditions, and links to other online weather pages.", "uk, united kingdom, uk coast, uk coastal forecasts, weather, road conditions, online, california roads and weather, quebec conditions, quebec weather, quebec forecast, alberta, british columbia, manitoba, quebec, new brunswick, nova scotia, nunavut, newfoundland, labrador, ontario, yukon, world, prince edward island, saskatchewan, northwest territories, current conditions and forecast, canadian weather forecast, canada current conditions, canada forecast, nevada weather and road conditions, ski conditions, roads and weather, weather and road conditions, california, nevada, arizona, oregon, current conditions, north dakota, colorado

roads and weather,colorado outdoors,noaa,colorado current conditions,hitmill,weather forecasts,weather warnings,noaa links,hitmill,manitoba conditions and forecast,current conditions,forecast,new brunswick,north america,usa,united states,,marine weather,aviation weather,road and weather conditions,severe weather,severe weather,preparation,safety tips,a weather watch,fleet weather,forecasts","","16709"
"http://www.hitmill.com/computers/fontdownload.html","hitmill.com",".com","Hitmill.com - Tutorial About Fonts"," Tutorial About Fonts: about finding a font and installing a font on your computer, or about making fonts.", "tutorial about fonts,how to install font,find font,install font,font tutorial,download font,laser chrome font,hitmill,font tutorial",","","6875"

"http://www.hitmill.com/html/index.html","hitmill.com",".com","Hitmill.com - HTML Tutorials for Students","HTML Tutorials for Students is HTML for beginners, a primary HTML site with new tutorials and resources for formatting text on a Web page, creating tables, how to add an image to a Web page, hexadecimal color values, FTP tutorial, PDF File tutorial, free images, and more...", "html tutorials for students,html tutorial,html for beginners,html primer,web page tutorial,free,online,html table tutorials,audio for Web pages,table tutorial,html table,html sound,free background textures,hex color,hex colour,web page sound,Web page music,meta tags,what is the web,history of computers,free backgrounds for Web pages,ftp tutorial,pdf files,free web space,optimizing for the search engines,web page promotion,history of the web,hexadecimal color values,hexadecimal, colour,colour-safe web palette,hex,hexidemical,fp add-ons,color-safe web palette,browsers,hitmill,html editors,what is html,advanced html,front page links,frontpage links,image fill,lettering,photoshop,clipart,photoshop lettering tutorial,hex color values,image in letter,web images,dynamic html,accept credit cards from your web site,hitmill,charge.com",","","11653"

"http://www.hitmill.com/dhtml/index.html","hitmill.com",".com","Hitmill.com - What is Dynamic HTML?","Dynamic HTML technologies are explained and explored. Tutorials are included in this site. Dynamic HTML is a group of technologies. See our definition and explanation of the technoligies compriising Dynamic HTML, or DHTML.", "what is dynamic html,what is dhtml,dhtml tutorial,dhtml primer,introduction to dhtml,dynamic

html tutorial,dynamic html primer,hitmill,introduction to dynamic
html,javascript,cascading style sheets,css-2,css2,css tutorial,javascript tutorials,cssp,css-
p","","7022"

"http://www.hitmill.com/programming/asp/aspnet.html","hitmill.com",".com","ASP and
ASP.NET Resources","ASP.NET resources for Web Development, products, forums,
free code sources, books -- resources for the ASP.NET development
community.", "asp.net,asp,web development,asp and
asp.net,resources,forums,books,hitmill,asp.net development community", "", "9904"

"http://www.hitmill.com/dhtml/css_intro.html","hitmill.com",".com","Introduction to
Cascading Style Sheets Rules","An introduction to Cascading Style Sheets rules for Web
pages.<META NAME=","free css tutorial:introduction to cascading style sheets rules,css
tutorial,cascading style sheets,css,css 1,css 2,css 3,css primer,cascading style sheets
tutorial,css3 working draft,css specification,hitmill", "", "17554"

"http://www.hitmill.com/internet/travel.html","hitmill.com",".com","Hitmill.com - Travel
Corner Travel Directory","This is a travel directory about travel topics from hitmill.com's
Travel Corner and includes topics and items needed by travelers including reservations
for travel, airline tickets, car rentals, hotel, motel, time zones, postal codes, TV listings,
cruises, trains, vacation getaways, maps, weather, and more.", "travel
corner,travel,hitmill,airfares,ticket reservation,airline ticket,air ticket,train
ticket,hotel,motel,reservation,car rental,car hire,hire car,car reservation,low cost air
fare,air fare,cheap tickets,language dictionaries,plane ticket,plane reservation,airline
reservation,train reservation,money exchanges,travel guides,tv guide,us
postage,maps,reservations", "", "26993"

"http://www.hitmill.com/html/editors.html","hitmill.com",".com","HTML
Editors","HTML Editor software programs are discussed. Links are given for the home
pages of the HTML Editor programs for both PCs and Mac computers.", "html editors are
discussed,links to home sites of the html editors,html coding with html editor,html
editor,frontpage 2003,taco html editor,hotdog html editor,golive html editor,microsoft
frontpage 2003,microsoft frontpage 2002,hitmill", "", "12273"

"http://www.hitmill.com/programming/vbs/vbscript.html","hitmill.com",".com","Hitmill.
com - VBScript For Beginners","Free online VBScript tutorial about implementing

server-side scripts, references for VBScript, and where to find VBScript code examples.", "vbscript for beginners,vbscript

tutorials,free,online,vbs,resources,vbscript,vbscript code,vbscript references,hitmill", "", "6757"

"http://www.hitmill.com/programming/js/javascript.html", "hitmill.com", ".com", "Free Online JavaScript Tutorials: JavaScript for Web Pages and Client-Side Web Coding", "This page offers free online JavaScript tutorials and is a primary JavaScript site for students and developers with special emphasis on learning to implement JavaScript for Web pages, or precisely JavaScript for client-side Web code; tutorials, tips and resources are included along with code examples for students. We recommended some prior knowledge of HTML markup language prior to learning JavaScript since many of the code examples are actual Web page applications of JavaScript client-side programming and ASP (Active Server Pages technology). Do not confuse JavaScript with Java.", "free online javascript tutorials,javascript for web pages,javascript tutorials,javascript primers,javascript,hitmill,client side programming,javascript button code,free,asp,active server pages,free javascript code,code examples,active server pagesaccept credit cards from your web site,hitmill ad for charge.com", "", "28071"

"http://www.hitmill.com/programming/ada.htm", "hitmill.com", ".com", "Ada Programming Language", "Collection of online resources for Ada Programming Language and Ada references. Page has moved to http://www.hitmill.com/programming/ada/ada.htm", "ada,programming,language,ada programming,ada95,hitmill,ada references,history of ada programming language", "", "3103"

"http://www.hitmill.com/programming/overview.html", "hitmill.com", ".com", "Programming Overview", "An overview of computer programming written for students .", "introduction to oop,introduction to computer programming,programming overview for students,computer programming,introduction to computer programming,programming primer,programming tutorial,overview of computer programming,hitmill", "", "9826"

"http://www.hitmill.com/webservices/index.htm", "hitmill.com", ".com", "Hitmill.com - Web Services Tutorial for Students", "Webservices for Students is a starting point to learning about this fairly new Web technology, along with the .NET technologies: C#,

ASP.NET, VB.NET, Visual C++.NET, VisualStudio.NET, and more.", "webservices tutorial, for students, web services tutorial, free, starting point, webservices, asp.net, c#, dotnet, hitmill", "", "15260"

"http://www.hitmill.com/html/namedcolors.html", "hitmill.com", ".com", "Hitmill.com - Hexadecimal Color Codes: Selected Named Colors", "Selected Named Colors is a reference for Web authors desiring to code the name of a color without using the alphanumeric code number for the color. When using a compounded name or a name with several parts do not use spaces between words, and use lowercase.", "hexadecimal color codes, hex color code, rgb color, html color code, hexadecimal colour code, hex colour code, html colour, hitmill color, hitmill, named colors, selected named colors, web authors reference, alphanumeric code colors, windows 3.2 colors", "", "26423"

"http://www.hitmill.com/xml/xml.html", "hitmill.com", ".com", "Hitmill.com - XML Resources", "XML resources including XML tutorials, XML references, resources, newsgroups, XML news, references and more including the XML schema, XML DOM, XSLT, and SVG.", "xml tutorials, xml resources, xml references, xml directory, for students, xml primer, free, online, hitmill, introduction to xml", "", "26206"

"http://www.hitmill.com/programming/index.html", "hitmill.com", ".com", "Hitmill.com - Programming Resources", "Computer Programming has primers and tutorials with code examples for students new to programming plus a compiled directory of programming resources and related topics for students, programmers and developers. Our site includes Ada, Simula, C, C++, Visual Basic 6, Visual Basic.NET, QBasic, Java, C#, .NET, ASP.NET; and also the scripting topics of JavaScript, VBScript, and ASP. There is an Introduction to Web Services for students. The markup languages such as HTML, XML, and XHTML have a separate page for each topic. (See links at bottom of page.)", "computer programming resources, computer programming tutorials for students, visual basic 6, vb6, programming resources for students, visual basic, visual basic.net, ada65, hitmill, overview of programming, programming languages, programming, simula, qbasic programming, visual basic programming, c++ for beginners, c# programming, c programming, history of c, history of c++, c++.net, .net framework, asp, vbscript, javascript programmng, java tutorials, qbasic, ada, simula", "", "17432"

"http://www.hitmill.com/programming/c.html", "hitmill.com", ".com", "Programming in C", "This is a collection of Web pages at this domain about the C Programming Language and includes topics such as the history of the C Programming Language, Primers and Tutorials, Newsgroups, References, Books and Ezines, Advanced C Programming and Products.", "programming in c,c programming,language,books,history of c,c programming history,c history,,history of the c programming language,c tutorial,c primer,c resources,c primer,c tutorial,c forum,c reference,news,c newsgroup", "", "10032"

"http://www.hitmill.com/programming/cpp/cpp.html", "hitmill.com", ".com", "hitmill.com - C++ Programming for the Beginning C++ Student", "A primary C++ site for the beginning c++ student; c++ programming for the beginning c++ student with tutorials for the c++ beginner, C++ Hello World tutorial and c++ help for students.", "c++ programming for the beginning c++ student,c++ beginner's tutorials,c++ code library,c++ help,hitmill", "", "5582"

"http://www.hitmill.com/programming/java.html", "hitmill.com", ".com", "Hitmill.com - Java Programming", "This is a compiled directory for Java Programming comprised of sections for Java Tutorials, Java Applets, Java articles, Java newsgroups, Java references, and Java books.", "java programming,free,online,java,tutorials,applets,java articles,java newsgroups,java reference,java primer,java book,free java applet,hitmill", "", "4494"

"http://www.hitmill.com/programming/realb/realb.htm", "hitmill.com", ".com", "REALbasic for Macintosh", "A programming environment for Mac developers.", "realbasic programming,realbasic,mac development,mac programming,environment,programming,hitmill", "", "5198"

"http://www.hitmill.com/html/promote/linkpartners.htm", "hitmill.com", ".com", "Hitmill.com - Link Partners", "Hitmill.com's reciprocal linking partners having computer-related technologies, businesses and services, of interest to students.<META NAME=", "hitmill.com,computer-related technologies, webpage,web design,reciprocal linking", "", "4754"

"http://www.hitmill.com/programming/qbasic.html", "hitmill.com", ".com", "Hitmill.com - QBasic Programming", "QBasic Programming resources with QBasic tutorials, QBasic chats, books, primers, textbooks, where to download QBasic, and compiler information

for students new to computer programming.", "qbasic, qbasic programming, qbasic tutorials, qbasic chats, qbasic books, qbasic primers, qbasic textbooks, download, 7.1, free, programming, language, languages, download, qbasic, hitmill", "", "16852"

"http://www.hitmill.com/programming/cpp/simula.html", "hitmill.com", ".com", "hitmill.com - Simula Programming Language", "Information and resources about the Simula programming language, from a historical perspective.", "exploring the simula programming language, simula, history of simula, simula 67, hitmill, simula programming, exploring simula, simula-67", "", "8516"

"http://www.hitmill.com/programming/vb.htm", "hitmill.com", ".com", "Hitmill.com - Visual Basic 6 Programming", "Free Visual Basic 6 tutorials, visual basic 6 beginner's page, visual basic 6 developer section, free visual basic 6 code, online newsgroups, books, maillists, free online vb6 tutorials for programming beginners and additional VB 6.0 programmer and developer resources.", "visual basic 6, programming, vb, vb6, visual basic, hitmill, resources, free online tutorial, vb6 programming beginner, visual basic, vb6, vb6 programmer, primer, visual basic tutorial, beginning programmer, advanced visual basic, activex, vb6 active x control, vb6 beginner, visual basic developer, vb6 developer, visual basic 6 beginner, vb beginner", "", "13425"

"http://www.hitmill.com/web_graphics/ps.html", "hitmill.com", ".com", "Photoshop References and Tutorials", "Free PhotoShop tutorials and references, Commercial Art links.", "photoshop, references, tutorials, primers, resources, directory, commercial art, hitmill", "", "11182"

"http://www.hitmill.com/index.htm", "hitmill.com", ".com", "Hitmill.com - Internet Training and Technology for Students", "Internet Technology Training for Students has original tutorials in the field of computer programming, Web pages and HTML, networking, Internet, Web Graphics, Web Services, XML, ASP, Web technologies as well as information for developers. There is a college courses resource site now with links to pages about math, English language, California History, California Geography and Geology, Comparative Religions, and more... There are pages interesting to college students such as news, sports, weather and page of interest to all, such as the Severe Weather Preparation and Safety Tips.", "internet training, technology, students, internet for

students,internet primer,internet tutorial,beginning programming,programming
tutorial,computer programming,kerberos,network security,macintosh
computer,mac,networking,computer networking tutorials,browsers,history of the
web,history of computers,simula language,simula,qbasic programming,photoshop
tutorials,html tutorials,math links,english links,california geology,california
geography,seismology,sports,css,javascript,asp,web services tutorial,apache
server,linux,introduction,primer", "", "12264"

"http://www.hitmill.com/programming/dotNET/csharp.html", "hitmill.com", ".com", "Hitm
ill.com - C# Programming", "You all know by now that C# kicks, and that there are
several sites by the name of C# kicks, but this is not one of them... this is a site for those
of you from the skill levels of beginner, to the advanced C# developer. This site has
annotated C# resources for C# programmers; including links to courses, chats, C#
articles, C# white papers, C# books,e-books, discussions, C# newsgroups, articles, C#
whitepapers, products, C# tools, code examples, and the topic of C# and Web Services.
Have fun.", "c#,c# kicks,C# programming,c# web
services,c#,csharp,sharp,source,programming,sharp editor,c-sharp,asp+,download,pdf,c#
anguage reference,asp.net,c# programmers guide,c# documentation,hitmill,c#
programming articles,dotNET,c# book,c# product,c# runtime,c# compiler,IDE,sample,c#
newsgroup,c-sharp newsgroup,sharp newsgroup,sharp news,c#
news,chapter,code,source,hitmill,c# directory,accept credit cards from your web
site", "", "23765"

"http://www.hitmill.com/internet/searchengines3.html", "hitmill.com", ".com", "Searching
the Internet", "Searching the Internet includes a linked list of searchengines and
directories to use when searching the Web or the Internet.", "searchengine,meta
search,how to search,hitmill", "", "12742"

"http://www.hitmill.com/computers/laptop.html", "hitmill.com", ".com", "Computers", "", ""
", "", "2325"

"http://www.hitmill.com/programming/asp/asp_idx2.html", "hitmill.com", ".com", "ASP
and the Internet:Active Server Pages Tutorial", "ASP and the Internet provides primers
and resources for ASP Web Development, server side scripting, an introduction to Active
Server Pages, VBScript for the beginner, and ASP tutorials. ASP and the Internet

discusses ASP Web application and development and resources for Windows IIS webmasters and developers. The page includes resources for students new to ASP.", "active server pages,asp tutorial,active server pages tutorial,asp and the internet,asp.net,vbscript,javascript,client-side programming,hitmill,server side includes,SSI files,asp components,asp objects,web application,web services", "", "9977" "http://www.hitmill.com/internet/lynx.html", "hitmill.com", ".com", "Lynx Browser", "A discussion about the text-only Lynx browser, with resources listed.", "lynx,browser,platforms,mailing lists,chat,source,text-only", "", "5353" "http://www.hitmill.com/roads/calif.htm", "hitmill.com", ".com", "Hitmill.com - California Roads and Weather", "California road conditions and highway conditions, weather forecasts and advisories; traffic and weather cams, marine weather, hydrometeorological information, ski links, severe weather advisories, posted California current watches and warnings, and severe weather safety tips.", "california roads and weather,california road conditions,roads and weather,california skiing,california,weather,road,road conditions,ski,weather forecasts,marine weather,off shore,offshore,wind surfing,severe atanasoff,hal9000,problem with hal,atanasoff-berry,vannevar bush,j presper eckert,john mauchly,univac,as we may think,bill hewlett,dave packard,stibitz,zanadu,ted nelson,steven wozniak,steve job,william h. gates,bill gates,j presper eckert,john mauchley,internet explorer,history of c++,licklider,kleinrock", "", "18202" "http://www.hitmill.com/roads/severe.html", "hitmill.com", ".com", "Hitmill.com - Severe Weather Preparation and Safety Tips", "Severe weather preparation and safety tips about storms, severe cold, winter storms, ice storms, floods, flash floods, dust storms, hail, hurricanes, lightening, tornados, and more.", "severe weather,preparation,weather,lightning,severe weather preparation,severe weather safety tips,safety,education,tips,lessons,tornado,flood,flash flood,flashflood,dust storm,hitmill,hail,cold,survival,ice storm,severe weather tips,weather tips and safety,winter storms,preparation,hurricane,earthquake,heat,lightening,thunder,severe,warning,weather, alert,gov,edu,org,online,weather safety tips and information,awareness", "", "21410" "http://www.hitmill.com/EQ2.htm", "hitmill.com", ".com", "HITMILL.COM: File Not Found", "Hitmill.com custom 404 page", "Links to hitmill.com home page", "", "2351"

"http://www.hitmill.com/computers/mac_page.html", "hitmill.com", ".com", "Hitmill.com - Apple Bites", "Mac computer links, OS X and UNIX links...", "Apple Computer resources for students,intel mac,i-mac,i-life 08,i-mac books,i-mac links,i-phone,mac life,hitmill,>", "", "9707"

"http://www.hitmill.com/programming/cpp/cppHistory.asp", "hitmill.com", ".com", "History of C++", "A researched article about the history of the C++ programming language with input from Bjarne Stroustrup and including a bibliography and related C++ links.", "history of c++,bjarne stroustrup,b,nb,algol68,bcpl,simula,unix,ken thompson,dennis ritchie,martin richards,c,programming in c++,hitmill,c++ programming,cheryl gribble", "Thu, 31 Mar 2005", "9491"

"http://www.hitmill.com/roads/minn.htm", "hitmill.com", ".com", "Minnesota Roads and Weather", "Road conditions and weather for Minnesota; current temperatures, maps, fishing minnesota, propane stations, Department of Transportation links, sky cams, current conditions and forecasts, live cams, minnesota on the Web, and much more...", "minnesota,maps and driving,directions,roads,nd,road conditions,traffic conditions,sky cams,current conditions,forecasts,temperature,department of transportation,traffic accidents,road,highway,incidents,hotels, tourism,reservations,recreation,ice,sleet,tornado,conditions,snow,ski,weather,travel,weather conditions,points of interest,maps,news,fishing minnesota,live cams,propane stations,minnesota on the Web.", "", "23439"

"http://www.hitmill.com/html/pastels.html", "hitmill.com", ".com", "Hitmill.com - Hexadecimal Color Codes: Pastels Part I", "This is one of six pages of our hexadecimal color values reference for Web authors. There is also a second set of pastel colors, there are free background textures in the HTML section, a page of named selected colors, a color picker tool, and non-dithering color value charts. Return to the Hexadecimal Color Codes main page to use the color picker tool, and to view the other pages in this expanded reference.", "hexadecimal color codes,pastel colors,hitmill,rrggbb,rgb color,pastel hex codes,hitmill,non-dithering,rgb color values,rgb colour,hex colour,hexadecimal rgb color,hexidecimal", "", "26122"

"http://www.hitmill.com/html/color_safe.html", "hitmill.com", ".com", "Hitmill - 216 Color Safe Web Palette", "A tutorial with examples in charts about the 216 color safe

Web palette. These are the non-dithering colors which may be coded into HTML color codes or style sheets, to look the same in various browsers on different computers. This page is part of a 6-page reference for hexadecimal color codes. Return to the main Hex. Reference page to use the color picker tool and to link to the other pages in this series.", "216 color safe web palette, what is a non-dithering color, color safe, browser safe, web palette, hitmill, free, online, tutorial, non-dithering color, color picker tool, primer, colour-safe, colour", "", "27800"

"http://www.hitmill.com/html/sitecolors.html", "hitmill.com", ".com", "hitmill.com - Complementary Color Scheme Tutorial
And Color Directory", "Tutorial about the split complementary color scheme and the complementary color scheme and a directory of color-related tutorials and color directory. Refer to the Color Directory for sites having color selectors and tools for helping to select color schemes, color tutorials, color theory, light and optics and how it relates to color, color references for hexadecimal color, rgb color, cmyk color; advanced color in design; bringing color objects into text; and color in digital graphics. ", "complimentary color scheme tutorial, colour scheme tutorial, color directory, colour directory, hitmill, Hexadecimal and RGB color", "", "19960"

"http://www.hitmill.com/html/pastels2.html", "hitmill.com", ".com", "Hitmill.com - Hexadecimal Color Codes: Pastels, Part 2", "This is one of six pages of our hexadecimal color values reference for Web authors. There is also a another set of pastel colors, free background textures in the HTML section, a page of named selected colors, a color picker tool, and non-dithering color value charts.", "hexadecimal color codes, hex code, hex color, web color, rgb color, rrggbb, hitmill, hexidecimal", "", "21270"

"http://www.hitmill.com/html/formattingcolor.html", "hitmill.com", ".com", "Hitmill.com - RGB Hexadecimal Color Tutorial", "", "color-safe, non-dithering, rgb color tutorial, hexadecimal color tutorial, free, online, colour tutorial, color tutorial, hitmill", "", "9582"

"http://www.hitmill.com/roads/internat.htm", "hitmill.com", ".com", "Hitmill.com - International Weather Resources", "International weather resources is a linked list of international weather sites.", "international weather, online, resources, free, hitmill, world, world-wide, weather information, weather directory, weather resources, weather links, africa, asia, mexico, puerto"

rico,australia,canada,canadian weather,australian weather,united kingdom,uk weather,usa weather,a weather watch,africa weather,europe weather,switzerland weather,spain weather,ireland weather,italy weather,swiss alps weather,converting between temperature scales","","21385"

"http://www.hitmill.com/html/colorchart.html","hitmill.com",".com","Hitmill.com - Hexadecimal Color Codes: Additional Colors","This is a five-page hexadecimal color code reference for Web authors. There is a full page of colour charts for non-dithering color-safe web palette. Hitmill has also included two pages of pastel hexadecimal color codes, one page of named colors, and this page with additional color codes listed in the two charts below. Try the color picker on the main page of the reference.", "hexadecimal color codes,named colors,hitmill,rgb color,rrggbg,color value,color,hex color,color-safe web palette,pastels,pastel hex codes,non-dithering color,colour,hexadecimal colour,hex colour,named colours,colour values,colour-safe,web palette,hexidecimal","","15883"

"http://www.hitmill.com/html/","hitmill.com",".com","Hitmill.com - HTML Tutorials for Students","HTML Tutorials for Students is HTML for beginners, a primary HTML site with new tutorials and resources for formatting text on a Web page, creating tables, how to add an image to a Web page, hexadecimal color values, FTP tutorial, PDF File tutorial, free images, and more...", "html tutorials for students,html tutorial,html for beginners,html primer,web page tutorial,free,online,html table tutorials,audio for Web pages,table tutorial,html table,html sound,free background textures,hex color,hex colour,web page sound,Web page music,meta tags,what is the web,history of computers,free backgrounds for Web pages,ftp tutorial,pdf files,free web space,optimizing for the search engines,web page promotion,history of the web,hexidecimal color values,hexidecimal, colour,colour-safe web palette,hex,hexidemical,fp add-ons,color-safe web palette,browsers,hitmill,html editors,what is html,advanced html,front page links,frontpage links,image fill,lettering,photoshop,clipart,photoshop lettering tutorial,hex color values,image in letter,web images,dynamic html,accept credit cards from your web site,hitmill,charge.com","","11653"

"http://www.hitmill.com/html/sound.html","hitmill.com",".com","Hitmill.com - HTML Sounds Tutorial, Bringing Sound to Your Web Pages","Code for HTML sound tags, how to have sound on your Web page. A free tutorial with code examples for coding HTML

sounds and linking to sounds, as well as how to play a sound as the Web page is loading.", "html sounds,web page sound tutorial,audio tutorial,html sound tutorial,how to put sound on your web page,code examples for html sound,hitmill,midi files,music files,mp3 files,sound plugins,sound software,audio software,audio plugins,audio player,sound player","", "18019"

"http://www.hitmill.com/html/webpromo.html","hitmill.com",".com","Web Page Promotion, Increase Web Traffic","This is a free online HTML tutorial about Web page promotion and how to increase Web traffic to your site. Determine who your audience shall be, optimize for the search engines, know your meta tags and register with search engines.", "web promotion,increase web traffic,hitmill,free html tutorial,meta tag tutorials,introduction to meta tags,meta tags,meta tag,tutorial,free,online tutorial,ftp tutorial,search engine optimization,optimize for the search engines,register with search engine,accept credit cards","", "5333"

"http://www.hitmill.com/xml/xml.asp","hitmill.com",".com","XML","This is a free XML tutorial, an introduction to xml. Page has moved, please change your bookmark to point to http://www.hitmill.com/xml/xml.html","xml tutorial,xml primer,free,online,hitmill,xml,lesson,introduction to xml,it tutorial","Sat, 16 Apr 2005","4748"

"http://www.hitmill.com/roads/arkansas.html","hitmill.com",".com","Hitmill.com - Arkansas Roads and Weather","Arkansas weather: current warnings, alerts; current conditions; Arkansas Traffic conditions, Web cams, Highway Information, severe weather warnings; maps, news, business and employment, travel, tourism, government pages, education, university, recreation, and related links.", "arkansas roads and weather,arkansas,online,weather,current conditions,alerts,advisories,warnings,conway,little rock,searcy,fayetteville,benton,fort smith,north little rock,roads and weather,road information,highway closures,highway information,traffic,information,accidents,street,map,conway,news,business,employment,travel,tourism,government,outdoors,recreation,sports,maps,related links,arkansas map,nursing,road,highway,street,incidents,hitmill,gardening","", "23553"

"http://www.hitmill.com/html/what_is.html","hitmill.com",".com","What is HTML","What is HTML? It is hypertext markup language. Read about HTML in the

article below where hypertext markup language is explained and described; a descriptive definition of HTML.", "definition of html, what is html, html, what is, hypertext markup language, hypertext markup language is explained and described, a descriptive definition of html", "", "11967"

"http://www.hitmill.com/html/primer.html", "hitmill.com", ".com", "HTML Tutorials", "HTML tutorials for students and Web authors.", "html, html tutorial, html text, html primer, hitmill, history of the web, cern, history of computers, web browsers, first web page, text editors, images, clipart, cascading style sheets, css, dhtml, dynamic html, graphics, software, image, free, html table tutorial, web page sound, html beginner's guide, html music, image tags, color-safe palette, pallet, palette, color safe, web, ", "", "6186"
"http://www.hitmill.com/roads/nevada.htm", "hitmill.com", ".com", "Hitmill.com - Nevada Roads and Weather", "Nevada weather and current conditions, road information, tourism, lodging, housing, web cams, traffic incidents, and more... For lodging near California ski resorts, see the California Roads and Weather.", "nevada road conditions, nevada weather, nevada tourism, reno motel, reno hotel, reno lodging, las vegas hotel, las vegas lodging, las vegas motel, nevada traffic conditions, current observations, nevada traffic accidents, nevada, emergency road, highway incident, chain requirements, conditions, minden, current conditions, casinos, gaming, map, tourism, geology, silver, mining, rivers, road, highway, incidents, conditions, snow, ski, weather, weather conditions, las vegas hotels, points of interest, maps, nevada hotels, nevada motel, nevada lodging, accommodations, rooms in Reno, tourist, nevada travel guide, information, statistics, hitmill", "", "19647"

"http://www.hitmill.com/html/hostingPortals.html", "hitmill.com", ".com", "Web Hosting Portals", "A discussion about Web hosting providers, web presence providers.", "Web host, web presence provider, HTML, web, hosting, provider, hosting company, iis, asp, frontpage host, Web page host, fp hosting, asp host, asp hosting", "", "4772"
"http://www.hitmill.com/html/primers/lesson1.html", "hitmill.com", ".com", "How to Create Your First Web Page", "This is an HTML Tutorial about how to create your first Web page", "html tutorial, how to create your first web page, hitmill, html primer, code html, web page, make web page", "", "21596"

"http://www.hitmill.com/html/aboutpdf.html", "hitmill.com", ".com", "Hitmill.com - PDF File Tutorial", "PDF file tutorial about What is a PDF file? and how to obtain the Adobe Acrobat Reader software used to read the PDF files or to get the Adobe Acrobat Reader Full Version to create PDF files. PDF Files end with the .pdf extension and are cross platform files meaning they can be read by any computer operating system. A PDF file can be locked against unauthorized editing. I have tried to include a few additional Web resources for those of you wishing more information.", "about pdf files, what are pdf files, pdf files and acrobat reader, hitmill, pdf file tutorial, pdf tutorial, tutorial, pdf, pdf format, pdf file, acrobat reader, adobe acrobat", "", "7932"

"http://www.hitmill.com/html/tables/tables.html", "hitmill.com", ".com", "HTML Table Tutorials: Introduction to Tables", "Tutorials for beginners about how to code tables using HTML; includes table backgrounds, images, headings, data cells, and more...", "lesson 1: introduction to html tables, html table tutorial, html, table, tables, tutorial, background, image, headings, table rows, table data, table elements, table heading, hitmill", "", "20791"

"http://www.hitmill.com/html/webimages.html", "hitmill.com", ".com", "Hitmill.com - An Introduction to Image Files", "This is a free online HTML tutorial explaining about cross-platform images suitable for use in a Web document and also explaining about image transparency. There are links to the image software for PaintShop Pro, PhotoShop, Quark Express and also some related links listed at the bottom of the article.", "web page images tutorial, free, online, html tutorial, introduction, web page images, html image, html picture, files, jpg, gif, tiff, bmp, transparent images, transparency, hitmill, cross-platform, cross platform, image transparency, gif89a, paintshop pro, photoshop, quark express, image software", "", "12000"

"http://www.hitmill.com/html/texture/freetexture.html", "hitmill.com", ".com", "Hitmill.com - Free Seamless Background Textures For Personal Web Pages", "Free seamless background textures for Web page including a parchment background.", "free seamless background textures, free background textures, parchment background, free, seamless, personal web page, personal, background texture, online, background, texture, web, free background images, hitmill", "", "8647"

"http://www.hitmill.com/html/clipart.html", "hitmill.com", ".com", "Hitmill.com - Web Clipart Tutorial", "Web clipart tutorial includes an introduction to images, coding an image for your Web page, where to find clipart on the Web, and Introduction to Graphics Software.", "web clipart tutorial, free, online, tutorial, clipart, web, clipart tutorial, image tutorial, graphics tutorial, html clipart, hitmill, web clipart, image tag, coding html images, code image tag", "", "12439"

"http://www.hitmill.com/html/screen.html", "hitmill.com", ".com", "Hitmill.com - Monitor Screen Resolutions", "Informative article about computer monitor screen resolutions and how to change the monitor screen resolution settings for a PC and for a Mac, including a new discussion about the aspect ratio of monitors and of resolution settings.", "computer monitor, screen resolutions, aspect ratio of monitor, aspect ratio of resolution setting, aspect ratios, LCD, VGA, SVGA, XGA, change resolution, monitor, VDT, screen, monitor resolution, screen resolution, computer monitor, display terminal resolution, vram, video adapter", "", "18965"

"http://www.hitmill.com/html/books.html", "hitmill.com", ".com", "Hitmill.com - HTML Books", "HTML for Students is a good starting point for anyone new to Web page authoring, HTML, Hypertext Markup Language. Begin by reading online information and then refer to the HTML book list, below, if you are looking for an additional reference to have on hand. See Hitmill.com's list of recommended HTML books.", "html books, html tutorials for students, hitmill, hitmill's list of recommended html books", "", "7936"

"http://www.hitmill.com/programming/homeapplets.html", "hitmill.com", ".com", "Java Applets, Where to Find Free Applets", "A reference for Web authors about free java applets.", "java, applet, applets, where, find, free", "", "9237"

"http://www.hitmill.com/html/graphics.html", "hitmill.com", ".com", "Hitmill - Introduction to Web Design Software", "This is an overview and introduction to design graphics software, meant as a tutorial for the beginning student and includes software useful for Web and Digital Design professionals.", "digital design software, web design software, design graphics software, hitmill, overview, tutorial, introduction", "", "17473"

"http://www.hitmill.com/html/picture.html", "hitmill.com", ".com", "Hitmill.com - How to Code A Web Page Image", "Code HTML images, Web page images, and how to code an

image onto your Web page. GIFs, JPEGs, Bitmaps, and what to do with a TIFF. ", "html image tag,picture,web page,gif image,jpeg image,bitmap images,tiff images,how to code a web page image,html image code tutorial,free,online,html tutorial,image code,picture code,code an image for your web page,web page images,html picture tutorial,html image tag,image tag,picture tag,jpg,jpeg,gif,bitmap,html,images,pictures,tags,tag,picture,image,hitmill,image codes", "", "25759"

"http://www.hitmill.com/html/promote/meta1.htm", "hitmill.com", ".com", "Meta Tag Tutorial #1: Recirecting a User to Another Page", "A meta tag tutorial discusses how to redirect a user to another Web page using meta information in the head of the HTML document.", "meta tag tutorial,redirect a user to another page,hitmill,meta tag primer,html document", "", "7492"

"http://www.hitmill.com/html/links.htm", "hitmill.com", ".com", "Hitmill.com - HTML Related Links", "HTML related links from the HTML tutorials Web site at hitmill.com", "html related links,free html tutorials for students", "", "7773"

"http://www.hitmill.com/dhtml/", "hitmill.com", ".com", "Hitmill.com - What is Dynamic HTML?", "Dynamic HTML technologies are explained and explored. Tutorials are included in this site. Dynamic HTML is a group of technologies. See our definition and explanation of the technologies comprising Dynamic HTML, or DHTML.", "what is dynamic html,what is dhtml,dhtml tutorial,dhtml primer,introduction to dhtml,dynamic html tutorial,dynamic html primer,hitmill,introduction to dynamic html,javascript,cascading style sheets,css-2,css2,css tutorial,javascript tutorials,cssp,css-p", "", "7022"

"http://www.hitmill.com/html/adv_html_design.html", "hitmill.com", ".com", "Hitmill.com - More HTML And Web Design", ">", "advanced html and web design", "", "13625"

"http://www.hitmill.com/dhtml/intro.html", "hitmill.com", ".com", "Hitmill.com - Introduction to DHTML", "An introductory article explaining the technologies comprising Dynamic HTML: What is DHTML?", "dynamic html,dhtml,tutorial,dynamic,html,free,javascript,js,jss,css,css1,css1,cascading style sheets,hitmill,css rules,css position,z-index,layer,document object model,what is dhtml", "", "15970"

"http://www.hitmill.com/programming/asp/index.htm","hitmill.com",".com","HITMILL.COM: File Not Found","Hitmill.com custom 404 page","Links to hitmill.com home page","","2351"

"http://www.hitmill.com/programming/asp/ssBasics.html","hitmill.com",".com","ASP: ASP Server-Side Basics for Active Server Pages","SSI tutorial: Server-Side Include Files for ASP","asp,active server pages,tutorial,basics,server-side basics,hitmill,ssi file,server-side include file,server side includes,asp,asp+,asp.net,active server pages,include file,ssi tutorial,asp tutorial","","8054"

"http://www.hitmill.com/programming/asp/welcome.html","hitmill.com",".com","Hitmill.com - Hello and Welcome ASP Tutorial with Date and Time","Free online ASP tutorial, Active Server Pages technology, with demo. code: Hello and Welcome ASP Tutorial, with date and time.","hello asp tutorial,date asp tutorial,time asp code,hello asp,asp.net tutorial,asp,active server pages","","5012"

AWCMA Meta extraction and titles of pages from www.hitmill.com

- 1:Apache Server Resources
- 2:ASP and ASP.NET Resources
- 3:ASP and the Internet:Active Server Pages Tutorial
- 4:ASP: ASP Server-Side Basics for Active Server Pages
- 5:Chalk Board Math Resources for Students
- 7:Chalk Line
- 8:College Course Resources and References
- 9:Computer Magazines and E-zines
- 10:Apache Server Resources
- 11:ASP and ASP.NET Resources
- 12:ASP and the Internet:Active Server Pages Tutorial
- 13:ASP: ASP Server-Side Basics for Active Server Pages
- 14:Chalk Board Math Resources for Students
- 16:Chalk Line
- 17:College Course Resources and References

- 18:Computer Magazines and E-zines
- 19:Free Online JavaScript Tutorials: JavaScript for Web Pages and Client-Side Web Coding
- 20:History of the Web Beginning at CERN
- 21:Hitmill.com - A Weather Watch
- 22:Hitmill.com - About HiTMILL.com
- 23:Hitmill.com - Ancient History and Civilizations
- 24:Hitmill.com - C# Programming
- 25:hitmill.com - C++ Programming for the Beginning C++ Student
- 26:Hitmill.com - California History
- 27:Hitmill.com - California Indians
- 28:Hitmill.com - California Roads and Weather
- 29:Hitmill.com - Comparative Religion and Religious Studies
- 30:Hitmill.com - Computer Hoax Tutorial
- 31:Hitmill.com - Create An Image-Fill for Text Using PhotoShop
- 32:Hitmill.com - Find News
- 33:Hitmill.com - FTP New User: An FTP Guide, Tutorial and Directory
- 35:Hitmill.com - Hexadecimal Color Codes: Selected Named Colors
- 36:Hitmill.com - Hexadecimal Color Values Reference
- 37:Hitmill.com - HTML Tutorials for Students
- 38:Hitmill.com - Internet Training and Technology for Students
- 40:Hitmill.com - Introduction to the Internet
- 41:Hitmill.com - Java Programming
- 42:Hitmill.com - Link Partners
- 43:Hitmill.com - PhotoShop Tutorial: Fill Text With Pattern-Fill
- 44:Hitmill.com - PhotoShop Tutorial: Lettering and Gradient-Fills
- 46:Hitmill.com - Programming Resources
- 47:Hitmill.com - Search Engine Tutorial
- 48:Hitmill.com - Seismology - Earthquakes
- 49:Hitmill.com - Severe Weather Preparation and Safety Tips
- 50:hitmill.com - Simula Programming Language

- 51:Hitmill.com - Travel Corner Travel Directory
- 52:Hitmill.com - Tutorial About Fonts
- 53:Hitmill.com - Tutorial About Kerberos
- 54:Hitmill.com - Visual Basic 6 Programming
- 55:Hitmill.com - Web Browser Tutorial for Students
- 57:Hitmill.com - Web Services Tutorial for Students
- 58:Hitmill.com - What is Dynamic HTML?
- 59:Hitmill.com - XML Resources
- 60:How To Fill Text With A Fabric
- 61:HTML Editors
- 62:Introduction to Cascading Style Sheets Rules
- 63:Introduction to Computers
- 64:Java Applets, Where to Find Free Applets
- 65:Photoshop References and Tutorials
- 66:Programming in C
- 67:Programming Overview
- 68:REALbasic for Macintosh
- 69:Uniform Resource Locator Tutorial: What is a URL?
- 70:Apache Server Resources
- 71:ASP and ASP.NET Resources
- 72:ASP and the Internet: Active Server Pages Tutorial
- 73:ASP: ASP Server-Side Basics for Active Server Pages
- 74:Chalk Board Math Resources for Students
- 76:Chalk Line
- 77:College Course Resources and References
- 78:Computer Magazines and E-zines
- 79:Free Online JavaScript Tutorials: JavaScript for Web Pages and Client-Side Web Coding
- 80:History of the Web Beginning at CERN
- 81:Hitmill.com - A Weather Watch
- 82:Hitmill.com - About HiTMilL.com

- 83:Hitmill.com - Ancient History and Civilizations
- 84:Hitmill.com - C# Programming
- 85:hitmill.com - C++ Programming for the Beginning C++ Student
- 86:Hitmill.com - California History
- 87:Hitmill.com - California Indians
- 88:Hitmill.com - California Roads and Weather
- 89:Hitmill.com - Comparative Religion and Religious Studies
- 90:Hitmill.com - Computer Hoax Tutorial
- 91:Hitmill.com - Create An Image-Fill for Text Using PhotoShop
- 92:Hitmill.com - Find News
- 93:Hitmill.com - Hexadecimal Color Codes: Selected Named Colors
- 94:Hitmill.com - Hexadecimal Color Values Reference
- 95:Hitmill.com - HTML Tutorials for Students
- 96:Hitmill.com - Introduction to the Internet
- 97:Hitmill.com - Java Programming
- 98:Hitmill.com - Link Partners
- 99:Hitmill.com - PhotoShop Tutorial: Fill Text With Pattern-Fill
- 100:Hitmill.com - PhotoShop Tutorial: Lettering and Gradient-Fills
- 102:Hitmill.com - Programming Resources
- 103:Hitmill.com - Search Engine Tutorial
- 104:Hitmill.com - Seismology - Earthquakes
- 105:Hitmill.com - Severe Weather Preparation and Safety Tips
- 106:hitmill.com - Simula Programming Language
- 107:Hitmill.com - Travel Corner Travel Directory
- 108:Hitmill.com - Tutorial About Fonts
- 109:Hitmill.com - Tutorial About Kerberos
- 110:Hitmill.com - Visual Basic 6 Programming
- 111:Hitmill.com - Web Browser Tutorial for Students
- 113:Hitmill.com - Web Services Tutorial for Students
- 114:Hitmill.com - What is Dynamic HTML?
- 115:Hitmill.com - XML Resources

- 116:How To Fill Text With A Fabric
- 117:HTML Editors
- 118:Introduction to Cascading Style Sheets Rules
- 119:Introduction to Computers
- 120:Java Applets, Where to Find Free Applets
- 121:Photoshop References and Tutorials
- 122:Programming in C
- 123:Programming Overview
- 124:REALbasic for Macintosh
- 125:Uniform Resource Locator Tutorial: What is a URL?
- 126:Apache Server Resources
- 127:ASP and ASP.NET Resources
- 128:ASP and the Internet:Active Server Pages Tutorial
- 129:ASP: ASP Server-Side Basics for Active Server Pages
- 130:Chalk Board Math Resources for Students
- 132:Chalk Line
- 133:College Course Resources and References
- 134:Computer Magazines and E-zines
- 135:Free Online JavaScript Tutorials: JavaScript for Web Pages and Client-Side Web Coding
- 136:History of the Web Beginning at CERN
- 137:Hitmill.com - A Weather Watch
- 138:Hitmill.com - About HiTMiLL.com
- 139:Hitmill.com - Ancient History and Civilizations
- 140:Hitmill.com - C# Programming
- 141:hitmill.com - C++ Programming for the Beginning C++ Student
- 142:Hitmill.com - California History
- 143:Hitmill.com - California Indians
- 144:Hitmill.com - California Roads and Weather
- 145:Hitmill.com - Comparative Religion and Religious Studies
- 146:Hitmill.com - Computer Hoax Tutorial

- 147:Hitmill.com - Create An Image-Fill for Text Using PhotoShop
- 148:Hitmill.com - Find News
- 149:Hitmill.com - Hexadecimal Color Codes: Selected Named Colors
- 150:Hitmill.com - Hexadecimal Color Values Reference
- 151:Hitmill.com - HTML Tutorials for Students
- 152:Hitmill.com - Introduction to the Internet
- 153:Hitmill.com - Java Programming
- 154:Hitmill.com - Link Partners
- 155:Hitmill.com - PhotoShop Tutorial: Fill Text With Pattern-Fill
- 156:Hitmill.com - PhotoShop Tutorial: Lettering and Gradient-Fills
- 158:Hitmill.com - Programming Resources
- 159:Hitmill.com - Search Engine Tutorial
- 160:Hitmill.com - Seismology - Earthquakes
- 161:Hitmill.com - Severe Weather Preparation and Safety Tips
- 162:hitmill.com - Simula Programming Language
- 163:Hitmill.com - Travel Comer Travel Directory
- 164:Hitmill.com - Tutorial About Fonts
- 165:Hitmill.com - Tutorial About Kerberos
- 166:Hitmill.com - Visual Basic 6 Programming
- 167:Hitmill.com - Web Browser Tutorial for Students
- 169:Hitmill.com - Web Services Tutorial for Students
- 170:Hitmill.com - What is Dynamic HTML?
- 171:Hitmill.com - XML Resources
- 172:How To Fill Text With A Fabric
- 173:HTML Editors
- 174:Introduction to Cascading Style Sheets Rules
- 175:Introduction to Computers
- 176:Java Applets, Where to Find Free Applets
- 177:Photoshop References and Tutorials
- 178:Programming in C
- 179:Programming Overview

- 180:REALbasic for Macintosh
- 181:Uniform Resource Locator Tutorial: What is a URL?
- 182:Apache Server Resources
- 183:ASP and ASP.NET Resources
- 184:ASP and the Internet: Active Server Pages Tutorial
- 185:ASP: ASP Server-Side Basics for Active Server Pages
- 186:Chalk Board Math Resources for Students
- 188:Chalk Line
- 189:College Course Resources and References
- 190:Computer Magazines and E-zines
- 191:Free Online JavaScript Tutorials: JavaScript for Web Pages and Client-Side Web Coding
- 192:History of the Web Beginning at CERN
- 193:Hitmill.com - A Weather Watch
- 194:Hitmill.com - About HiTMILL.com
- 195:Hitmill.com - Ancient History and Civilizations
- 196:Hitmill.com - C# Programming
- 197:hitmill.com - C++ Programming for the Beginning C++ Student
- 198:Hitmill.com - California History
- 199:Hitmill.com - California Indians
- 200:Hitmill.com - California Roads and Weather
- 201:Hitmill.com - Comparative Religion and Religious Studies
- 202:Hitmill.com - Computer Hoax Tutorial
- 203:Hitmill.com - Create An Image-Fill for Text Using PhotoShop
- 204:Hitmill.com - Find News
- 205:Hitmill.com - FTP New User: An FTP Guide, Tutorial and Directory
- 207:Hitmill.com - Hexadecimal Color Codes: Selected Named Colors
- 208:Hitmill.com - Hexadecimal Color Values Reference
- 209:Hitmill.com - HTML Tutorials for Students
- 210:Hitmill.com - Internet Training and Technology for Students
- 212:Hitmill.com - Introduction to the Internet

- 213:Hitmill.com - Java Programming
- 214:Hitmill.com - Link Partners
- 215:Hitmill.com - PhotoShop Tutorial: Fill Text With Pattern-Fill
- 216:Hitmill.com - PhotoShop Tutorial: Lettering and Gradient-Fills
- 218:Hitmill.com - Programming Resources
- 219:Hitmill.com - Search Engine Tutorial
- 220:Hitmill.com - Seismology - Earthquakes
- 221:Hitmill.com - Severe Weather Preparation and Safety Tips
- 222:hitmill.com - Simula Programming Language
- 223:Hitmill.com - Travel Corner Travel Directory
- 224:Hitmill.com - Tutorial About Fonts
- 225:Hitmill.com - Tutorial About Kerberos
- 226:Hitmill.com - Visual Basic 6 Programming
- 227:Hitmill.com - Web Browser Tutorial for Students
- 229:Hitmill.com - Web Services Tutorial for Students
- 230:Hitmill.com - What is Dynamic HTML?
- 231:Hitmill.com - XML Resources
- 232:How To Fill Text With A Fabric
- 233:HTML Editors
- 234:Introduction to Cascading Style Sheets Rules
- 235:Introduction to Computers
- 236:Java Applets, Where to Find Free Applets
- 237:Photoshop References and Tutorials
- 238:Programming in C
- 239:Programming Overview
- 240:REALbasic for Macintosh
- 241:Uniform Resource Locator Tutorial: What is a URL?
- 242:Free Online JavaScript Tutorials: JavaScript for Web Pages and Client-Side Web Coding
- 243:History of the Web Beginning at CERN
- 244:Hitmill.com - A Weather Watch

- 245:Hitmill.com - About HiTMILL.com
- 246:Hitmill.com - Ancient History and Civilizations
- 247:Hitmill.com - C# Programming
- 248:hitmill.com - C++ Programming for the Beginning C++ Student
- 249:Hitmill.com - California History
- 250:Hitmill.com - California Indians
- 251:Hitmill.com - California Roads and Weather
- 252:Hitmill.com - Comparative Religion and Religious Studies
- 253:Hitmill.com - Computer Hoax Tutorial
- 254:Hitmill.com - Create An Image-Fill for Text Using PhotoShop
- 255:Hitmill.com - Find News
- 256:Hitmill.com - FTP New User: An FTP Guide, Tutorial and Directory
- 258:Hitmill.com - Hexadecimal Color Codes: Selected Named Colors
- 259:Hitmill.com - Hexadecimal Color Values Reference
- 260:Hitmill.com - HTML Tutorials for Students
- 261:Hitmill.com - Internet Training and Technology for Students
- 263:Hitmill.com - Introduction to the Internet
- 264:Hitmill.com - Java Programming
- 265:Hitmill.com - Link Partners
- 266:Hitmill.com - PhotoShop Tutorial: Fill Text With Pattern-Fill
- 267:Hitmill.com - PhotoShop Tutorial: Lettering and Gradient-Fills
- 269:Hitmill.com - Programming Resources
- 270:Hitmill.com - Search Engine Tutorial
- 271:Hitmill.com - Seismology - Earthquakes
- 272:Hitmill.com - Severe Weather Preparation and Safety Tips
- 273:hitmill.com - Simula Programming Language
- 274:Hitmill.com - Travel Corner Travel Directory
- 275:Hitmill.com - Tutorial About Fonts
- 276:Hitmill.com - Tutorial About Kerberos
- 277:Hitmill.com - Visual Basic 6 Programming
- 278:Hitmill.com - Web Browser Tutorial for Students

- 280:Hitmill.com - Web Services Tutorial for Students
 - 281:Hitmill.com - What is Dynamic HTML?
 - 282:Hitmill.com - XML Resources
 - 283:How To Fill Text With A Fabric
 - 284:HTML Editors
 - 285:Introduction to Cascading Style Sheets Rules
 - 286:Introduction to Computers
 - 287:Java Applets, Where to Find Free Applets
 - 288:Photoshop References and Tutorials
 - 289:Programming in C
 - 290:Programming Overview
 - 291:REALbasic for Macintosh
 - 292:Uniform Resource Locator Tutorial: What is a URL?
- 1: apache,apache resources,apache links,apache directory,apache http server,php,mysql,free,tutorial,apache server resources,hitmill,server-side include,files,htaccess file,php/mysql,apache books,file,apache developer,http project,apache project,ssi file,unix reference
 - 2: This is an Apache Server guide for users, developers, webmasters, and students as we include user/developer information, Apache forums, Apache FAQ, PHP server-side includes, HTACCESS file, Apache Server books, and link to the Apache Project, additional Apache HTTP Server resources, and a UNIX resource list.
 - 3: asp.net,asp,web development,asp and asp.net,resources,forums,books,hitmill,asp.net development community
 - 4: ASP.NET resources for Web Development, products, forums, free code sources, books -- resources for the ASP.NET development community.
 - 5: active server pages,asp tutorial,active server pages tutorial,asp and the internet,asp.net,
vbscript,javascript,client-side programming,hitmill,server side includes,SSI files,
asp components,asp objects,web application,web services
 - 6: ASP and the Internet provides primers and resources for ASP Web Development, server side scripting, an introduction to Active Server Pages, VBScript for the beginner, and ASP tutorials. ASP and the Internet discusses ASP Web application and development

and resources for Windows IIS webmasters and developers. The page includes resources for students new to ASP.

7: asp,active server pages,tutorial,basics,server-side basics,hitmill,ssi file,server-side include file,server side includes,asp,asp+,asp.net,active server pages,include file,ssi tutorial,asp tutorial

8: SSI tutorial: Server-Side Include Files for ASP

9: hitmill,mathematics,research,resources,women mathematicians,history of mathematics,algebra tutorials,geometry tutorials,pre-calculus,tutorial,precalculus,calculus tutorial,abstract algebra tutorials,miscellaneous math links,math links for students and educations,teaching math,algebra,pre-algebra,geometry theorems,geometry,calculus,teaching calculus,careers in mathematics,chalk board mathematics,women in mathematics,women in science,math resources,math homework help

10: Chalk Board is a Mathematics directory site with resources for students and educators.

11: chalk line,fantasy football,nfl football,football,afc,nfc,football scores,news,sports scores,chalk line,football

scores,sporting,sport,sports,score,scoreboard,polo,gymnastics,nfl,butkus,equestrian,equestrian polo,baseball scores,baseball,hitmill,football news,football injuries,baseball news

12: Chalk Line is a directory of sports news, scores and resources ... for polo, gymnastics, NFL, equestrian, baseball, football, fantasy football, baseball scores, and more...

13: college course resources and references,comparative religion,hitmill,online,free,computer programming,seismology,english,math,math links,math directory,california history,california geography,computer programming,visual basic,vb,c++,6,ancient history and

civilizations,sports,travel,weather,california indians,photoshop,california geography,california history,history of computers,computer history,math courses,

14: College course resources, college Preparatory classes, resources for math courses, Comparative Religion, PhotoShop Graphics, Ancient Civilizations, English, Computer Programming, Visual Basic and C++ Programming, History of Computers, California History, California Indians, California Geography, and more...

- 15: online computer and internet magazines,online,internet,ezone,magazine,e-zine,directory,computer e-zine,computer ezine,it e-zine,it magazine,computer technology e-zine,networking magazine,processor magazine,robotics magazine,vb user magazine,c++ user magazine,.net magazine,hitmill,computers,programming languages,ieee,robotics,processor,networking,wlan,wireless,wifi,bluetooth,macintosh,mac,c,apple,os x,ipod,imac,mac g5,online news,travel,
- 16: ...compiled directory of computer and network magazines include topics such as computer technology, networking, processors and hardware, programming languages, robotics, visual basic, .NET technology, c++ programming, c# programming, xml magazines and more..
- 17: apache,apache resources,apache links,apache directory,apache http server,php,mysql,free,tutorial,apache server resources,hitmill,server-side include,files,htaccess file,php/mysql,apache books,file,apache developer,http project,apache project,ssi file,unix reference
- 18: This is an Apache Server guide for users, developers, webmasters, and students as we include user/developer information, Apache forums, Apache FAQ, PHP server-side includes, HTACCESS file, Apache Server books, and link to the Apache Project, additional Apache HTTP Server resources, and a UNIX resource list.
- 19: asp.net,asp,web development,asp and asp.net,resources,forums,books,hitmill,asp.net development community
- 20: ASP.NET resources for Web Development, products, forums, free code sources, books -- resources for the ASP.NET development community.
- 21: active server pages,asp tutorial,active server pages tutorial,asp and the internet,asp.net,
vbscript,javascript,client-side programming,hitmill,server side includes,SSI files,
asp components,asp objects,web application,web services
- 22: ASP and the Internet provides primers and resources for ASP Web Development, server side scripting, an introduction to Active Server Pages, VBScript for the beginner, and ASP tutorials. ASP and the Internet discusses ASP Web application and development and resources for Windows IIS webmasters and developers. The page includes resources for students new to ASP.

- 23: asp,active server pages,tutorial,basics,server-side basics,hitmill,ssi file,server-side include file,server side includes,asp,asp+,asp.net,active server pages,include file,ssi tutorial,asp tutorial
- 24: SSI tutorial: Server-Side Include Files for ASP
- 25: hitmill,mathematics,research,resources,women mathematicians,history of mathematics,algebra tutorials,geometry tutorials,pre-calculus,tutorial,precalculus,calculus tutorial,abstract algebra tutorials,miscellaneous math links,math links for students and educations,teaching math,algebra,pre-algebra,geometry theorems,geometry,calculus,teaching calculus,careers in mathematics,chalk board mathematics,women in mathematics,women in science,math resources,math homework help
- 26: Chalk Board is a Mathematics directory site with resources for students and educators.
- 27: chalk line,fantasy football,nfl football,football,afc,nfc,football scores,news,sports scores,chalk line,football scores,sporting,sport,sports,score,scoreboard,polo,gymnastics,nfl,butkus,equestrian,equestrian polo,baseball scores,baseball,hitmill,football news,football injuries,baseball news
- 28: Chalk Line is a directory of sports news, scores and resources ... for polo, gymnastics, NFL, equestrian, baseball, football, fantasy football, baseball scores, and more...
- 29: college course resources and references,comparative religion,hitmill,online,free,computer programming,seismology,english,math,math links,math directory,california history,california geography,computer programming,visual basic,vb,c++,6,ancient history and civilizations,sports,travel,weather,california indians,photoshop,california geography,california history,history of computers,computer history,math courses,
- 30: College course resources, college Preparatory classes, resources for math courses, Comparative Religion, PhotoShop Graphics, Ancient Civilizations, English, Computer Programming, Visual Basic and C++ Programming, History of Computers, California History, California Indians, California Geography, and more...
- 31: online computer and internet magazines,online,internet,ezone,magazine,e-zine,directory,computer e-zine,computer ezine,it e-zine,it magazine,computer technology

- e-zine, networking magazine, processor magazine, robotics magazine, vb user magazine, c++ user magazine, .net magazine, hitmill, computers, programming languages, ieee, robotics, processor, networking, wan, wireless, wyfy, bluetooth, macintosh, mac, apple, os x, ipod, imac, mac g5, online news, travel,
- 32: ...compiled directory of computer and network magazines include topics such as computer technology, networking, processors and hardware, programming languages, robotics, visual basic, .NET technology, c++ programming, c# programming, xml magazines and more..
- 33: free online javascript tutorials, javascript for web pages, javascript tutorials, javascript primers, javascript, hitmill, client side programming, javascript button code, free, asp, active server pages, free javascript code, code examples, active server pages accept credit cards from your web site, hitmill ad for charge.com
- 34: This page offers free online JavaScript tutorials and is a primary JavaScript site for students and developers with special emphasis on learning to implement JavaScript for Web pages, or precisely JavaScript for client-side Web code; tutorials, tips and resources are included along with code examples for students. We recommended some prior knowledge of HTML markup language prior to learning JavaScript since many of the code examples are actual Web page applications of JavaScript client-side programming and ASP (Active Server Pages technology). Do not confuse JavaScript with Java.
- 35: history of the web beginning at cern, berners-lee, xanadu, as we were, hypertext, http, mosaic, robert callilliau, html, mark andreesen, jim clark, netscape corp, web statistics, browser history, history of browsers, web demographics, web history, history of the web, hitmill
- 36: The History of the Web Beginning at CERN is an original research article by Cheryl Gribble with Bibliography and Related Links.
- 37: uk, united kingdom, uk coast, uk coastal forecasts, weather, road conditions, online, california roads and weather, quebec conditions, quebec weather, quebec forecast, alberta, british columbia, manitoba, quebec, new brunswick, nova scotia, nunavut, newfoundland, labrador, ontario, yukon, world, prince edward island, saskatchewan, northwest territories, current conditions and forecast, canadian weather forecast, canada current conditions, canada forecast, nevada weather and road

conditions,ski conditions,roads and weather,weather and road
conditions,california,nevada,arizona,oregon,current conditions,north dakota,colorado
roads and weather,colorado outdoors,noaa,colorado current conditions,hitmill,weather
forecasts,weather warnings,noaa links,hitmill,manitoba conditions and forecast,current
conditions,forecast,new brunswick,north america,usa,united states,,marine
weather,aviation weather,road and weather conditions,severe weather,severe
weather,preparation,safety tips,a weather watch,fleet weather,forecasts

38: A free compiled online directory of weather links for the United States, Canada and
North America, including marine, aviation, and noaa weather links, road and weather
conditions,state weather, ski conditions, marine weather, aviation weather, road and
weather current conditions, and links to other online weather pages.

39: about hitmill,it

training,students,developers,computer,internet,technology,training,students,web
technology,programing,networking,news,weather,california

history,free,online,hitmill,about

40: Internet technology training and tutorials for students and developers; computer
networking topics, programming resources and tutoring; web technology tutoring,
training and resources for students; html training, tutoring and resources and much more.
See home page at www.hitmill.com.

41: ancient civilizations,ancient history,mesopotamia,ur,ninevah,assyrian
history,assyria,babylonia,babylon,sumer,hitmill,rome,greece,medieval

42: Web resources for the study of Ancient History and Civilizations: Mesopotamia, Ur,
Ninevah, Assyrian History, Assyria, Babylonia, Babylon, Sumer; Rome, Greece,
Medieval

43: C# programming,c# web services,c#,csharp,sharp,source,programming,sharp
editor,c-sharp,asp+,download,pdf,c# anguage reference,asp.net,c# programmers guide,c#
documentation,hitmill,c# programming articles,dotNET,c# book,c# product,c#
runtime,c# compiler,IDE,sample,c# newsgroup,c-sharp newsgroup,sharp
newsgroup,sharp news,c# news,chapter,code,source,hitmill,c# directory,accept credit
cards from your web site

- 44: Annotated C# resource site for C# programmers; includes links to courses, chats, C# articles, C# white papers, C# books, e-books, courses, discussions, C# newsgroups, articles, C# whitepapers, products, tools, code examples, and the topic of C# and Web Services.
- 45: c++ programming for the beginning c++ student, c++ beginner's tutorials, c++ code library, c++ help, hitmill
- 46: A primary C++ site for the beginning c++ student; c++ programming for the beginning c++ student with tutorials for the c++ beginner, C++ Hello World tutorial and c++ help for students.
- 47: california history, index, robert louis stevenson, jack london, hitmill, history, california earthquakes, fort ross, bear flag revolt, california explorers, history of american indians in california, donner party, berryessa family, benecia, california gold rush, russian settlement in california, sutter's fort, old bale mill in napa valley, boothe state park, lesson plan, fourth grade geography, college links,, fort ross, napa history, napa publisher, earthquake 1906, jessamyn west, college links, early california settlers, the donner party, bear flag revolt, sam brannan, calistoga, earthquake of 1906, indian currency and barter, sonoma, californios, napa valley, hitmill, related links
- 48: California History is a compiled California history index for students and teachers, and Californians wanting to know more about California history. The page has indexed resources for the Bear Flag Revolt, early explorers, California gold rush, California literature, geography, geology and is rich with resources about California.
- 49: california indians, directory, american indians of california, federally recognized tribes, california history, timeline, american indians, native americans, hitmill, alliance of california tribes, mission record of california indians
- 50: This is a compiled directory about the American Indians of California, the California Indians, some of whom were the earliest native americans in California.
- 51: california roads and weather, california road conditions, roads and weather, california skiing, california, weather, road, road conditions, ski, weather forecasts, marine weather, off shore, offshore, wind surfing, severe weather safety tips, aviation weather, highway conditions, road closures, current conditions, highway closures, highway construction, highway

incidents,
accidents,incidents,road,highway,construction,projects,closures,storm
warning,advisory,advisories,severe weather preparation,weather safety tips,wind,river
level,flood stages,napa valley,current watches and
warnings,tourism,travel,trains,transit,truckee,hitmill,tahoe,mountain,sierra,susanville,ny
ck,blue canyon,sonoma,flood,river reports,ski resorts,southern california,traffic cams,ski
conditions,airports,storm,rivers

52: California road conditions and highway conditions, weather forecasts and advisories;
traffic and weather cams, marine weather, hydrometeorological information,
ski
links, severe weather advisories, posted California current watches and warnings, and
severe weather safety tips.

53: comparative religion,world religions,religious comparisons,compare
religion,religious studies,native american

religion,online,religion,society,anthropology,philosophy,native american
spirituality,greek orthodox,roman catholic,seventh day

adventist,methodist,presbyterian,lutheran,baptist,catholic,roman

catholic,catholocism,church of england,church of scotland,mormon,greek
orthodox,eastern

religion,buddhism,islam,hinduism,hitmill,christian,christianity,hindism,judaism,bible,ne
w testament,christ,christian,virtual,index,directory,links

54: Web resources for the study of Comparative Religion and Religious Studies.

55: e-mail,hoax,investigation,primer,tutorial,computer hoax tutorial,hoax list,computer
hoaxes,web hoax,e-mail hoax,hitmill

56: This computer hoax tutorial will help you to not perpetuate hoaxes by forwarding
them in e-mail to anyone else without investigating the hoax threat to find out if it is on
the hoax list.

57: image fill,lettering,font,text,photoShop tutorial,hitmill,put image into lettering

58: This is a free online PhotoShop tutorial about how to use image fills for text, putting
an image into lettering, text image fill, font fills, text with image.

59: news resources,find news,local news,national news,world news,drudge report,local
news, state newspapers, state news,international

news,state,national,world,internation,news,weather,sports,hitmill

- 60: Internet search tool for finding local news, state news, national and international news, sports, weather and travel information.
- 61: ftp,file transfer protocol,ftp guide,ftp new user,ftp tutorial,ftp directory,what is ftp,introduction to file transfer protocol,introduction to ftp,how to ftp,download ftp,hitmill,ftp search,ftp for students
- 62: This is an FTP guide and tutorial for new FTP users about File Transfer Protocol (FTP) and an FTP tutorial about how to move files such as HTML documents between one computer to another computer over the Internet, using WS_FTP Software. There is an FTP directory included of FTP-related links.
- 63: hexadecimal color codes,hex color code,rgb color,html color code,
hexadecimal colour code,hex colour code,html colour,hitmill color,hitmill,named colors,selected named colors,web authors reference,alphanumeric code colors,windows 3.2 colors
- 64: Selected Named Colors is a reference for Web authors desiring to code the name of a color without using the alphanumeric code number for the color. When using a compounded name or a name with several parts do not use spaces between words, and use lowercase.

- 65: hexadecimal color code reference,hexadecimal color values reference,color reference,colour reference,rgb color values,rgb values list,hex color,hitmill,colour,scheme selector,convert hexadecimal to rgb
- 66: Here are six pages of color reference for hexadecimal color values and a colour scheme selector tool where you can select one color or see a whole color scheme and how the colors would look together. Each page has the hexadecimal color value number with each color and there are several tables of colors and corresponding hexadecimal values. Follow the links to get the full benefit of this color reference. Use the scheme selector tool to convert hexadecimal values to rgb numbers.
- 67: html tutorials for students,
html tutorial,html for beginners,html primer,web page tutorial,free,online,html table tutorials,audio for Web pages,table tutorial,html table,html sound,free background textures,hex color,hex colour,
web page sound,Web page music,meta tags,
what is the web,history of computers,free backgrounds for Web pages,
ftp tutorial,pdf files,free web space,optimizing for the search engines,
web page promotion,history of the web,hexidecimal color

values,hexidecimal, colour,colour-safe web palette,hex,hexidemical,fp add-
ons,
color-safe web palette,browsers,hitmill,html editors,what is html,advanced
html,front page links,frontpage links,image
fill,lettering,photoshop,
clipart,photoshop lettering tutorial,hex color values,image
in letter,
web images,dynamic html,accept credit cards from your web
site,hitmill,charge.com,

68: HTML Tutorials for Students is HTML for beginners, a primary HTML
site
with new tutorials and resources for formatting text on a Web page,
creating tables,
how to add an image to a Web page, hexadecimal color values,
FTP tutorial, PDF
File tutorial, free images, and more...

69: internet training,technology,students,internet for students,internet primer,internet
tutorial,beginning programming,programming tutorial,computer
programming,kerberos,network security,macintosh computer,mac,networking,computer
networking tutorials,browsers,history of the web,history of computers,simula
language,simula,qbasic programming,photoshop tutorials,html tutorials,math
links,english links,california geology,california
geography,seismology,sports,css,javascript,asp,web services tutorial,apache
server,linux,introduction,primer

70: Internet Technology Training for Students has original tutorials in the field of
computer programming, Web pages and HTML, networking, Internet, Web Graphics,
Web Services, XML, ASP, Web technologies as well as information for developers.
There is a college courses resource site now with links to pages about math, English
language, California History, California Geography and Geology, Comparative
Religions, and more... There are pages interesting to college students such as news,
sports, weather and page of interest to all, such as the Severe Weather Preparation and
Safety Tips.

71: introduction to the internet,internet primer,internet tutorial,hitmill,introduction to
email,introduction to browsers,search tutorial,history of the web,history of
computers,history of the internet,what is the internet

72: This is a free online Internet tutorial and introduction to the Internet for students.
This page introduces students to the components of the Internet, such as e-mail, browsers,

the Web also known as the World Wide Web, how to search the Web, and a link to the section about HTML or Hypertext Markup Language used to make Web pages. The Internet is a world-wide network of computer networks joined together by communication lines, wireless connections and satellite connections. The Internet is not owned by any government nor individual nor corporation.

73: java programming,free,online,java,tutorials,applets,java articles,java newsgroups,java reference,java primer,java book,free java applet,hitmill

74: This is a compiled directory for Java Programming comprised of sections for Java Tutorials, Java Applets, Java articles, Java newsgroups, Java references, and Java books.

75: hitmill.com,computer-related technologies, webpage,web design,reciprocal linking

76: Hitmill.com's reciprocal linking partners having computer-related technologies, businesses and services, of interest to students.
<META NAME=

77: photoshop tutorial,fill text with pattern-fill,hitmill,pattern-fill lettering,adobe creative suite 4

78: PhotoShop Tutorial: Filling text with a pattern-fill.

79: photoshop tutorial,lettering and gradient fills,photoshop primer,lettering tutorial,photoshop 6,gradient-fill,lettering,text,image-fill,create text-fills from an image using photoshop 6,lettering tutorial,hitmill

80: Free PhotoShop Tutorial: Discussion of PhotoShop lettering, and filling text with an image or with a gradient fill. PhotoShop 6 was used.

81: computer programming resources,computer programming tutorials for students,visual basic 6,vb6,programming resources for students,visual basic,visual

basic.net,ada65,hitmill,overview of programming,programming languages,programming,simula,qbasic programming,visual basic programming,c++ for beginners,c# programming,c programming,history of c,history of c++,c++.net,.net framework,asp,vbscript,javascript programmng,java tutorials,qbasic,ada,simula

82: Computer Programming has primers and tutorials with code examples for students new to programming plus a compiled directory of programming resources and related topics for students, programmers and developers. Our site includes Ada, Simula, C, C++, Visual Basic 6, Visual Basic.NET, QBasic, Java, C#, .NET, ASP.NET; and also the scripting topics of JavaScript, VBScript, and ASP. There is an Introduction to Web

Services for students. The markup languages such as HTML, XML, and XHTML have a separate page for each topic. (See links at bottom of page.)

83: search engine tutorial, what is a search engine, use search tools, search the internet, web, search, internet, where to find search information, education search, music search, game search, gopher search, searching the web, web search, search, usenet news, meta tag, tutorial, hitmill, understanding internet, search

tools, music, mp3, napster, find, searching, tools, gopher, software, rating, page rank

84: This is a free online search engine tutorial for students and includes what is a search engine, where to find search information, how to search, specialized searches, usenet news searches, meta search, Web search, gopher search, music search, how to use search tools, find music and mp3 files, how search engines work, how they find your Web page, how SEs rank pages, and even search engine ratings and search engine information for Web masters. In no time at all you will be searching the Web, the Internet and learning how to find specific search information on the Web.

85: tectonics, earthquake, magnitude, map, shake map, california

earthquake, seismograph, seismogram, earthquakes, report

earthquake, canada, canadian, japan earthquake, world wide earthquakes, temblor, earth tremor, usgs, pacific rim, ring of fire, hitmill, geological hazards, fault zones, earthquake faults, seismicity, seismology, safety tips, earthquake preparation.

86: Seismology-Earthquake quick reference

87: severe weather, preparation, weather, lightning, severe weather preparation, severe weather safety tips, safety, education, tips, lessons, tornado, flood, flash flood, flashflood, dust storm, hitmill, hail, cold, survival, ice storm, severe weather tips, weather tips and safety, winter

storms, preparation, hurricane, earthquake, heat, lightening, thunder, severe, warning, weather, alert, gov, edu, org, online, weather safety tips and information, awareness

88: Severe weather preparation and safety tips about storms, severe cold, winter storms, ice storms, floods, flash floods, dust storms, hail, hurricanes, lightening, tornados, and more.

89: exploring the simula programming language, simula, history of simula, simula 67, hitmill, simula programming, exploring simula, simula-67

- 90: Information and resources about the Simula programming language, from a historical perspective.
- 91: travel corner,travel,hitmill,airfares,ticket reservation,airline ticket,air ticket,train ticket,hotel,motel,reservation,car rental,car hire,hire car,car reservation,low cost air fare,air fare,cheap tickets,language dictionaries,plane ticket,plane reservation,airline reservation,train reservation,money exchanges,travel guides,tv guide,us postage,maps,reservations,
- 92: This is a travel directory about travel topics from hitmill.com's Travel Corner and includes topics and items needed by travelers including reservations for travel, airline tickets, car rentals, hotel, motel, time zones, postal codes, TV listings, cruises, trains, vacation getaways, maps, weather, and more.
- 93: tutorial about fonts,how to install font,find font,install font,font tutorial,download font,laser chrome font,hitmill,font tutorial
- 94: Tutorial About Fonts: about finding a font and installing a font on your computer, or about making fonts.
- 95: tutorial about kerberos,kerberos tutorial,what is kerberos,kerberos authentication,tutorial,computer networking class,primer,kerberos,reference,hitmill,free,online,secret-key,kerberos,single sign on technology,network,authentication,free,cryptography,computer networking,computer security,kerberos v5,ticket,management,log on,password,applications,viewing tickets,protocol
- 96: This is a free, online tutorial about Kerberos, and reference for Computer Networking students, about the Kerberos authentication protocol which uses secret-key cryptography..this is a free kerbos tutorial for students new to computer networking classes.
- 97: visual basic 6,programming,vb,vb6,visual basic,hitmill,resources,free online tutorial,vb6 programming beginner,visual basic,vb6,vb6 programmer,primer,visual basic tutorial,beginning programmer,advanced visual basic,activex,vb6 active x control,vb6 beginner,visual basic developer,vb6 developer,visual basic 6 beginner,vb beginner
- 98: Free Visual Basic 6 tutorials, visual basic 6 beginner's page, visual basic 6 developer section, free visual basic 6 code, online newsgroups, books, maillists, free online vb6

tutorials for programming beginners and additional VB 6.0 programmer and developer resources.

99: web browsers, web browsers tutorial, browser article, web browser tutorial, about browsers, chrome browser, chrome, google chrome, firefox, firefox3, lynx viewer, ncsa mosaic, mosaic browser, internet browser, web accessibility, browser compatibility, speech enabled browser, amaya, w3c, lynx browser, msie, microsoft internet explorer, netscape navigator, ncsa mosaic, hitmill, opera browser, add-ons, ie, msie, browser plug-ins, opera, ie7, ie8, web browser,

100: A Web browser tutorial with discussion about Web browsers, types of browsers, Google's new Chrome browser, Mozilla.com's Firefox3 browser, Microsoft's Internet Explorer 8 Beta, Mosaic, Opera, Lynx, and the Amaya editor/browser of the World Wide Web Consortium... and related links.

101: webservices tutorial, for students, web services tutorial, free, starting point, webservices, asp.net, c#, dotnet, hitmill

102: Webservices for Students is a starting point to learning about this fairly new Web technology, along with the .NET technologies: C#, ASP.NET, VB.NET, Visual C++.NET, VisualStudio.NET, and more.

103: what is dynamic html, what is dhtml, dhtml tutorial, dhtml primer, introduction to dhtml, dynamic html tutorial, dynamic html primer, hitmill, introduction to dynamic html, javascript, cascading style sheets, css-2, css2, css tutorial, javascript tutorials, cssp, css-p

104: Dynamic HTML technologies are explained and explored. Tutorials are included in this site. Dynamic HTML is a group of technologies. See our definition and explanation of the technologies comprising Dynamic HTML, or DHTML.

105: xml tutorials, xml resources, xml references, xml directory, for students, xml primer, free, online, hitmill, introduction to xml

106: XML resources including XML tutorials, XML references, resources, newsgroups, XML news, references and more including the XML schema, XML DOM, XSLT, and SVG.

107: how, to, fill, text, with, fabric, an, image, how to, text-fill, text fill, scanner, scanned, fabric, image, photoshop, images, picture, tutorial, primer, lesson

108: Filling text with a scanned fabric

109: html editors are discussed, links to home sites of the html editors, html coding with html editor, html editor, frontpage 2003, taco html editor, hotdog html editor, golive html editor, microsoft frontpage 2003, microsoft frontpage 2002, hitmill

110: HTML Editor software programs are discussed. Links are given for the home pages of the HTML Editor programs for both PCs and Mac computers.

111: free css tutorial: introduction to cascading style sheets rules, css tutorial, cascading style sheets, css, css 1, css 2, css 3, css primer, cascading style sheets tutorial, css3 working draft, css specification, hitmill

112: An introduction to Cascading Style Sheets rules for Web pages.
<META NAME=

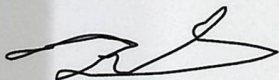
113: introduction to computers, operations of a computer, computer introduction tutorial, hitmill, central processing unit, computer operations

AWCMA

Ethics of use

- AWCMA system is a part of information retrieval revolution.
- Will not harm your web server and utilize your resources.
- We do our best to keep your web server as secure as we can.
- Every update on the system will be mailed directly to all participant.
- No porn web sites or illegal web servers will be a part of AWCMA system.
- Ready to fix and maintain any problem happened in your web server related to system deploy or installation.
- Any problem in our system we will alert you to keep you updated.
- The use of coocrawlers will be fair and under technical considerations.
- The system will be free for the web servers want to participate only during the testing (BETA) period.
- It is not allowed to steal the code of the system to implement such similar applications.
- Our system will also be a light friend on the network as possible.
- We respect your contents and we will be more honor to plant your seeds and generate our own trees.

AWCMA Group



APPENDIX B

Important ^{Code ~~From~~ Modules} pieces of Code


```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * JFrame.java
 *
 * Created on Feb 1, 2009, 5:56:53 PM
 */
package AWCMA;

import jade.core.ProfileImpl;
import jade.core.Runtime;
import jade.wrapper.AgentContainer;

//----- JDK imports -----

import javax.swing.UIManager.*;
import java.util.Hashtable;
import javax.swing.JOptionPane;

/**
 *
 * @author Haitham Hijazi
 */
public class JFrame extends javax.swing.JFrame {
    public AgentContainer c;
    public static float maxm=0;

    public static int sleeptime=0;
    public static Hashtable y=new Hashtable();
    public static Hashtable x=new Hashtable();
    int k=0;
    /** Creates new form JFrame */
    public JFrame() {
        initComponents();
        textField1.setText("Crawl time...");
        textField2.setText("");
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");

        ProfileImpl pr = new ProfileImpl();
        ProfileImpl pr1 = new ProfileImpl();

        try {
            Runtime r = Runtime.instance();
            r.setCloseVM(true);
            pr = new ProfileImpl("127.0.0.1", -1,null);

```



```

c = r.createAgentContainer(pr);
c.acceptNewAgent("NRCA", new namesResolverCoordinator()).start();
c.acceptNewAgent("CrawlerManagerAgent", new Agent1()).start();
c.acceptNewAgent("PerformanceReceiverAgent",
performanceReceiverAgent()).start();
}
catch(Exception e){}
}

```

new

```

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code"> // GEN-
BEGIN: initComponents
private void initComponents() {

```

```

buttonGroup1 = new javax.swing.ButtonGroup();
buttonGroup2 = new javax.swing.ButtonGroup();
jTabbedPane1 = new javax.swing.JTabbedPane();
jPanel1 = new javax.swing.JPanel();
jButton2 = new javax.swing.JButton();
textField2 = new java.awt.TextField();
jButton1 = new javax.swing.JButton();
textField1 = new java.awt.TextField();
label1 = new java.awt.Label();
jRadioButton1 = new javax.swing.JRadioButton();
jRadioButton2 = new javax.swing.JRadioButton();
label2 = new java.awt.Label();
jTextField3 = new javax.swing.JTextField();
jButton4 = new javax.swing.JButton();
jLabel7 = new javax.swing.JLabel();
jPanel3 = new javax.swing.JPanel();
jScrollPane2 = new javax.swing.JScrollPane();
jTextArea1 = new javax.swing.JTextArea();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jSeparator1 = new javax.swing.JSeparator();
jScrollPane1 = new javax.swing.JScrollPane();
jTextArea3 = new javax.swing.JTextArea();
jPanel4 = new javax.swing.JPanel();
jButton3 = new javax.swing.JButton();
jTextField1 = new javax.swing.JTextField();
jTextField2 = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jPanel2 = new javax.swing.JPanel();
jScrollPane3 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
jScrollPane4 = new javax.swing.JScrollPane();
jTextPane1 = new javax.swing.JTextPane();
jLabel4 = new javax.swing.JLabel();
jComboBox1 = new javax.swing.JComboBox();
jLabel1 = new javax.swing.JLabel();

```



```

jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
jMenuItem2 = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jButton2.setText("Start Crawling");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

textField2.setText("textField2");
textField2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        textField2ActionPerformed(evt);
    }
});

jButton1.setText("Set time");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

textField1.setText("textField1");
textField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        textField1ActionPerformed(evt);
    }
});

label1.setFont(new java.awt.Font("Dialog", 1, 12));
label1.setText("Sleep time:");

buttonGroup1.add(jRadioButton1);
jRadioButton1.setFont(new java.awt.Font("Tahoma", 1, 11));
jRadioButton1.setForeground(new java.awt.Color(0, 51, 204));
jRadioButton1.setText("Parallel crawling");
jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton1ActionPerformed(evt);
    }
});

buttonGroup1.add(jRadioButton2);
jRadioButton2.setFont(new java.awt.Font("Tahoma", 1, 11));
jRadioButton2.setForeground(new java.awt.Color(0, 0, 204));
jRadioButton2.setText("Serial crawling");
jRadioButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton2ActionPerformed(evt);
    }
});

```



```

.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(126, 126, 126)
.addComponent(jButton2)))
.addContainerGap(482, Short.MAX_VALUE)
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(86, 86, 86)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addComponent(jButton1)
.addComponent(label2,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(label1,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(textField2,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(76, 76, 76)
.addComponent(textField1,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(34, 34, 34)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jTextField3,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton4)
.addComponent(jLabel7,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(71, 71, 71)
.addComponent(jRadioButton1)
.addGap(18, 18, 18)
.addComponent(jRadioButton2)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
163, Short.MAX_VALUE)
.addComponent(jButton2)
.addGap(70, 70, 70))
);

jTabbedPane1.addTab("Configurations", jPanel1);

jTextArea1.setColumns(20);
jTextArea1.setRows(5);

```



```

.addGap(18, 18, 18)
.addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 186,
short.MAX_VALUE)
.addGap(88, 88, 88))
);

jTabbedPane1.addTab("Results", jPanel3);

jButton3.setFont(new java.awt.Font("Tahoma", 1, 11));
jButton3.setText("Show performance");
jButton3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton3ActionPerformed(evt);
}
});

jTextField1.setFont(new java.awt.Font("Tahoma", 1, 11));
jTextField1.setText("jTextField1");

jTextField2.setFont(new java.awt.Font("Tahoma", 1, 11));
jTextField2.setText("jTextField2");

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 11));
jLabel5.setText("CPU Usage:");

jLabel6.setFont(new java.awt.Font("Tahoma", 1, 11));
jLabel6.setText("Memory Usage:");

javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(
jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createSequentialGroup()
.addGap(152, 152, 152)
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createSequentialGroup()
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel5)
.addComponent(jLabel6)
.addGap(28, 28, 28)
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
.addComponent(jTextField2, javax.swing.GroupLayout.Alignment.LEADING,
.addComponent(jTextField1, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addGroup(jPanel4Layout.createSequentialGroup()
.addGap(55, 55, 55)
.addComponent(jButton3)))
.addComponent(jLabel6, javax.swing.GroupLayout.Alignment.LEADING,
.addGap(658, short.MAX_VALUE))
);
jPanel4Layout.setVerticalGroup(

```



```

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createSequentialGroup())
.addGap(76, 76, 76)
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel5)
.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE),
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(28, 28, 28)
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE),
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel6))
.addGap(60, 60, 60)
.addComponent(jButton3)
.addContainerGap(294, Short.MAX_VALUE))
);

jTabbedPane1.addTab("Statistics", jPanel4);

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {"Source", null},
        {null, null}
    },
    new String [] {
        "JADE Installation", "User Manual"
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class, java.lang.String.class
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }
});

jScrollPane3.setViewportViewView(jTable1);

jScrollPane4.setViewportViewView(jTextPanel);

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabel4.setText(" Ethics");

javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel2Layout.createSequentialGroup()
.addGap(95, 95, 95)

```



```

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)
        .addComponent(jScrollPane4, javax.swing.GroupLayout.DEFAULT_SIZE, 461,
Short.MAX_VALUE)
        .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel2Layout.createSequentialGroup())
        .addComponent(jLabel4)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
425, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(489, 489, 489))
    );
    jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(jPanel2Layout.createSequentialGroup())
        .addGap(64, 64, 64)
        .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(13, 13, 13)
        .addComponent(jLabel4)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane4, javax.swing.GroupLayout.PREFERRED_SIZE,
184, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(179, Short.MAX_VALUE))
    );

    jTabbedPane1.addTab("System Info", jPanel2);

    jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Websites" }));
    jComboBox1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jComboBox1ActionPerformed(evt);
        }
    });

    jLabel1.setText("jLabel1");

    jMenu1.setText("File");

    jMenuItem1.setText("Exit");
    jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItem1ActionPerformed(evt);
        }
    });
    jMenu1.add(jMenuItem1);

    jMenuBar1.add(jMenu1);

    jMenuItem2.setText("About");
    jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {

```



```

jMenu2ActionPerformed(evt);
}
});

jMenuItem2.setText("About AWCMA");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jMenuItem2ActionPerformed(evt);
}
});
jMenu2.add(jMenuItem2);

jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout
    = new
    javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE, 127,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 138,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
                        598, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                ))
            .addContainerGap(156, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 14,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
                        482, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                ))
            .addContainerGap(262, Short.MAX_VALUE))
);

```



```

pack();
} // </editor-fold> // GEN-END: initComponents

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jButton2ActionPerformed
jTextArea1.setText("");
jTextArea3.setText("");

// TODO add your handling code here:
startContainer();
k++;

} // GEN-LAST:event_jButton2ActionPerformed

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jMenuItem1ActionPerformed
System.exit(0);

// TODO add your handling code here:
} // GEN-LAST:event_jMenuItem1ActionPerformed

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jMenuItem2ActionPerformed
JOptionPane.showMessageDialog(null, "hi", "dt",
JOptionPane.PLAIN_MESSAGE); // TODO add your handling code here:
} // GEN-LAST:event_jMenuItem2ActionPerformed

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jMenuItem2ActionPerformed
JOptionPane.showMessageDialog(null, "Done by:Haitham W.Hijazi\n
Moiad S.Zalloum \n Special Thanks to the Supervisor Dr Radwan Tahboub
\n Eng. Eid Badr ", "About_AWCMA", JOptionPane.PLAIN_MESSAGE);
// TODO add your handling code here:
} // GEN-LAST:event_jMenuItem2ActionPerformed

private void textField1ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_textField1ActionPerformed
textField1.setText(""); // TODO add your handling code here:
} // GEN-LAST:event_textField1ActionPerformed

private void comboBox1ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jComboBox1ActionPerformed
textField1.setText(""); // TODO add your handling code here:
} // GEN-LAST:event_jComboBox1ActionPerformed

private void textField2ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_textField2ActionPerformed
// TODO add your handling code here:
} // GEN-LAST:event_textField2ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jButton1ActionPerformed
sleepTime=Integer.parseInt(JFrame.textField2.getText()); //
// TODO add your handling code here:
}

```



```

    } //GEN-LAST:event_jButton1ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jButton2ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jButton2ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jButton1ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jButton1ActionPerformed

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton3ActionPerformed
        try {
            String webName1=(String)jComboBox1.getSelectedItemAt();

            String containerName1= (String)x.get(jComboBox1.getSelectedItemAt());
            c.acceptNewAgent("PerformanceTesterAgent",
            PerformanceTesterAgent(webName1,containerName1)).start();
            new
        }
        catch(Exception e){}
    } //GEN-LAST:event_jButton3ActionPerformed

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButton4ActionPerformed
        String k;
        k=jTextField3.getText();
        maxm=Float.valueOf(k).floatValue();
    } //GEN-LAST:event_jButton4ActionPerformed

    public void startContainer() {
        String webName=(String)jComboBox1.getSelectedItemAt();

        String containerName= (String)x.get(jComboBox1.getSelectedItemAt());

        try {

            c.acceptNewAgent("CrawlingAndMonitoring"+k,
            Agent2(webName,containerName,y,jRadioButton1.isSelected())).start();
            new

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new JFrame().setVisible(true);
            }
        });
    }

```



```

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.ButtonGroup buttonGroup2;
public static javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
public static javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
public static javax.swing.JComboBox jComboBox1;
public static javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
public static javax.swing.JRadioButton jRadioButton1;
private javax.swing.JRadioButton jRadioButton2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JTabbedPane jTabbedPane1;
public static javax.swing.JTable jTable1;
public static javax.swing.JTextArea jTextArea1;
public static javax.swing.JTextArea jTextArea3;
public static javax.swing.JTextField jTextField1;
public static javax.swing.JTextField jTextField2;
public static javax.swing.JTextField jTextField3;
public static javax.swing.JTextPane jTextPane1;
private java.awt.Label label1;
private java.awt.Label label2;
public static java.awt.TextField textField1;
public static java.awt.TextField textField2;
// End of variables declaration//GEN-END:variables

```

```

}

```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package AWCMA;

```

```

import jade.domain.JADEAgentManagement.*;
import jade.domain.mobility.MobilityOntology;

```

```

import jade.content.*;
import jade.content.lang.sl.SLCodec;
import jade.content.onto.basic.Action;
import jade.content.onto.basic.Result;

```

```

import jade.core.AID;
import jade.core.Agent;
import jade.core.ContainerID;
import jade.core.behaviours.*;

```

```

import jade.lang.acl.*;

```

```

//----- JDK imports -----
-----

```

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import javax.swing.UIManager.*;
import java.util.*;
import jade.core.Location;

```

```

/**

```

```

 *

```

```

 * @author Haitham Hijazi

```

```

 */

```

```

public class Agent2 extends Agent {
    boolean repeatFlag=false;
    String tempWebName=null;

```

```

    Hashtable hashTabelOfContainersName = new Hashtable();

```

```

    int next = 0;

```

```

    int numberOfLocations = 0;

```

```

    int k = 0;

```

```

    String locations[] = new String[10];

```

```

    String webName;

```

```

    String containerName;

```

```

    long time;

```

```

    int sleeptime1;

```

```

    boolean parallelFlag=false;

```



```

public Agent2(String webName, String containerName, Hashtable y, boolean
parallel) {
    this.parallelFlag=parallel;
    this.tempWebName=webName;
    this.hashTabelOfContainersName = y;
    this.webName = webName;
    this.containerName = containerName;
}

@Override
public void setup() {
    sleeptime1=JFrame.sleeptime;

    getContentManager().registerLanguage(new SLCodec());
    getContentManager().registerOntology(MobilityOntology.getInstance());
    jade.util.leap.Iterator it = Locations();
    String locationString;
    Location location;
    java.util.List list = new ArrayList();
    while (it.hasNext()) {
        location = (Location) it.next();
        locationString = location.getName();
        if (!locationString.equals("Main-Container")           &&
!locationString.equals(this.here().getName())               &&
!locationString.equals(containerName)) {
            locations[k++] = locationString;
        }
    }

    numberOfLocations = k;

    time = System.currentTimeMillis();
    sleeptime1 = JFrame.sleeptime;
    System.out.print(sleeptime1);
    doWait(20);

    try {

        ContainerID loc = new ContainerID(containerName, null);
        this.doMove(loc);

    } catch (Exception e) {
    }

}

@Override
public void afterMove() {
    int j = 0;

```



```

public Agent2(String webName, String containerName, Hashtable y, boolean
parallel) {
    this.parallelFlag=parallel;
    this.tempWebName=webName;
    this.hashTabelOfContainersName = y;
    this.webName = webName;
    this.containerName = containerName;
}

@Override
public void setup() {
    sleeptime1=JFrame.sleeptime;

    getContentManager().registerLanguage(new SLCodec());
    getContentManager().registerOntology(MobilityOntology.getInstance());
    jade.util.leap.Iterator it = Locations();
    String locationString;
    Location location;
    java.util.List list = new ArrayList();
    while (it.hasNext()) {
        location = (Location) it.next();
        locationString = location.getName();
        if (!locationString.equals("Main-Container")
!locationString.equals(this.here().getName())
!locationString.equals(containerName)) {
            locations[k++] = locationString;
        }
    }

    numberOfLocations = k;

    time = System.currentTimeMillis();
    sleeptime1 = JFrame.sleeptime;
    System.out.print(sleeptime1);
    doWait(20);

    try {

        ContainerID loc = new ContainerID(containerName, null);
        this.doMove(loc);

    } catch (Exception e) {
    }

}

@Override
public void afterMove() {
    int j = 0;

```



```

int s = 0;
int h = 0;
int ki = 1;
StringBuffer x = new StringBuffer();
StringBuffer ax = new StringBuffer();

StringBuffer tit = new StringBuffer();
StringBuffer kwl = new StringBuffer();

String p1 = webName.substring(webName.indexOf(".") + 1,
webName.length());
String p2 = webName.substring(0, webName.indexOf("."));

File selectedDirectory = new File("root" + File.separator + p1 +
File.separator + p2);
long b = selectedDirectory.lastModified();

File[] childs1 = selectedDirectory.listFiles();

for (int ix = 0; ix < childs1.length; ++ix) {

File[] childs2 = childs1[ix].listFiles();

for (int kx = 0; kx < childs2.length; ++kx) {

if (childs2[kx].getName().endsWith("htm") ||
childs2[kx].getName().endsWith("html")) {

String temp;

try {

FileReader fr = new FileReader(childs2[kx]);
BufferedReader br = new BufferedReader(fr);
while ((temp = br.readLine()) != null) {
if (temp.contains("href") && temp.contains("http") &&
!temp.contains("javascript")) { //javascript will add additional load
String[] tempString;
tempString = temp.split("\\");
for (int y = 0; y < tempString.length; ++y) {
if (tempString[y].contains("http")) {
if (tempString[y].contains(">")) {
tempString[y] = tempString[y].substring(tempString[y].indexOf(">") + 1,
tempString[y].indexOf("<"));
tempString[y].indexOf("<");
// Statement stmt = con.createStatement();
}
}
x.append("/*++j +*/ " + tempString[y] + "\n");
String x2 = null;
x2 = tempString[y].substring(tempString[y].indexOf("http://") + 7);
String x3 = null;
x3 = x2.substring(0, x2.indexOf("/"));

```



```

ax.append("/***h + */:" + x3 + "\n");

}

}

}

if (temp.contains("TITLE")) {

String[] tempString1;
tempString1 = temp.split("TITLE");

for (int u = 1; u < tempString1.length; u = u + 2)

{
tit.append(++s + ":" +
tempString1[u].substring(tempString1[u].indexOf(">") + 1,
tempString1[u].indexOf("<")) + "\n");

time = System.currentTimeMillis() - time;

}

}

} catch (Exception e) {
}

}

}

try {

File selectedDirectory1 = new File("root" + File.separator + p1 +
File.separator + p2);

File[] childs11 = selectedDirectory1.listFiles();
for (int ix1 = 0; ix1 < childs11.length; ++ix1) {

File[] childs22 = childs11[ix1].listFiles();
for (int kx1 = 0; kx1 < childs22.length; ++kx1) {

```



```

        (childs22[kx1].getName().endsWith("htm")
        childs22[kx1].getName().endsWith("html")) {
    String tempo;

    boolean flag = true;
    String temp1;

    FileReader kw = new FileReader(childs22[kx1]);
    BufferedReader brl = new BufferedReader(kw);
    temp1 = "";
    while ((tempo = brl.readLine()) != null) {
        if (tempo.toLowerCase().contains("<meta")) {
            flag = true;
            if (!(tempo.contains(">"))) {

                while (flag) {
                    temp1 = brl.readLine();

                    if (temp1.contains(">")) {
                        tempo = tempo + temp1;
                        flag = false;
                    } else {
                        tempo = tempo + temp1;
                    }

                }

            }

            if (tempo.toLowerCase().contains("keywords") ||
            tempo.toLowerCase().contains("description"))
            {
                String temp3 = tempo.substring(tempo.indexOf("\"") + 1,
                tempo.length());
                String temp4 = temp3.substring(0, temp3.indexOf("\""));
                temp4.split(",");

                kw1.append(ki++ + ": " + temp4 + "\n");

            }

        }

    }

    }

    }

    }

    } catch (Exception e) {

        ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
        msg.addReceiver(new AID("CrawlerManagerAgent", AID.ISLOCALNAME));

```



```

msg.setContent(x.toString() + "^" +
kwl.toString() + "^" + ax.toString()); + tit.toString() + "^" +
send(msg);

if(!parallelFlag)
{
if ((next != numberOfLocations)) {
webName = (String) hashTabelOfContainersName.get(locations[next]);

ContainerID loc = new ContainerID(locations[next++], null);
doMove(loc);
} else {
next=0;
webName=tempWebName;
doWait(sleeptime1*1000);

ContainerID loc = new ContainerID(containerName, null);
doMove(loc);

}
}
else
{
doWait(sleeptime1*1000);
afterMove();
send(msg);
}

}

public jade.util.leap.Iterator Locations() { // to get the containers
participated
jade.util.leap.Iterator it = null;
try {
sendRequest(new Action(getAMS(), new QueryPlatformLocationsAction()));
MessageTemplate mt =
MessageTemplate.and(MessageTemplate.MatchSender(getAMS()),
MessageTemplate.MatchPerformative(ACLMessage.INFORM));
ACLMessage resp = blockingReceive(mt);
ContentElement ce = getContentManager().extractContent(resp);
Result result = (Result) ce;
it = result.getItems().iterator();
return it;
} catch (Exception e) {
e.printStackTrace();
}
return it;
}

public void sendRequest(Action action) {
ACLMessage request = new ACLMessage(ACLMessage.REQUEST);
request.setLanguage(new SLCodec().getName());
request.setOntology(MobilityOntology.getInstance().getName());

```



```

try {
    getFragmentManager().fillContent(request, action);
    request.addReceiver(action.getActor());
    send(request);
} catch (Exception ex) {
    ex.printStackTrace();
}
}
}

```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package AWCMA;

import java.util.StringTokenizer;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.*;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import java.io.*;
import jade.core.behaviours.*;

/**
 *
 * @author Haitham Hijazi
 */
public class Agent1 extends Agent {

    long time;
    long time1;

    @Override
    public void setup() {

        Behaviour b = new cbeh();
        this.addBehaviour(b);

    }
    public class cbeh extends CyclicBehaviour {

        public void action() {
            int p = 0;
            time = System.currentTimeMillis();

            ACLMessage msg = myAgent.blockingReceive();

            StringTokenizer st = new StringTokenizer(msg.getContent(),
"^");
            JFrame.jTextArea3.setText(/*p++ */ st.nextToken());

            JFrame.jTextArea1.setText("");
            JFrame.jTextArea1.setText(st.nextToken());

            JFrame.jTextArea1.append(st.nextToken());

            time1 = System.currentTimeMillis() - time;
            JFrame.textField1.setText("Crawling completion time:" +
time1 / 1000 + " seconds");

        }

    }
}

```



```

package AWCMA;
//----- non-JDK imports -----

import jade.content.*;

import jade.core.*;
import jade.core.AID;
import jade.core.Agent;
import jade.core.Location;
import jade.core.mobility.*;

import jade.domain.FIPAAgentManagement.*;
import jade.domain.JADEAgentManagement.*;
import jade.domain.introspection.*;
import org.nslclient4j.*;
import jade.lang.acl.ACLMessage;

//----- JDK imports -----

import java.io.File;
/**
 *
 * @author Haitham Hijazi
 */

public class namesResolver extends Agent {

    public Location location;
    public namesResolver() {}
    public namesResolver(Location loc) {
        this.location = loc;
    }

    @Override
    public void setup() {
        doWait(20);

        this.doMove(location);
    }

    @Override
    public void afterMove() {

        NSClient4j client=null;
        try{
            client=new NSClient4j("127.0.0.1");
        }

        catch (NSClient4JException e) {
            System.err.println("Exception Getting Stat:" + e);
        }
    }
}

```



```

String WSNames = getWSNames();
sendMessage(WSNames);
doDelete();
}

public String getWSNames() {
    String data = "";
    File[] files = null;
    File[] files2 = null;
    try {
        File file = new File("root");
        files = file.listFiles();
        files2=files[0].listFiles();
    } catch (Exception e) {}
    data+=files2[0].getName()+"."+files[0].getName();

    data+="%" + this.here().getName();
    return data;
}

public void sendMessage(String msgData) {
    ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
    msg.addReceiver(new AID("NRCA", AID.ISLOCALNAME));
    msg.setContent(msgData);
    send(msg);
}
}

```



```

package AWCMA;
//----- non-JDK imports -----

import jade.content.*;
import jade.content.lang.sl.SLCodec;
import jade.content.lang.sl.SLCodec;
import jade.content.onto.basic.Action;
import jade.content.onto.basic.Action;
import jade.content.onto.basic.Action;
import jade.content.onto.basic.Result;
import jade.content.onto.basic.Result;

import jade.core.Agent;
import jade.core.Agent;
import jade.core.Agent;
import jade.core.Location;
import jade.core.Location;
import jade.core.Location;
import jade.core.behaviours.Behaviour;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.mobility.*;

import jade.domain.FIPAAgentManagement.*;
import jade.domain.JADEAgentManagement.*;
import jade.domain.introspection.*;
import jade.domain.mobility.MobilityOntology;
import jade.domain.mobility.MobilityOntology;
import jade.domain.mobility.MobilityOntology;

import jade.lang.acl.*;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

//----- JDK imports -----

import java.io.*;

import java.util.ArrayList;
import java.util.StringTokenizer;
import java.util.Hashtable;
/**
 *
 * @author Haitham Hijazi
 */
public class namesResolverCoordinator extends Agent {
    public int j = 0;
    public int k = 0;

```



```

@Override
public void setup() {
    getContentManager().registerLanguage(new SLCodec());
    getContentManager().registerOntology(MobilityOntology.getInstance());
    jade.util.leap.Iterator it = Locations();
    String locationString;
    Location location;
    java.util.List list = new ArrayList();
    while (it.hasNext()) {
        location = (Location) it.next();
        locationString = location.getName();
        if (!locationString.equals("Main-Container")
            && !locationString.equals(this.here().getName())) {
            ++k;
        }
    }

    // System.out.print("t1\n");
    Behaviour by = new recBeh();
    this.addBehaviour(by);
    try {
        // System.out.print("t2\n");
        startResolving();
    } catch (Exception e) {}
}

public void startResolving() throws Exception {
    int i = 0;
    jade.util.leap.Iterator it = Locations();
    while (it.hasNext()) {
        Location lo = (Location) it.next();
        if (!lo.getName().equals("Main-Container")
            && !lo.getName().equals(here().getName())) {
            this.getContainerController().acceptNewAgent("NRA " +
                ++i,
                new namesResolver(lo)).start();
        }
    }
}

@Override
public Location here() {
    return super.here();
}

@Override
public void takeDown() {
    System.out.print("taken down");
}

public void done() {
    System.out.println("done()");
}

```



```

    }

    public jade.util.leap.Iterator Locations() { // to get the
containers participated
        jade.util.leap.Iterator it = null;
        try {
            sendRequest(new Action(getAMS(), new
QueryPlatformLocationsAction()));
            MessageTemplate mt =
MessageTemplate.and(MessageTemplate.MatchSender(getAMS()),
MessageTemplate.MatchPerformative(ACLMessage.INFORM));
            ACLMessage resp = blockingReceive(mt);
            ContentElement ce =
getContentManager().extractContent(resp);
            Result result = (Result) ce;
            it = result.getItems().iterator();
            return it;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return it;
    }

    public void sendRequest(Action action) {
        ACLMessage request = new ACLMessage(ACLMessage.REQUEST);
        request.setLanguage(new SLCodec().getName());
        request.setOntology(MobilityOntology.getInstance().getName());
        try {
            getContentManager().fillContent(request, action);
            request.addReceiver(action.getActor());
            send(request);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    class recBeh extends CyclicBehaviour {
        public void action() {

            try {
                ACLMessage msg = myAgent.receive();
                if (msg != null) {
                    j++;
                    StringTokenizer st = new
StringTokenizer(msg.getContent(), "%");
                    while (st.hasMoreTokens()) {

                        String webName = st.nextToken().toLowerCase();
                        String containerName = st.nextToken();
                        JFrame.x.put(webName, containerName);
                        JFrame.y.put(containerName, webName);
                        JFrame.jComboBox1.removeItem(webName);
                        JFrame.jComboBox1.addItem(webName);
                    }
                    if (j == k) {

```



```
JFrame.jLabel1.setText("The web servers.");  
doDelete();  
    }  
  } else {  
    block();  
  }  
} catch (Exception e) {  
  e.printStackTrace();  
}  
}  
}
```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package AWCMA;
import jade.domain.JADEAgentManagement.*;
import jade.domain.mobility.MobilityOntology;

import jade.content.*;
import jade.content.lang.sl.SLCodec;
import jade.content.onto.basic.Action;
import jade.content.onto.basic.Result;

import jade.core.AID;
import jade.core.Agent;
import jade.core.ContainerID;
import jade.core.behaviours.*;

import jade.lang.acl.*;

import org.nslclient4j.*;
import org.nslclient4j.jmx.NSClient4jService;
import java.io.*;
//----- JDK imports -----
-----

import java.io.File;

/**
 *
 * @author Haitham Hijazi
 */
public class PerformanceTesterAgent extends Agent {

    String webName;
    String containerName;
    String CReport;
    String MReport;//memory usage report

    public PerformanceTesterAgent(String webName, String
containerName) {
        this.webName=webName;
        this.containerName=containerName;
    }
    @Override
    public void setup() {
        // getContentManager().registerLanguage(new SLCodec());
        // getContentManager().registerOntology(MobilityOntology.getInstance());
    }

```



```

try {

    ContainerID loc = new ContainerID(containerName, null);
    this.doMove(loc);

} catch (Exception e) {
}

}

@Override

public void afterMove(){
while(true){
    try {

        NSClient4j client = new NSClient4j("127.0.0.1");
        CReport=client.getPerfMonCounter("CPU");
        MReport=client.getPerfMonCounter("\\Memory\\% Committed Bytes In
Use");

        try{
            Thread.sleep(1000);
        }catch(Exception e){System.out.println(e);}

        } catch (NSClient4JException e) {
            System.err.println("Exception Getting Stat:" + e);
        }

        ACLMessage msg1 = new ACLMessage(ACLMessage.INFORM);
        msg1.addReceiver(new AID("PerformanceReceiverAgent",
AID.ISLOCALNAME));
        msg1.setContent(CReport.toString()+"^"+MReport.toString());
        send(msg1);

    }

}

}
}

```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package AWCMA;
import jade.core.Agent;
import jade.lang.acl.ACLMessage;
import jade.core.behaviours.*;
import java.util.StringTokenizer;
import java.awt.*;
import java.applet.*;

import java.io.File;
import java.util.Date;
import jxl.*;
import jxl.write.*;

/**
 *
 * @author Haitham Hijazi
 */
public class PerformanceReceiverAgent extends Agent {

    @Override
    public void setup() {

        Behaviour beh = new cbehaviour();
        this.addBehaviour(beh);

    }

    public class cbehaviour extends CyclicBehaviour{
        String r;
        public void action(){
            ACLMessage msg1 = myAgent.blockingReceive();

            StringTokenizer string = new StringTokenizer(msg1.getContent(),
"\"");

            JFrame.jTextField1.setText("");
            JFrame.jTextField1.setText(string.nextToken());
            JFrame.jTextField2.setText("");
            String s=string.nextToken();

            JFrame.jTextField2.setText(s);

            float k=0;
            k=Float.valueOf(s).floatValue();

            if(k<JFrame.maxm)
{

```



```

javax.swing.JOptionPane.showMessageDialog(null, "In safe mode\n You can
choose the strategy right now", "performance
report", javax.swing.JOptionPane.INFORMATION_MESSAGE);
}

```

```

else
javax.swing.JOptionPane.showMessageDialog(null, "suffering web site " +
JFrame.jComboBox1.getSelectedItem().toString()+ "\n switch on
Cococrawlers", "Alert", javax.swing.JOptionPane.WARNING_MESSAGE);
}
}
}

```

1. Alan Hrydos and Marc Neuk, "Towards a Scalable, Robust Web Crawler".
2. Benl Ghalil and Zohar Shaleh, "Identifying, Assessing and Determining Relevance of Heterogeneous Internet documents", Institute of Informatics, Slovak Academy of Sciences, May 2003.
3. Marc Neuk and Jared L. Wiener, "Towards Fast Search Crawling Yields High Quality Pages", 2001.
4. Soumen Chakrabarti and Manish Vaid, "Fast and Deep Data: Fast and Deep crawling: a new approach to large-scale Web resource discovery", Computer Science and Engineering, Indian Institute of Technology, Bombay, 2000, India.
5. Christos Doukas, Vasilis Prokopiou, Athina Chalki, "Creating a robust, adaptive and selective incremental crawler", University of Patras.
6. Jinghui Chen, Hector Garcia-Molina, "parallel crawler", 2001.
7. Marc Neuk, Alan Hrydos, "High Performance Web Crawling", IAC Research Report I-73, Computer Systems Research, 2002.
8. Developing multi-agent system with JADE, John Wiley & sons, Apr 2007.
9. Graduation project, "Detecting plagiarism university, Elg Bar and Abdullah Alkham, "Plagiarism Detection System Using Mobile Agents (PLADMA)", 2006.
10. Joseph Khalil Ibrahim, "JADE Java Agent Development Framework JADE", November 2007.
11. JIPA, <http://www.jipa.org>, 2007.
12. <http://www.jipa.org>, 2007.
13. <http://www.jipa.org>, 2007.

References

1. Monica Peshave, "How search engines work and a web crawler application"
2. <http://www.wikipedia.com>
3. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, "*Introduction to Information Retrieval*", Cambridge University Press 2008.
4. Dustin Boswell, "Distributed High-performance Web Crawlers", 2003.
5. Allan Heydon and Marc Najork, "Mercator: A Scalable, Extensible Web Crawler".
6. Emil Gatial and Zoltan Balogh, "Identifying, Retrieving and Determining Relevance of Heterogenous Internet. documents", Institute of Informatics, Slovak Academy of Sciences, May 2003.
7. marc Najork and Janet L. Wiener, "Breadth-First Search Crawling Yields High-Quality Pages", 2001.
8. Soumen Chakrabarti and Martin van den Berg and Byron Dom, "Focused crawling: a new approach to topic-specific Web resource discovery", Computer Science and Engineering, Indian Institute of Technology, Bombay, 400076, India.
9. Christos Bouras, Vassilis Pouloupoulos, Athena Thanou,, "creating a polite, adaptive and selective incremental crawler", *University of Patras*.
10. Junghoo Cho, Hector Garcia-Molina, "parallel crawlers", 2001.
11. Marc Najork, Allan Heydon, "High-Performance Web Crawling", SRC Research Report 173, Compaq Systems Research, 2002.
12. Developing multi-agent system with JADE, John wiley & sons, Apr 2007
13. Graduation project, Palestine polytechnic university, Eid Badr and Abdullah Abdeen, "Plagiarism Detection System Using Mobile Agents (PLADMA)", 2008.
14. Ismail Khalil Ibrahim, "UE Java Agent DEvelopment framework JADE", November 2007.
15. FIPA, <http://standards.computer.org/fipa/assets/XC00001K.pdf>.
16. <http://java.sun.com/javase/technologies/index.jsp>
17. <http://www.netbeans.org/features.com>

18. Stan Franklin and Art Graesser, "Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents", Institute for Intelligent Systems University of Memphis, March 1996.
19. R.Tahboub, V.Iazarescu,"Mobile agents in WWW access optimization",ACIDCA-ICMI'2005.
20. Wayne Jansen, Tom Karygiannis, "Mobile Agent Security", National Institute of Standards and Technology, Computer Security Division,2006.
21. http://www.webopedia.com/TERM/F/fault_tolerance.html
22. Odysseas Papapetrou, Stavros Papastavrou, George Samaras," Distributed Indexing of the Web Using Migrating Crawlers",june 2001.
23. Youn'es Hafri, Chabane Djeraba," Dominos: A New Web Crawler's Design" Novemper 2002.
24. Jan Fiedler and Joachim Hammer, "USING THE WEB EFFICIENTLY: MOBILE CRAWLERS", UF Technical Report, Number TR98-007, Gainesville, FL, June 1998.
25. H. Arafat Ali," -Agent System for Specific Domain Search Engine Based on Distributed Classification Approach", *Computers Engineering & Systems Dept., Faculty of Engineering, Mansoura, Egypt, September 2007.*
26. Sarah Koenig, "Examples of web crawler architectures ",November 2007.
27. Gordon Mohr, Michael Stack, Igor Ranitovic, Dan Avery and Michele Kimpton," An Introduction to Heritrix An open source archival quality web crawler", *4th International Web Archiving Workshop,2004.*
28. http://www.googleguide.com/google_works.html
29. M.P.S.Bhatia and Divya Gupta," Discussion on Web Crawlers of Search Engine", *Proceedings of 2nd National Conference on Challenges & Opportunities in Information Technology, Mandi Gobindgarh. March 29, 2008.*
30. Nick Craswell, Jason Haines, Brendan Humphreys, Chris Johnson and Paul Thistlewaite, "aglets: a good idea for spidering", 2002
31. Raquel Trillo, Sergio Ilarri and Eduardo Mena, "Comparison and Performance Evaluation of Mobile Agent Platforms", 2007, university of Zaragoza
32. E.Cortese, F.Quarta, G.Vitaglione, "Scalability and Performance of JADE message transport system",2006.