P. 3622

CONCEPTUAL SEARCH USING ONTOLOGY BASED KNOWLEDGE BASE

PROJECT REPORT

Submitted by

VIGNESH.V.C Reg.No: 0710108056

VIVEK.K Reg.No: 0710108058

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University of Technology, Coimbatore)

COIMBATORE – 641 049

APRIL 2011

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE - 641049

Department of Computer Science and Engineering

PROJECT WORK, April 2011

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**CONCEPTUAL SEARCH USING ONTOLOGY BASED KNOWLEDGE BASE**", the bonafide work of **VIGNESH.V.C , VIVEK.K** who carried out the project work under my supervision.

C. Ramathilagam

[Mrs.C.RAMATHILAGAM, M.E.,]

Project Guide

S. Devaki

[Mrs.S.Devaki, M.E.,]

Head of the Department

The candidates with **University Register Nos. 0710108056 & 0710108058** was examined by us in project viva-voce examination held on 20/04/2011.

S. Devaki

Internal Examiner

20/4/11

External Examiner

DECLARATION

We,

VIGNESH.V.C — **Reg.No: 0710108056**
VIVEK.K — **Reg.No: 0710108058**

Hereby declare that the project entitled **"CONCEPTUAL SEARCH USING ONTOLOGY BASED KNOWLEDGE BASE"**, submitted in partial fulfillment to Anna University as the project work of Bachelor of Engineering (Computer Science and Engineering) degree, is record of original work done by us under the supervision and guidance of Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore.

Place: Coimbatore
Date: 20/04/2011

[Vignesh.V.C]

[Vivek.K]

Project Guided by,

[Mrs.C.Ramathilagam, M.E.,]
Assistant Professor
Department of Computer Science and Engineering,
Kumaraguru College of Technology,
(An Autonomous Institution)
Coimbatore-641 049.

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for enabling us to complete this project.

We express our profound gratitude to our Chairman **Padmabhusan Arutselvar Dr.N.Mahalingam, B.Sc., F.I.E.,** for giving this opportunity to pursue this course.

We would like to thank Dr.S.Ramachandran, **Ph.D.,** *Principal* for providing the necessary facilities to complete our thesis.

We take this opportunity to thank **Dr.S.Thangasamy Ph.D.,** *Dean, Research and Development,* for his precious suggestions. We also thank **Mrs.S.Devaki M.S.,** *HOD,* Department of Computer Science and Engineering, for her support and timely motivation.

We register our hearty appreciation to the Guide **Mrs.C.Ramathilagam, M.E.,** *Assistant Professor,* Department of Computer Science and Engineering, our Project advisor. We thank for her support, encouragement and ideas. We thank her for the countless hours she has spent with us, discussing everything from research to academic choices.

We would like to convey our honest thanks to all **Teaching** staff members and **Non Teaching** staffs of the department for their support. We would like to thank all our classmates who gave us a proper light moments and study breaks apart from extending some technical support whenever we needed them most.

ABSTRACT

With the development of the web, information "Big Bang" has taken place on the Internet. Search engines have become one of the most helpful tools for obtaining useful information over the internet.

However, instead of caring about the semantics of the information, the machine on the current web cares about the location and display of information only. Because of this short coming of such current search techniques this search results by even the most popular search engines cannot produce satisfactory results.

Semantic search has been one of the motivations of the Semantic Web since it was envisioned. We propose a model for the exploitation of ontology-based knowledge bases to improve search over large document repositories and index the document with semantic annotation.

Our model is a combination of keyword based (vector space model) search and the conceptual (semantic) search to form a new type of search which is much faster than the semantic search and has more relevancy than the normal search. For this purpose, our approach includes an ontology-based scheme for the semiautomatic annotation of documents and a retrieval system.

The retrieval model is based on an adaptation of the classic vector-space model, including an annotation weighting algorithm, and a ranking algorithm. Implementation of the system is with multiple domains and results shows clear improvements with respect to keyword-based search.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

OWL	-	Web Ontology Language
RDF	-	Resource Description Framework
HTML	-	Hyper Text Markup Language
XML	-	eXtensible Markup Language

CHAPTER-1

1. INTRODUCTION

The use of ontologies to overcome the limitations of keyword-based search has been put forward as one of the motivations of the Semantic Web since its emergence in the late 1990s. One way to view a semantic search engine is as a tool that gets formal ontology-based queries from a client, executes them against a knowledge base (KB), and returns tuples of ontology values that satisfy the query. These techniques typically use Boolean search models based on an ideal view of the information space as consisting of no ambiguous, no redundant, formal pieces of ontological knowledge. In this view, the information retrieval (IR) problem is reduced to a data retrieval task.

A knowledge item is both a correct or an incorrect answer to a given information request, thus search results are assumed to be always 100 percent precise, and there is no notion of an approximate answer to an information need. While this conception of semantic search brings key advantages already, our work aims at taking a step beyond. A purely Boolean ontology-based retrieval model makes sense when the whole information corpus can be fully represented as an ontology-driven knowledge base. But, there are well-known limits to the extent to which knowledge can be formalized this way. First, because of the huge amount of information currently available worldwide in the form of unstructured text and media documents, converting this volume of information into formal ontological knowledge at an affordable cost is currently an unsolved problem in general. Second, documents hold a value of their own and are not equivalent to the sum of their pieces no matter how well formalized and interlinked. The replacement of a document by a bag of knowledge atoms

inevitably implies a loss of information value, and although it may be useful to break documents down into smaller information units that can be reused and reassembled to serve different purposes, it is often appropriate to keep the original documents in the system.

Third, wherever ontology values carry free text, Boolean semantic search systems do a full-text search within the string values. In fact, if the string values hold long pieces of free text, a form of keyword-based search is taking place in practice beneath the ontology-based query model since, in a way, unstructured documents are hidden within ontology values, whereby the "perfect match" assumption starts to become arguable, and search results may start to grow in size. While this may be manageable and sufficient for small knowledge bases, the Boolean model does not scale properly for massive document repositories where searches typically return hundreds or thousands of results. Boolean search does not provide clear ranking criteria, without which the search system may become useless if the retrieval space is too big.

In this project, we propose an ontology-based retrieval model meant for the exploitation of full-fledged domain ontologies and knowledge bases, to support semantic search in document repositories. In contrast to Boolean semantic search systems, in our perspective full documents, rather than specific ontology values from a KB, are returned in response to user information needs. The search system takes advantage of both detailed instance-level knowledge available in the KB, and topic taxonomies for classification.

1.1 PROJECT DESCRIPTION

ONTOLOGY

Ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts. It is used to reason about the entities within that domain, and may be used to describe the domain.

Main components of ontologies include:

- Classes
- Attributes
- Relations
- Function terms

1.2 PROJECT COMPONENTS

XML provides an elemental syntax for content structure within documents, yet associates no semantics with the meaning of the content contained within. XML is not at present a necessary component of Semantic Web technologies in most cases, as alternative syntaxes exists, such as Turtle. Turtle is a de facto standard, but has not been through a formal standardization process.

- **XML Schema** is a language for providing and restricting the structure and content of elements contained within XML documents.
- **RDF** is a simple language for expressing data models, which refer to objects ("resources") and their relationships. An RDF-based model can be represented in XML syntax.
- **RDF Schema** extends RDF and is a vocabulary for describing properties and classes of RDF-based resources, with semantics for generalized-hierarchies of such properties and classes.
- **OWL** adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties and characteristics of properties (e.g. symmetry), and enumerated classes.

CHAPTER – 2

SYSTEM DEVELOPMENT

2.1 SYSTEM ANALYSIS

System analyses can be defined as a reduction of an entire system by studying various operation performed and their relationship within system and examination of business activity with a view to identify the problem areas.

2.1.1 EXISTING SYSTEM

An index term, subject term, subject heading, or descriptor, in information retrieval, is a term that captures the essence of the topic of a document. Index terms make up a controlled vocabulary for use in bibliographic records. They are an integral part of bibliographic control, which is the function by which libraries collect, organize and disseminate documents. They are used as keywords to retrieve documents in an information system, for instance, a catalog or a search engine. A popular form of keywords on the web is tags which are directly visible and can be assigned by non-experts also. Index terms can consist of a word, phrase, or alphanumerical term. They are created by analyzing the document either manually with subject indexing or automatically with automatic indexing or more sophisticated methods of keyword extraction. Index terms can either come from a controlled vocabulary or be freely assigned.

Keywords are stored in a search index. Common words like articles (a, an, the) and conjunctions (and, or, but) are not treated as keywords because it is

inefficient to do so. Almost every English-language site on the Internet has the article "the", and so it makes no sense to search for it. The most popular search engine, Google removed stop words such as "the" and "a" from its indexes for several years, but then re-introduced them, making certain types of precise search possible again.

DRAWBACKS

Web 2.0 websites allow users to do more than just retrieve information. By increasing what was already possible in "Web 1.0", they provide the user with more user-interface, software and storage facilities, all through their browser. This has been called "Network as platform" computing. Users can provide the data that is on a Web 2.0 site and exercise some control over that data. These sites may have an "Architecture of participation" that encourages users to add value to the application as they use it.

2.1.2 PROPOSED SYSTEM

The need for the proposed system arises from the limitations of the existing system, which is a software package. The Primary Objective of the proposed system is to achieve competitiveness and oneness of the software developing system.

Semantic Web application areas are experiencing intensified interest due to the rapid growth in the use of the Web, together with the innovation and renovation of information content technologies. The Semantic Web is regarded as an integrator across different content, information applications and systems, it

also provides mechanisms for the realization of Enterprise Information Systems. The rapidity of the growth experienced provides the impetus for researchers to focus on the creation and dissemination of innovative Semantic Web technologies, where the envisaged 'Semantic Web' is long overdue. Often the terms 'Semantics', 'metadata', 'ontologies' and 'Semantic Web' are used inconsistently. In particular, these terms are used as everyday terminology by researchers and practitioners, spanning a vast landscape of different fields, technologies, concepts and application areas. Furthermore, there is confusion with regard to the current status of the enabling technologies envisioned to realize the Semantic Web. In a paper presented by Gerber, Barnard and Van der Merwe the Semantic Web landscape is charted and a brief summary of related terms and enabling technologies is presented. The architectural model proposed by Tim Berners-Lee is used as basis to present a status model that reflects current and emerging technologies

2.2 DEVELOPMENT ENVIRONMENT

2.2.1 HARDWARE CONFIGURATION

Processor Name	: Pentium IV
Processor Speed	: 1.7 GHz
Memory (RAM)	: 256 MB
Hard Disk	: 10 GB

2.2.2 SOFTWARE CONFIGURATION

Operating System	-	Windows XP and above
Software Tools	-	Microsoft Visual studio 2010 (C#.Net)
Application Server	-	Internet Information Server
Datastore	-	XML

2.2.3 SOFTWARE DESCRIPTION

Microsoft Visual studio 2008 (C#.Net):

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native codes together with managed code for all platforms supported by Microsoft Windows.

C# is a multi-paradigm programming language encompassing imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within the .NET initiative C# is one of the programming languages designed for the Common Language Infrastructure. C# is intended to be a simple, modern, general-purpose, object-oriented programming language.

CHAPTER – 3

PROJECT PLAN AND FEASIBILITY STUDY

3.1 TEAM STRATEGY AND WORK

The team consists of 2 members all are qualified for the system to be developed. The system was first designed, coded and implemented in the local host using in-built visual studio's server and testing process was done.

3.2 DEVELOPMENT SCHEDULE

3.2.1 Milestones

Milestones are being established for each and every module to improve the product visibility. It enhances the development process to become more tangible. It exposes errors, which help in improving the product quality and increase project communication. In our application it has been done sub-module wise.

3.2.2 Reviews

Review issues lists, are prepared to identify problem area within the product. As a programmer we do the following reviews.

- Critical Design Review
- Source code Review
- Acceptance Test Review

3.3 FEASIBILITY ANALYSIS

The feasibility study is very rough analysis of the viability of a project. It is however a highly desirable checkpoint that should be completed before committing to more resources. Feasibility study is conducted to obtain an overview of the problem and to roughly whether feasible solutions exist prior to committing substantial resources to a project.

The primary objective of a feasibility study is to assess three types of feasibility.

- Operational feasibility
- Technical feasibility
- Economic feasibility

3.3.1 Operational feasibility

Operational feasibility study is must, because it ensures that that the project implements in the organization work the feasibility should be high.

The Operational feasibility is high in this project as it allows to register for mark updating, result viewing and provides good interface, which is easy and friendly for the user to use it.

3.3.2 Technical feasibility

Technical feasibility analysis makes a comparison of the level of technology available and the same is required for the development of the product. The level of technology accounts for factors such as the programming language, the machine environment, the programming practices and the software tools.

Resource availability such as Pentium 3 processor with 128MB RAM, software and tools required for the project are available at the organization. Hence it is technical feasible.

3.3.3 Economic feasibility

This is the most important aspects that has to be critically evaluated. This includes the feasibility study of cost - benefit analysis. This is an assessment of the economic justification for a computer based system project. Most of the software are available in the web. Hence the threat of financial non-feasibility does not exist. It is determined that benefits out beat the cost of implementation and the system is considered to be economically feasible.

CHAPTER – 4

MODULE DESCRIPTION

4.1 Introduction

Our system consists of 6 modules which are listed as follows:

- Crawling
- Document indexing.
- Weighting.
- Keyword based search.
- Ontology document creation.
- Ranking.

4.1.1 CRAWLING

A **Web crawler** is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. It starts with a list of URLs to visit.

As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit.

Search engines use crawling as a means of providing up-to-date data.

4.1.2 DOCUMENT INDEXING

Many of the words in a document do not describe the content, words like *the, is*. By using document indexing those non-significant words (function words) are removed from the document vector, so the document will only be represented by content bearing words.

4.1.3 WEIGHTING

"Term frequency-inverse document frequency" model is used . It specifies how important the term is to the document. The term specific weights in the document vectors are products of local and global parameters.

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ is the number of occurrences of the considered term (t_i) in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j, that is, the size of the document $|d_j|$.

$$\mathrm{idf}_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

Where,

- $|D|$: cardinality of D, or the total number of documents in the corpus
- $|\{j : t_i \in d_j\}|$: number of documents where the term t_i appears (that is $n_{i,j} \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use $1 + |\{j : t_i \in d_j\}|$

Then , $(\text{tf-idf})_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$

4.1.4 KEYWORD BASED SEARCH

The Keyword based search is the oldest way of searching through the web. It finds the keyword from the user query and then searches the web for those particular keywords alone, then displays the results accordingly based on its ranking algorithm.

4.1.5 ONTOLOGY DOCUMENT CREATION

The process of giving meaning to the raw data obtained from the crawling process is termed as Ontology document creation. In our system, the raw data is been classified based on its domain and their relationship are defined.

4.1.6 RANKING

The Ranking algorithm is used to order the unordered results for the user's query .We used Location/Frequency ranking algorithm.

Location: where the term occurs.

Frequency: how often the term occurs.

CHAPTER 5

SYSTEM DESIGN AND IMPLEMENTATION

5.1 SYSTEM DESIGN

System Design is a solution, a "how to" approach to the creation of new system It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. A Design goes through the logical and physical stages of development. Design is a creative process that involves working with the unknown new system, rather than analyzing the existing system. Thus, in analysis it is possible to produce the correct model of existing system.

5.1.1 INPUT DESIGN

Input screens form the primary interfaces between the user and software. It clearly describes the users what type of data should be given. The input screens are designed in such a way that it has a simple and user-friendly layout.

The screens are designed to capture all the necessary and sufficient data to the system. It includes the validations that are to be done during data entry. The data is the basis for information systems. Without data, there is no system, but data must be provided in the format acceptable to the user. A screen is actually a display station that has a buffer for storing data. The main objective of the screen design is for simplicity, accurate and quick data capture or entry

5.1.2 OUTPUT DESIGN

Computer output is the most important and direct source of information to the user. The efficient and intelligible output design improves the system's relationship with the user and help in decision making.

The output should be designed around the requirements of the user. Outputs from the computers are required primarily to communicate the result of processing to the users and to provide a permanent copy of the results for later communication. The entire system appears fine if the produced output is well qualified, otherwise it causes the entire system to fail.

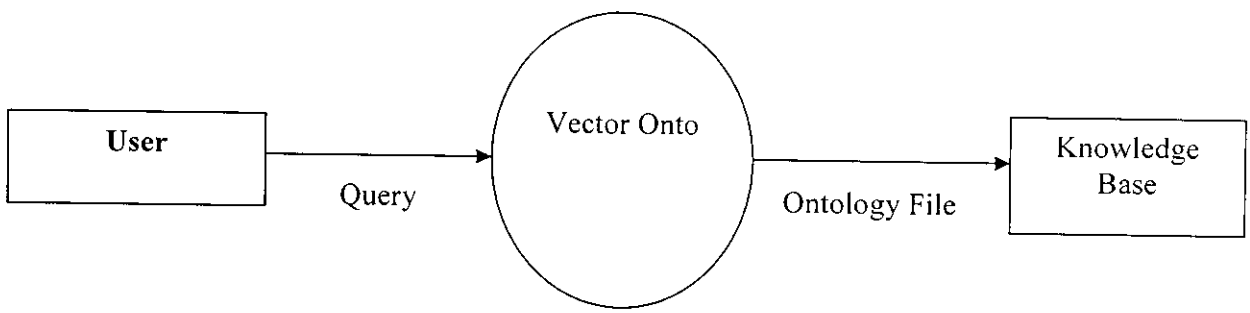
Designing computer output should proceed in an organized, will thought out manner. The right input element ensures that the developed output is qualified one. It makes the users to training themselves easily and effectively. The term output applies to any information produced by an information system, whether printed or displayed.

5.1.3 DATA BASE DESIGN

A database is a collection of interrelated data with minimum redundancy to serve the user quickly and efficiently. The data are stored in tables. It contains the individual records, the sequence in which the records are held on the storage medium and the order in which they may be accessed.

Without data there is no system, but the data must be provided in the right form for input and the information produced must be in a format acceptable to the user.

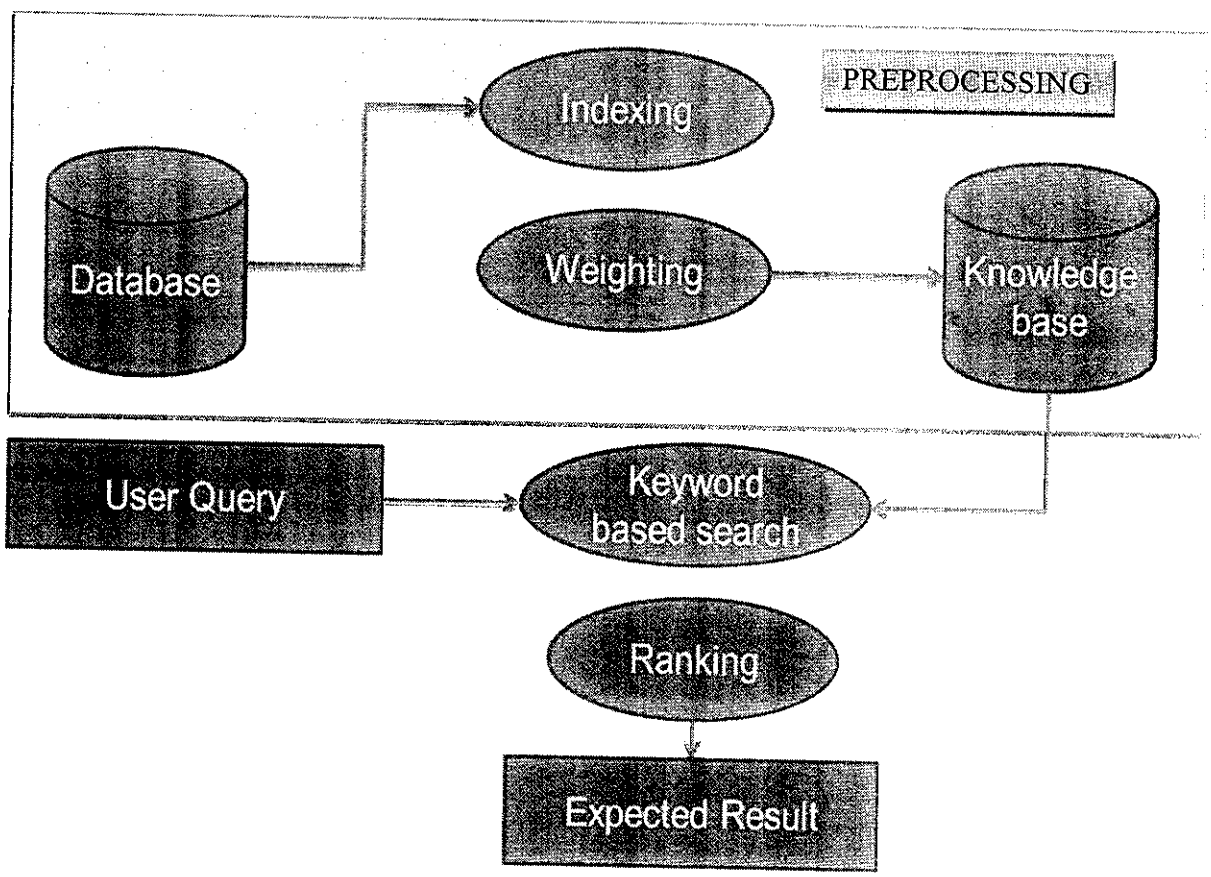
D 1.1 CONTEXT ANALYSIS



D 1.2 DATA FLOW DIAGRAM

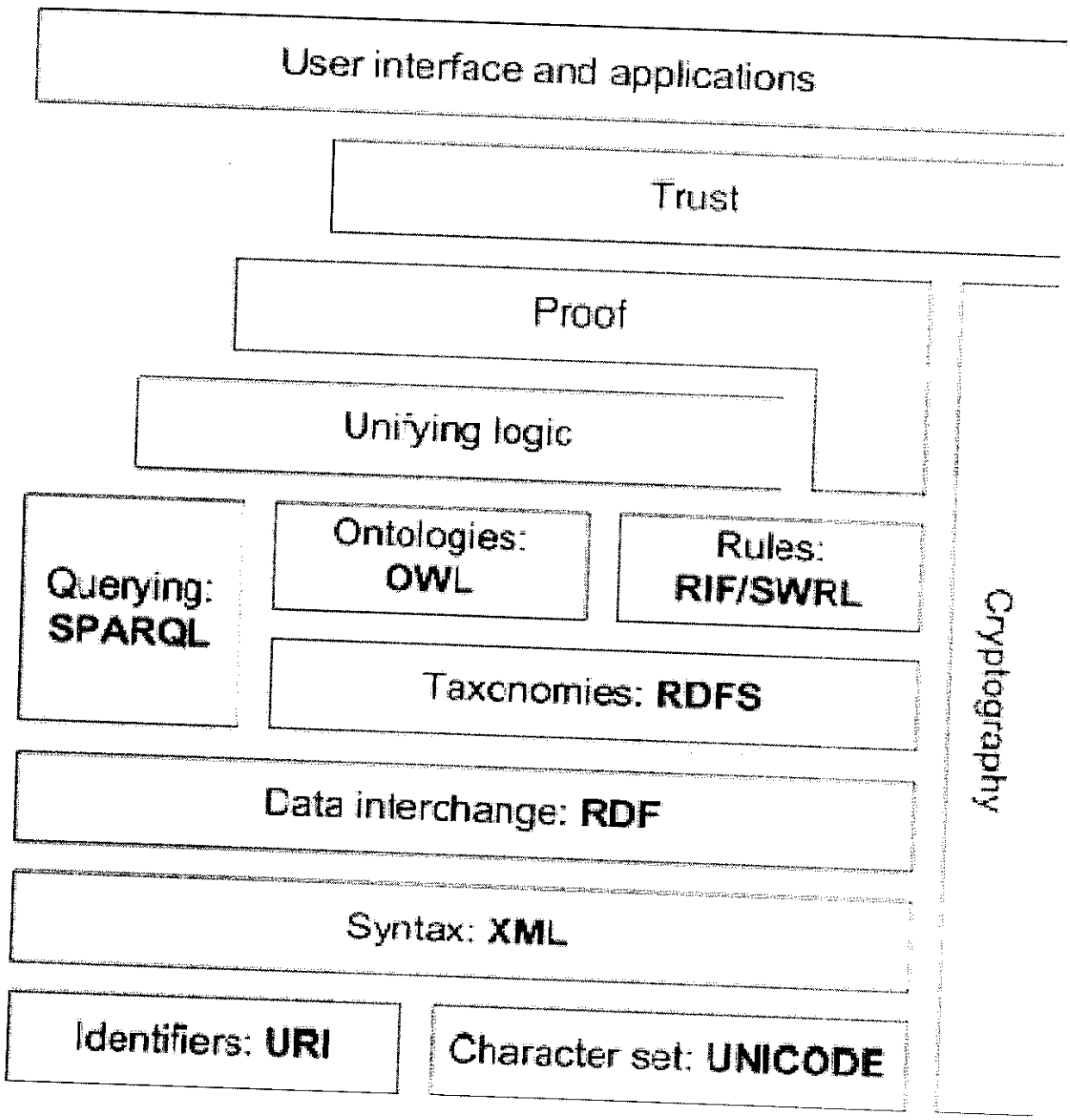
Data Flow Diagram (DFD) is a modeling tool that allows picturing system as a network of functional process to one another by pipelines of data. They are also widely used for representation of external and top-level design specification. The DFD shows the interface between the system and external terminators. Data Flow Diagram is also called as" Bubble Chart".

The bubble represents the process, the line represents the data flow and rectangle represents the entity.



Data Flow Diagram

D 1.3 SEMANTIC WEB STACK



5.2 SYSTEM TESTING

The philosophy behind testing is to find errors. The common view of testing is to bring the program without errors. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Once the source code has been generated, software must be tested to uncover as many errors as possible before delivery to the customer. In order to find the highest possible number of errors, tests must be conducted systematically and test cases must be designed using disciplined techniques.

5.2.1 Verification and Validation

Verification refers to the set of activities that ensures that system correctly implements a specific function. Validation refers to different set of activities that ensure that the system has been built is traceable to customer requirements. Verification and validation encompass a wide array of software quality assurance (SQA) activities that include formal reviews, quality, performance monitoring, documentation review, database review.

Search field

The search field can contain alphabets, numbers and special characters. This validation is checked.

5.2.2 Unit Testing

Unit testing focuses verification effort on the smallest unit of software unit of software design, the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

Test 1:

Procedure:

Either of the two radio button has to be checked.

Solution:

The problem is solved by making the keyword based radio button checked by default.

5.2.3 Integration Testing

Once the modules are tested individually under the testing strategy, it is necessary to put all these modules together-interfacing. It is here that the data can be lost across the interface, one module can have an inadvertent, adverse effect on another.

Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. The objective is to take unit testing modules and build a program structure that has been dictated by design.

5.3 IMPLEMENTATION

Implementation means the process of converting a new or revised system design into an operational one. It is the most crucial stage in achieving a new successful system and in giving a confidence on the new system for the users that it will work efficiently and effectively. In this phase, we build the components either from scratch or by composition. Given the architecture document from the design phase and requirement document from the analysis phase, we can build exactly what has been requested.

This phase deals with issues of quality, performance, baselines, libraries and debugging. The end deliverable is the product itself. There are three types of implementation:

1. Implementation of a computer system to replace a manual system.

2. Implementation of new computer system to replace an existing one.

3. Implementation of a modified application to replace an existing one, using the same computer.

Implementation of our System comes under third category. At the end of the specific period, the system performance and the reliability are tested. Implementation is the key stage in achieving a successful new system.

CHAPTER 6

FUTURE DEVELOPMENT

- Modification and enhancement can be made affecting any other part of the program because of the user friendliness and understandability of the project.
- The data screens can be upgraded and menus can be easily added when required. Items can be added to the forms when there comes a necessity of new data.
- The system has much scope in the future and it can be developed to add more features to satisfy the user's request and company's request.
- Further, the dataset can also be extended to be independent of domain.

CHAPTER 7

CONSLUSION

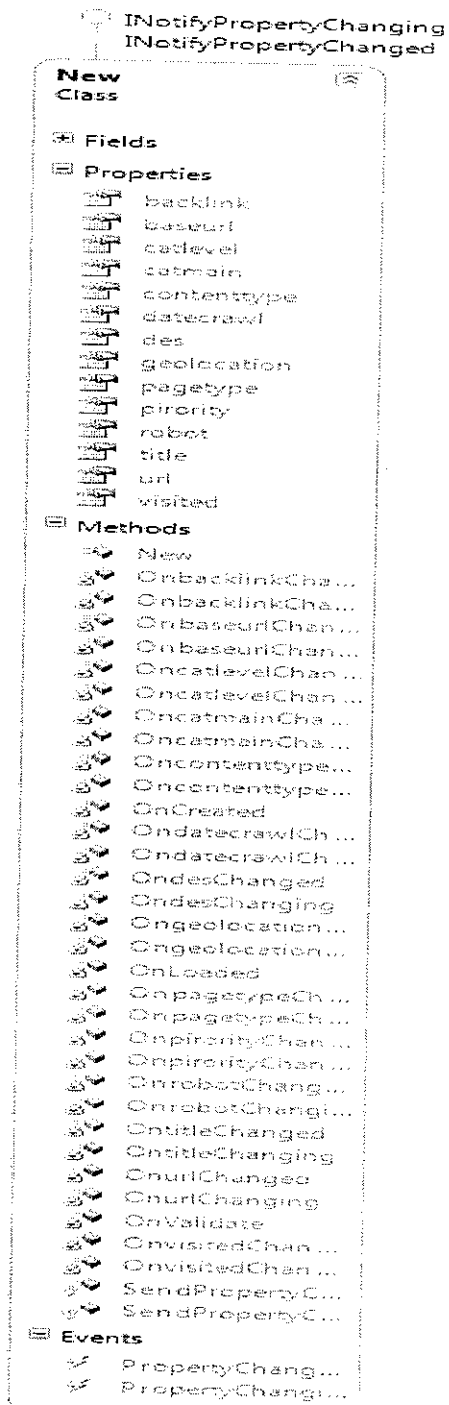
The research presented here started as a continuation of previous work on the construction, exploitation, and maintenance of domain-specific KBs using Semantic Web technologies. While some basic semantic search facilities were included in these prior proposals, there was significant room for improvement because of the rather low level of semantic detail, since the search was essentially based on types of documents and taxonomic classifications. The aim of our current work is to provide better search capabilities which yield a qualitative improvement over keyword-based full-text search, by introducing and exploiting finer-grained domain ontologies.

Our system worked with multiple domain concepts and generates automatic ontology representation of webpage. Our approach can be seen as an evolution of the classic vector-space model, where keyword-based indices are replaced by an ontology-based KB, and a semiautomatic ontology annotation of webpages.

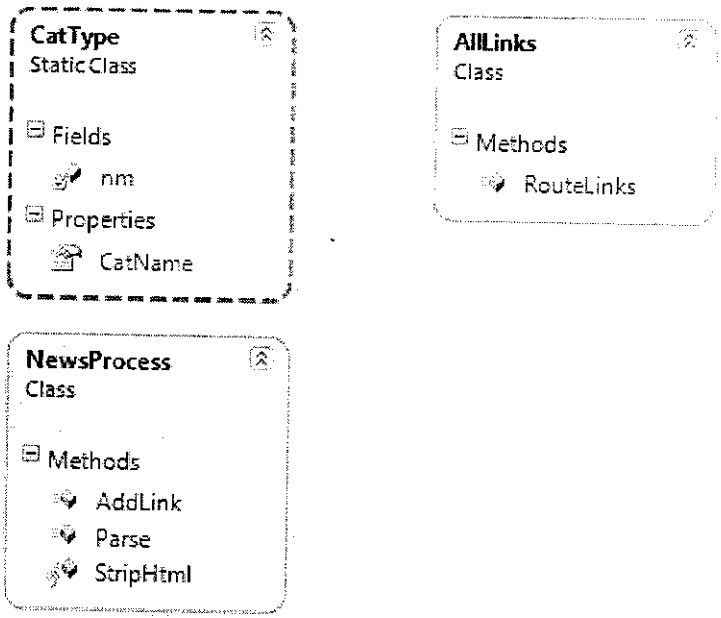
APPENDIX 1

CLASSES AND PROPERTIES

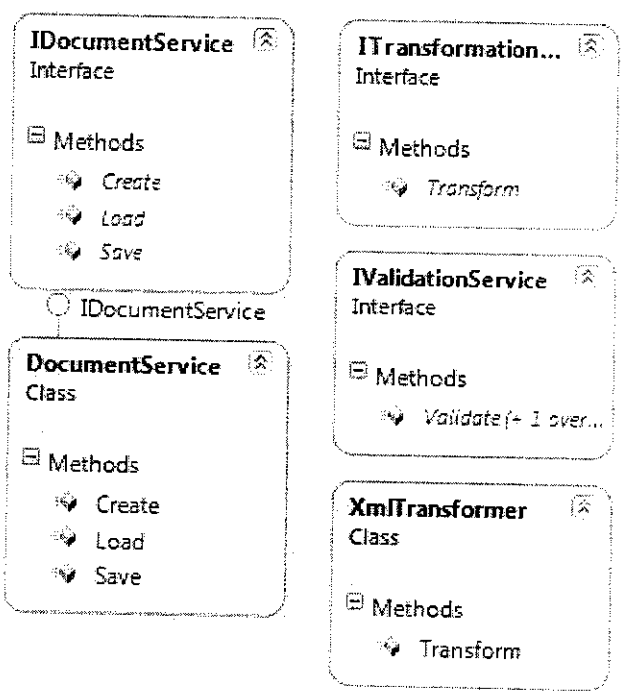
Web Class

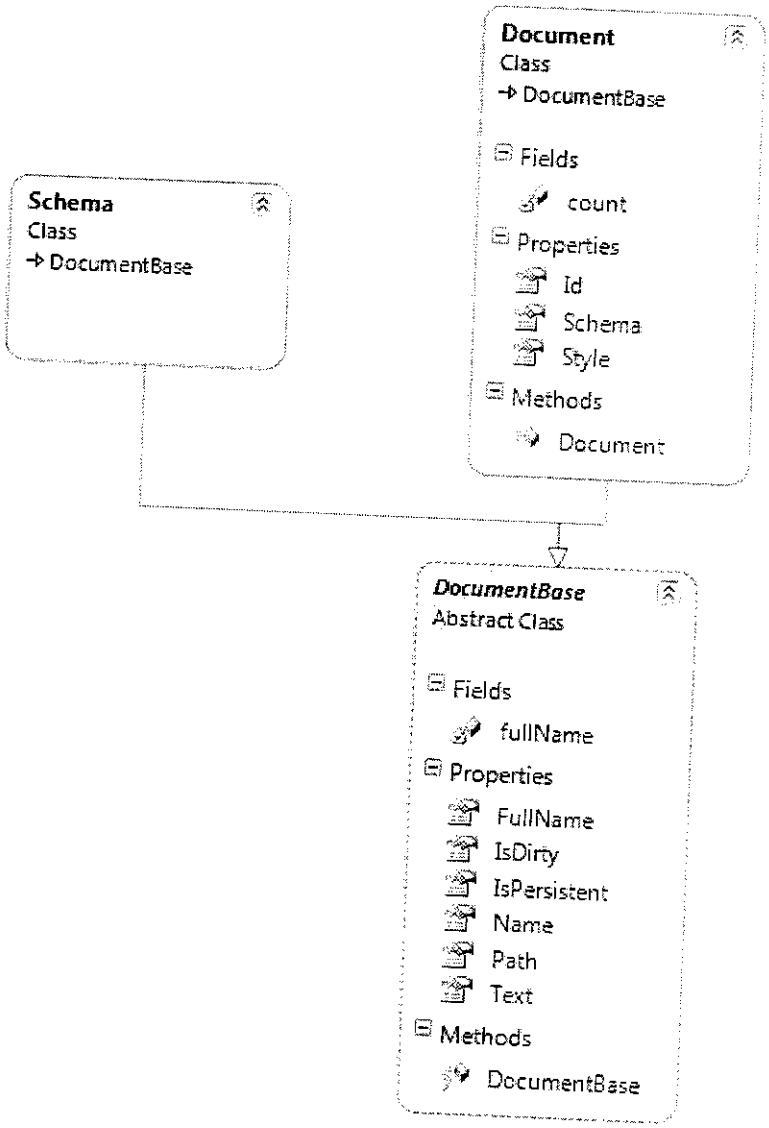


Process



SYNTAX




Document
Class
DocumentBase
Fields
count
Properties
Id
Schema
Style
Methods
Document
Schema
Class
DocumentBase
DocumentBase
Abstract Class
Fields
fullName
Properties
FullName
IsDirty
IsPersistent
Name
Path
Text
Methods
DocumentBase

Classes

rdf:Resource - the class resource, everything
rdfs:Literal - the class of XML literal values, e.g. strings and integers
rdf:XMLLiteral - the class of XML literal values
rdfs:Class - the class of classes
rdf:Property - the class of properties
rdfs:Datatype - the class of RDF datatypes
rdf:Statement - the class of RDF statements
rdf:Alt, rdf:Bag, rdf:Seq - containers of alternatives, unordered containers, and ordered containers (rdfs:Container is a super-class of the three)
rdfs:Container - the class of RDF containers
rdfs:ContainerMembershipProperty - the class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of rdfs:member
rdf:List - the class of RDF Lists
rdf:nil - an instance of rdf:List representing the empty list

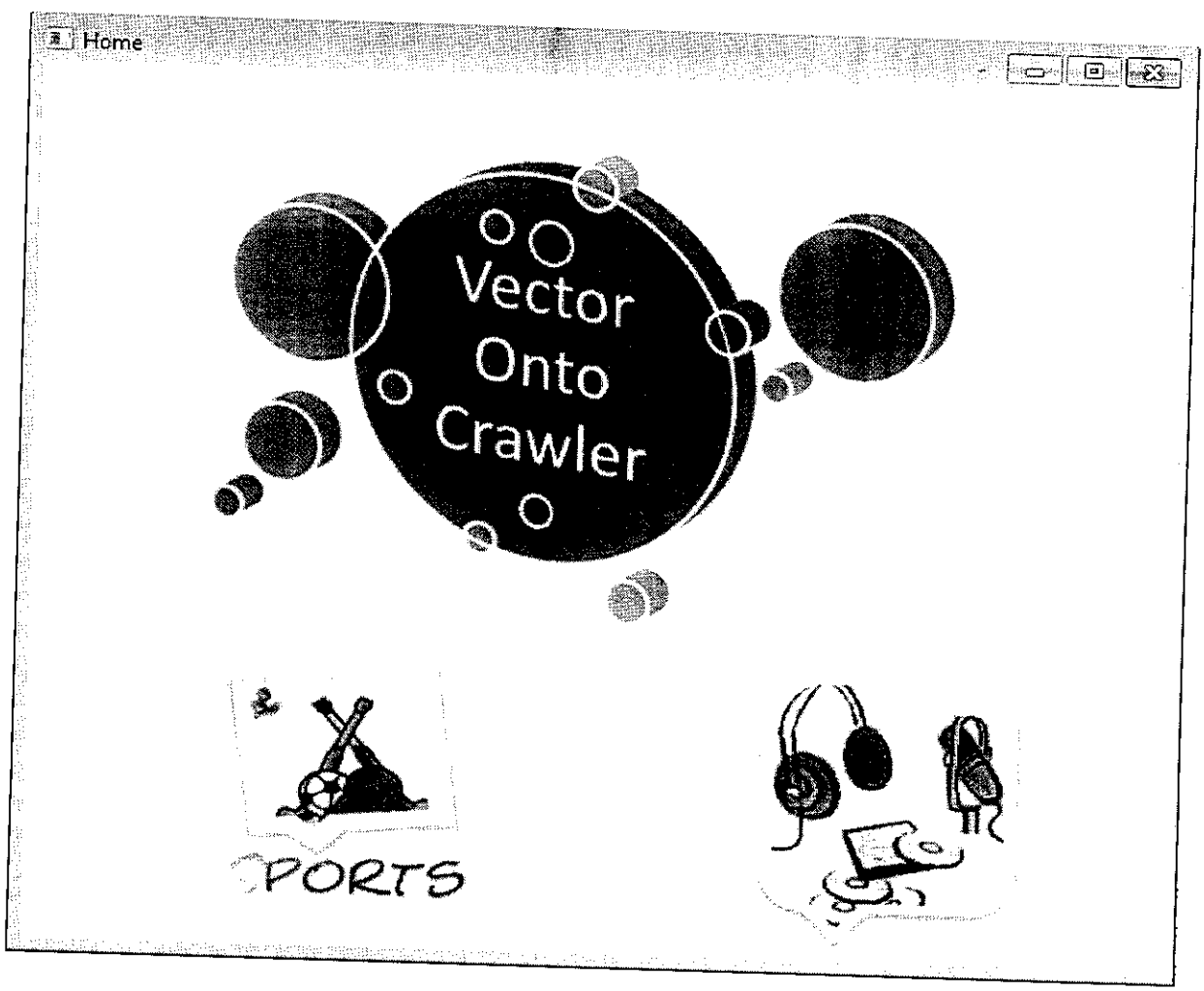
Properties

rdf:type - an instance of rdf:Property used to state that a resource is an instance of a class
rdfs:subClassOf - the subject is a subclass of a class
rdfs:subPropertyOf - the subject is a subproperty of a property
rdfs:domain - a domain of the subject property
rdfs:range - a range of the subject property
rdfs:label - a human-readable name for the subject
rdfs:comment - a description of the subject resource
rdfs:member - a member of the subject resource
rdf:first - the first item in the subject RDF list
rdf:rest - the rest of the subject RDF list after the first item
rdfs:seeAlso - further information about the subject resource
rdfs:isDefinedBy - the definition of the subject resource
rdf:value - idiomatic property used for structured values
rdf:subject - the subject of the subject RDF statement
rdf:predicate - the predicate of the subject RDF statement
rdf:object - the object of the subject RDF statement

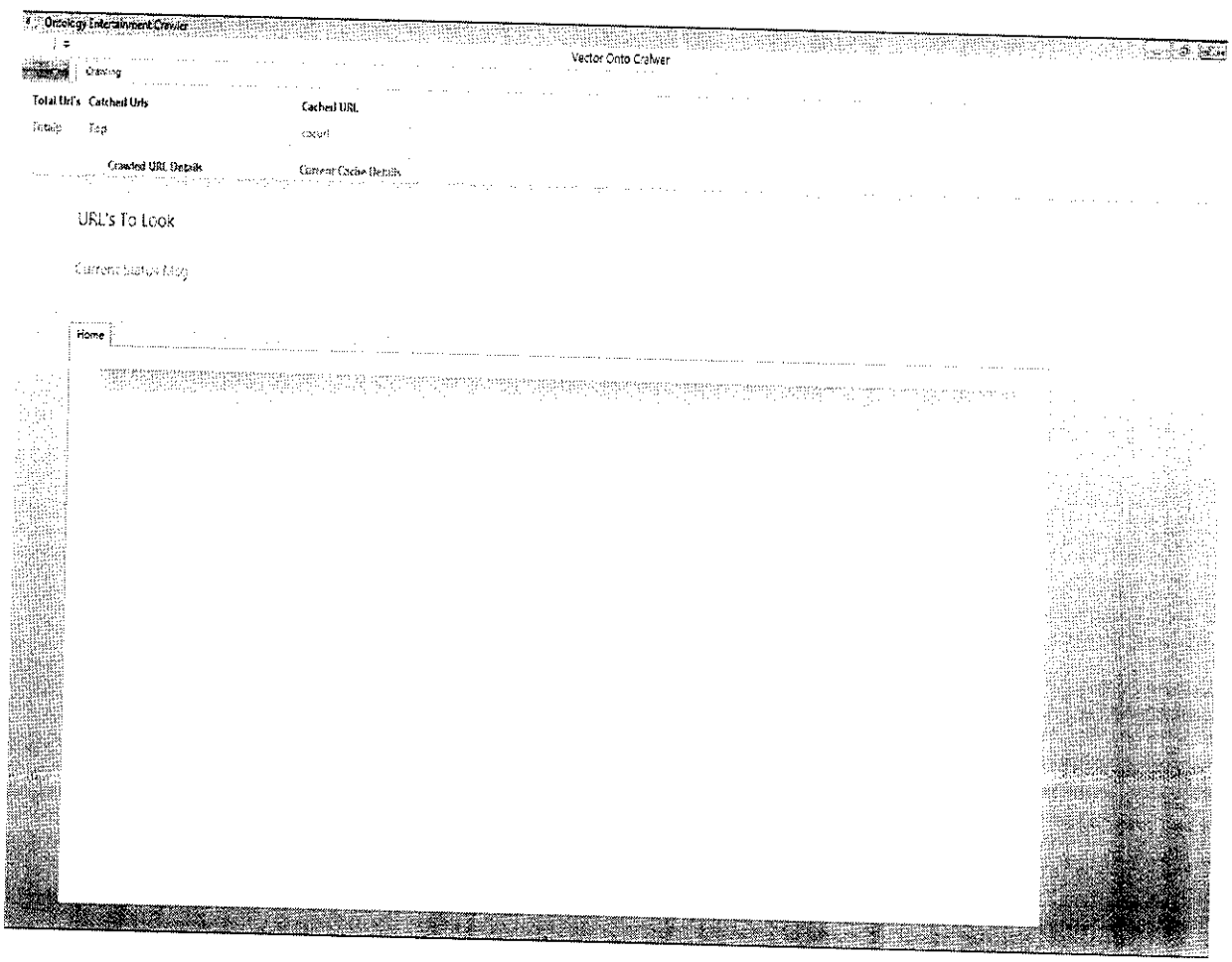
APPENDIX 2

SCREEN SHOTS

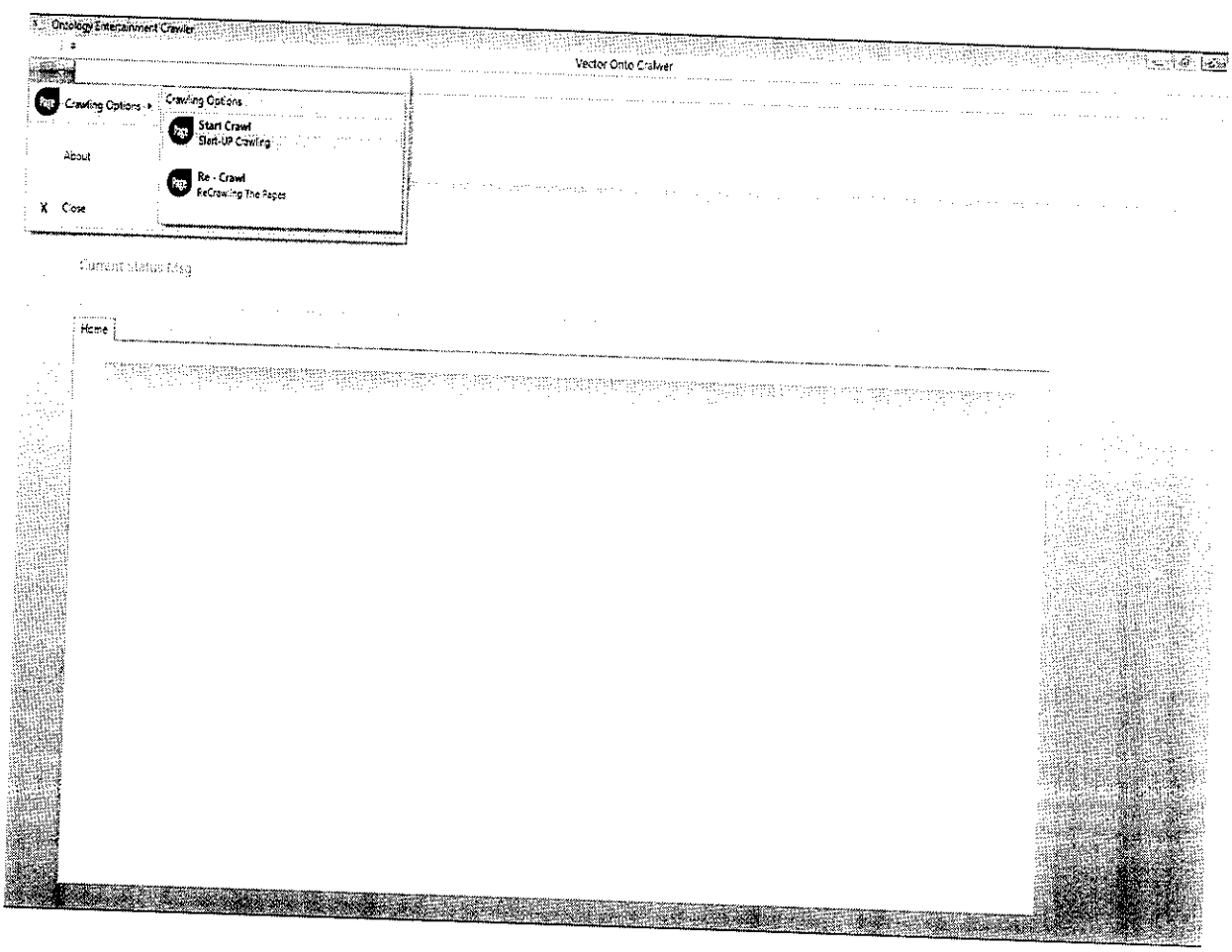
HOME SCREEN



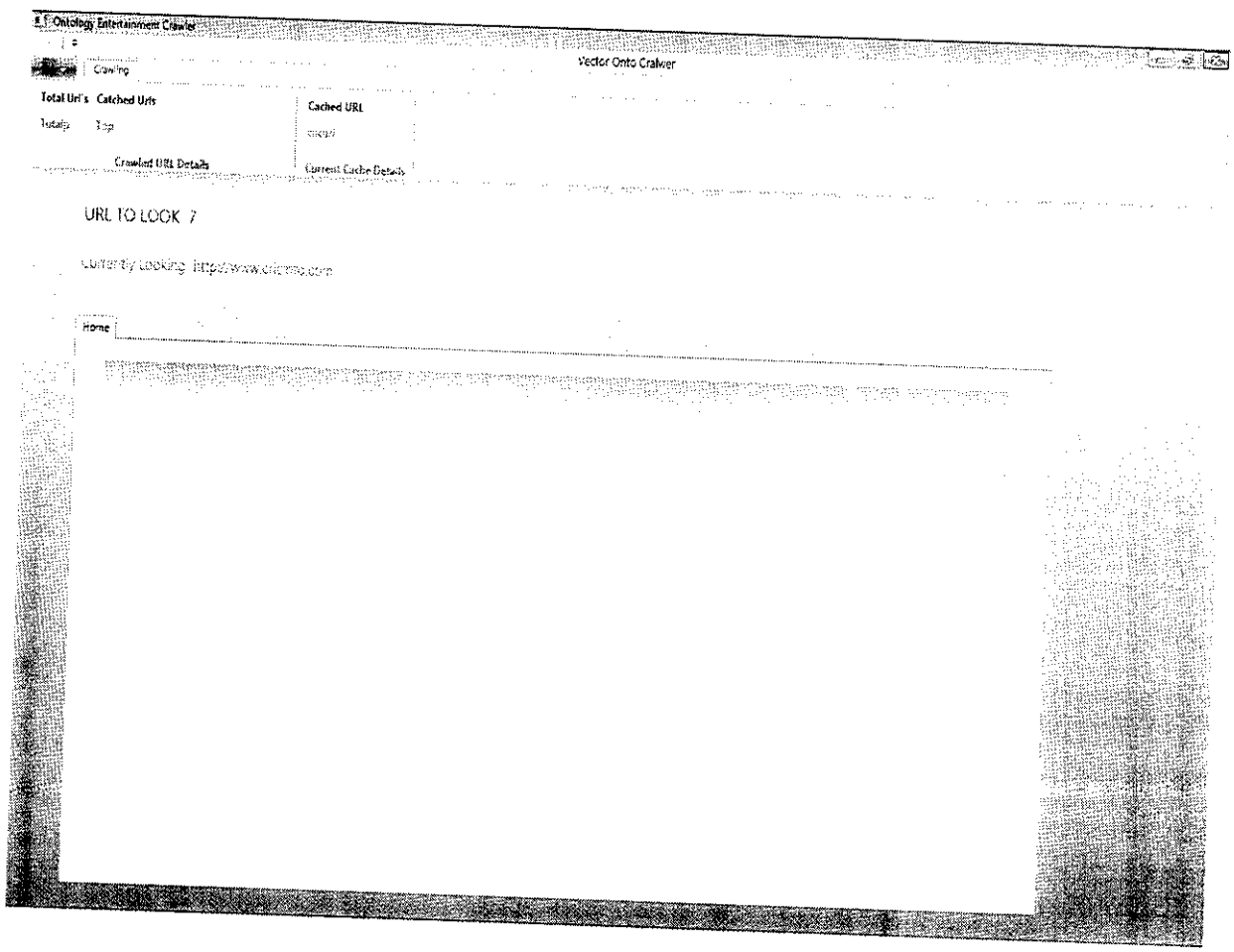
CRAWLING HOME



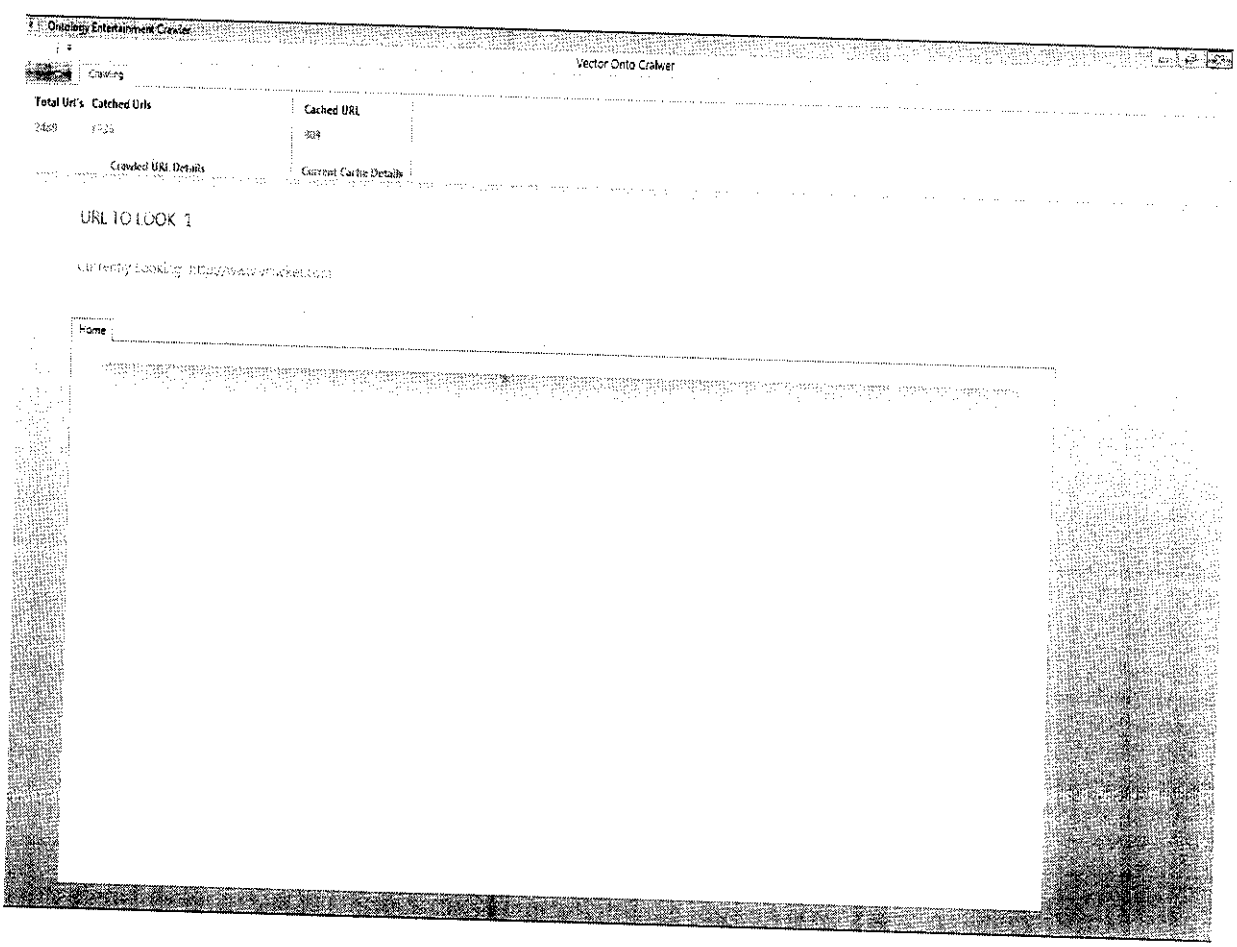
STARTING FIRST PHASE CRAWLING



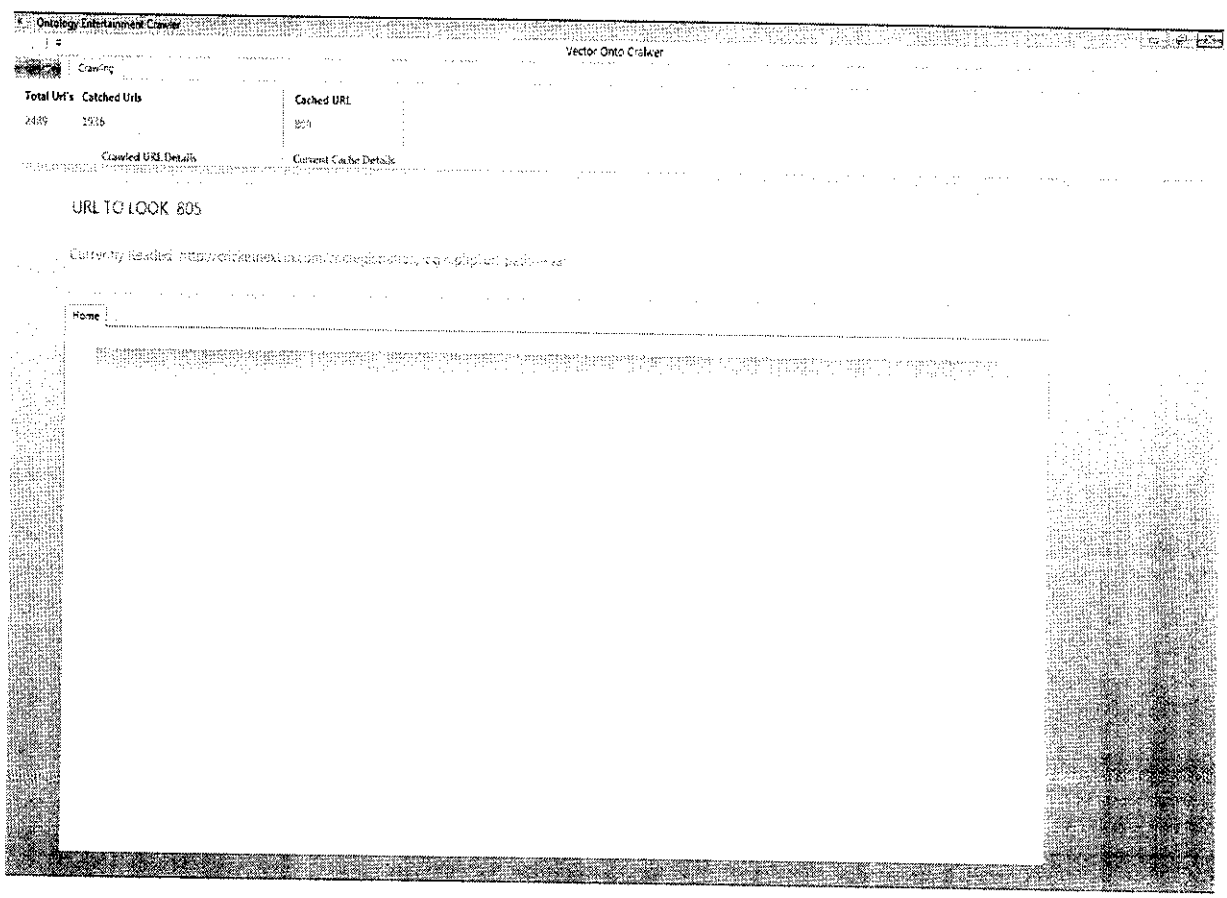
CRAWLING PAGES



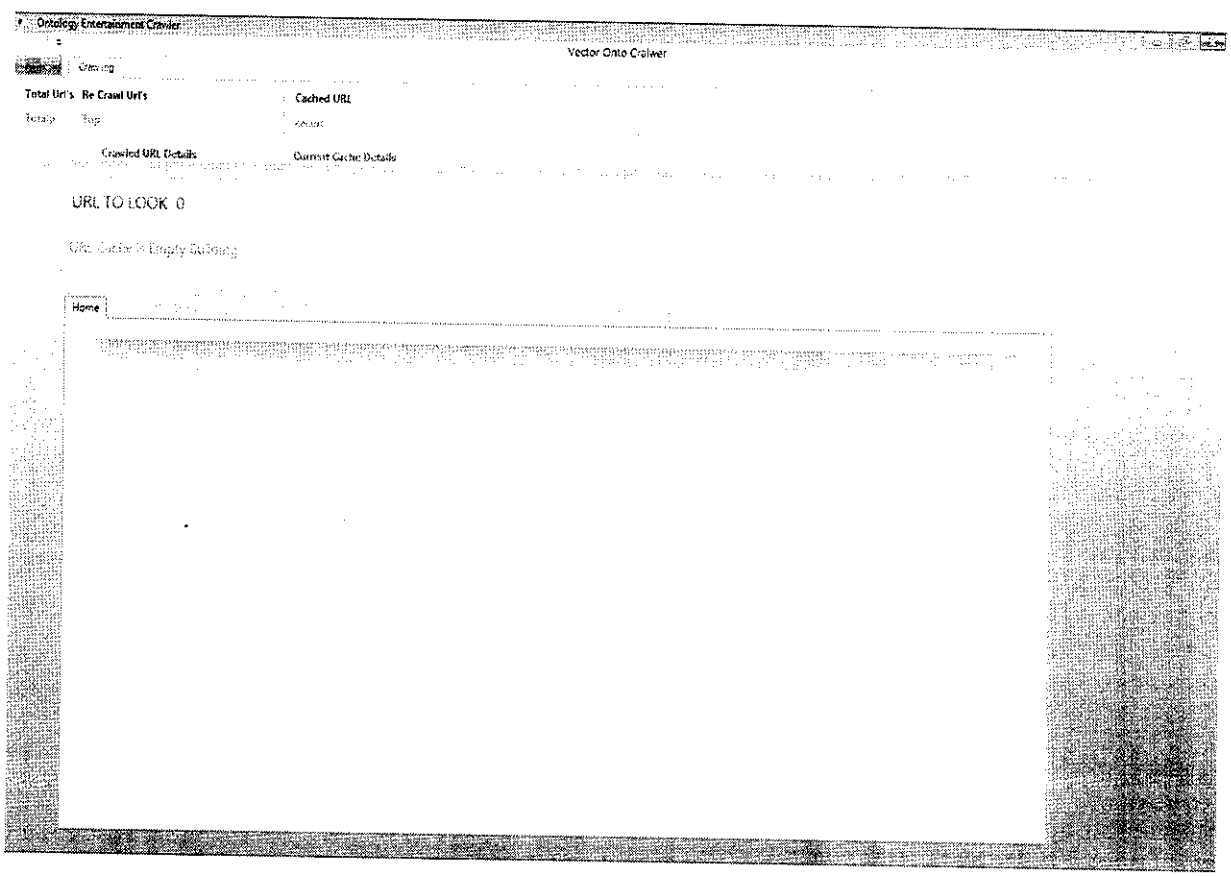
CRAWLING FINISHED PAGES



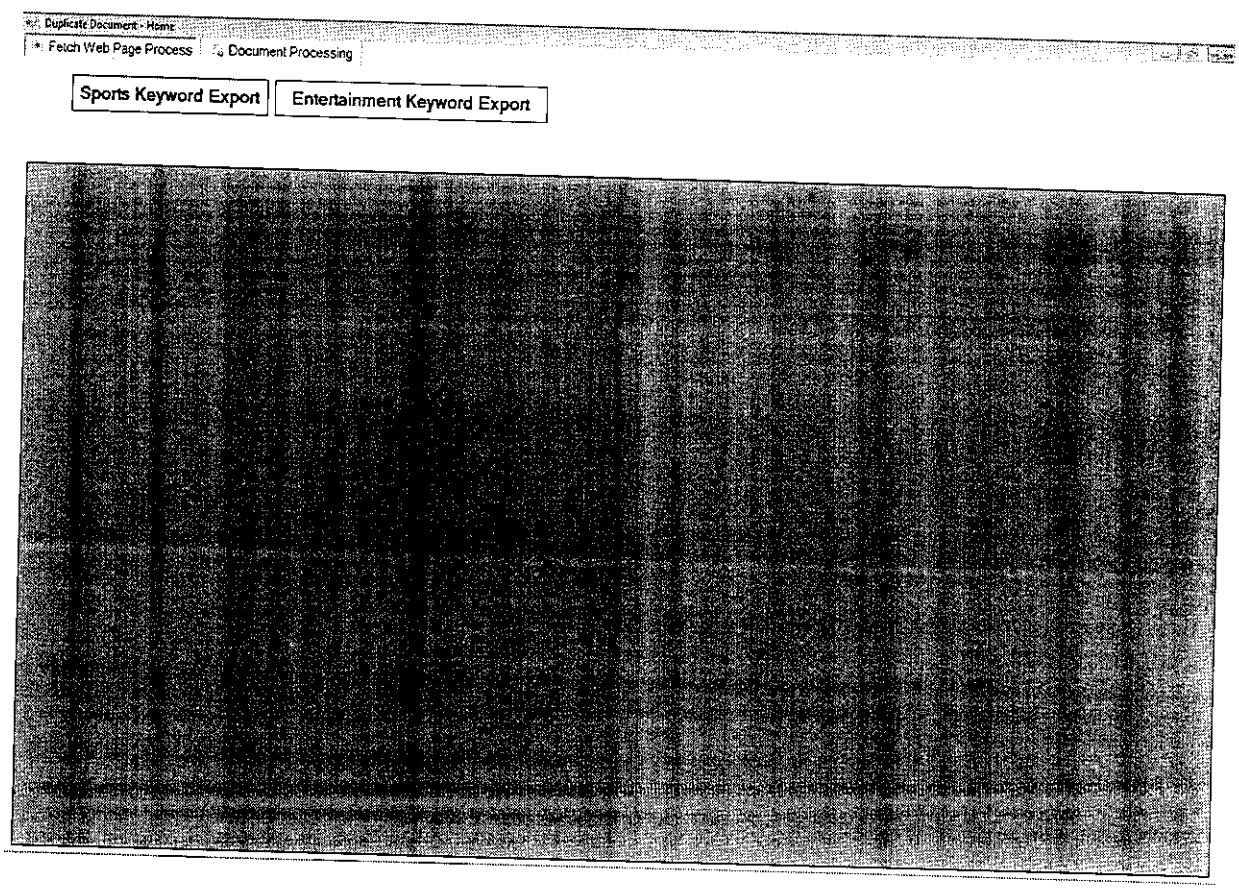
CRAWLING SECOND PHASE OF PAGES



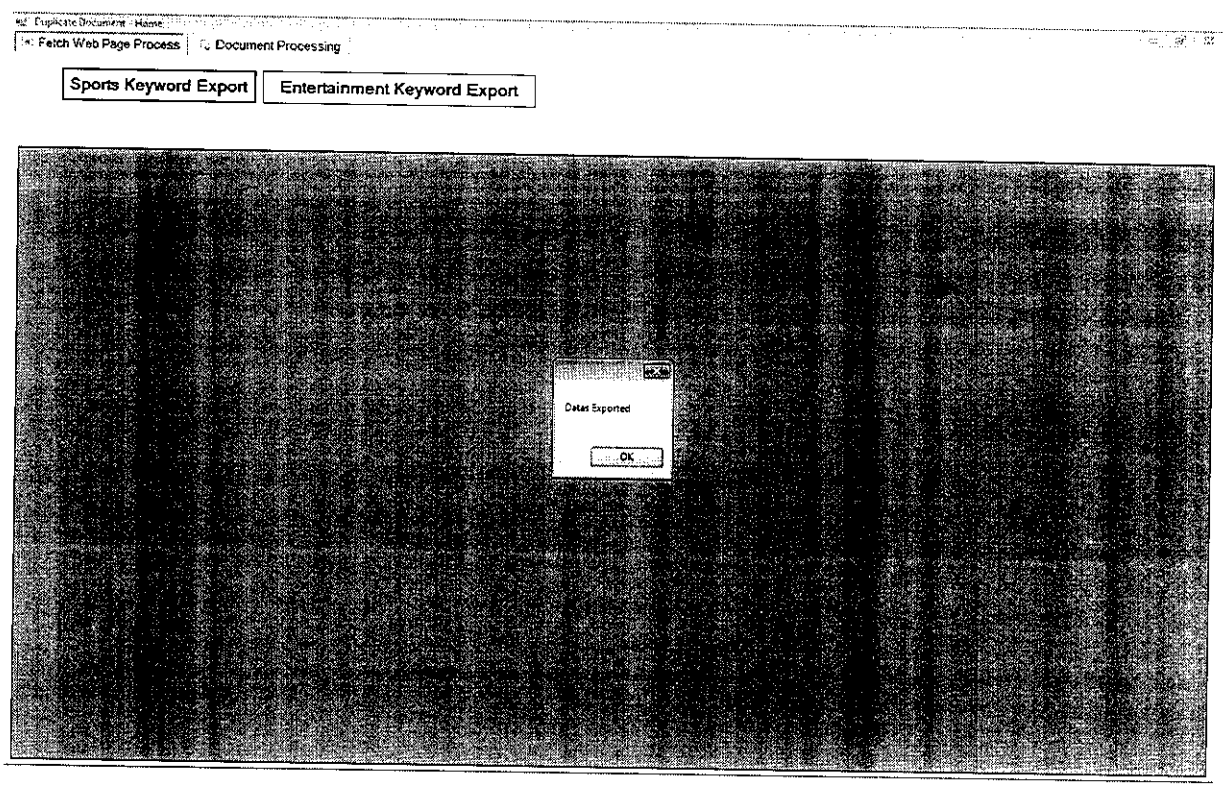
BUILDING URL CACHE



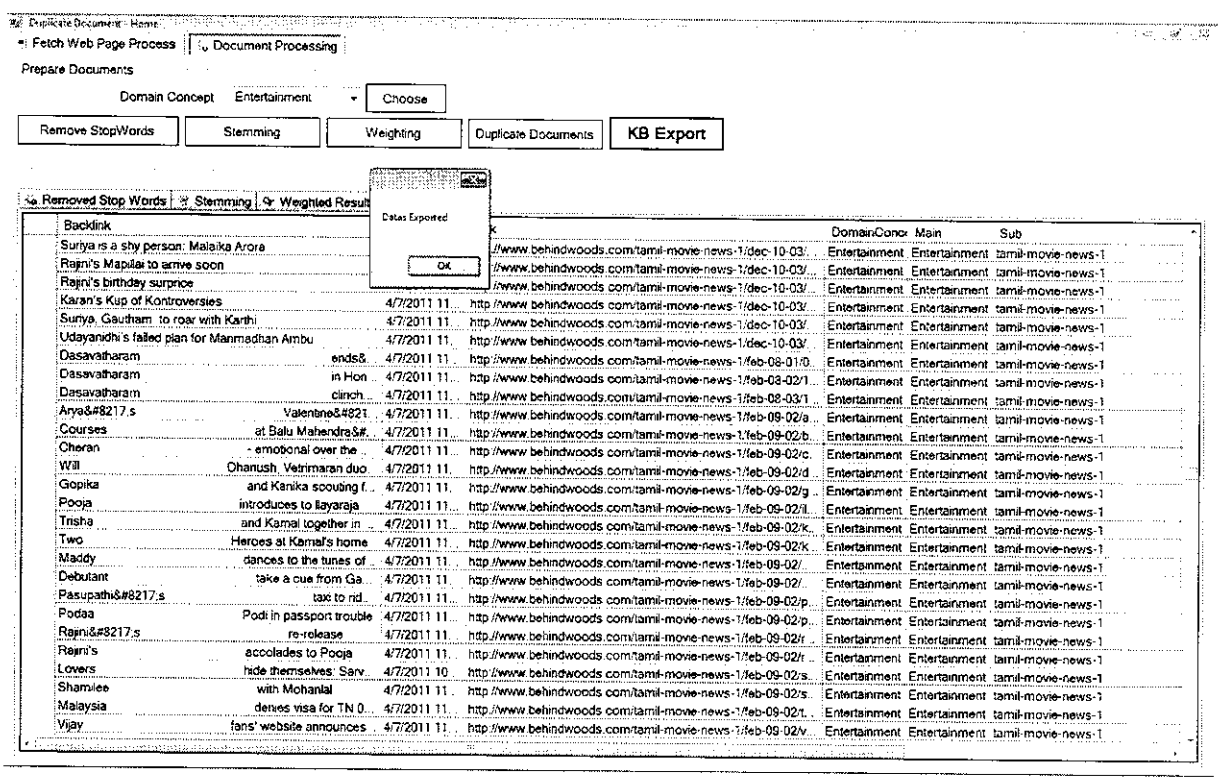
EXPORTER HOME



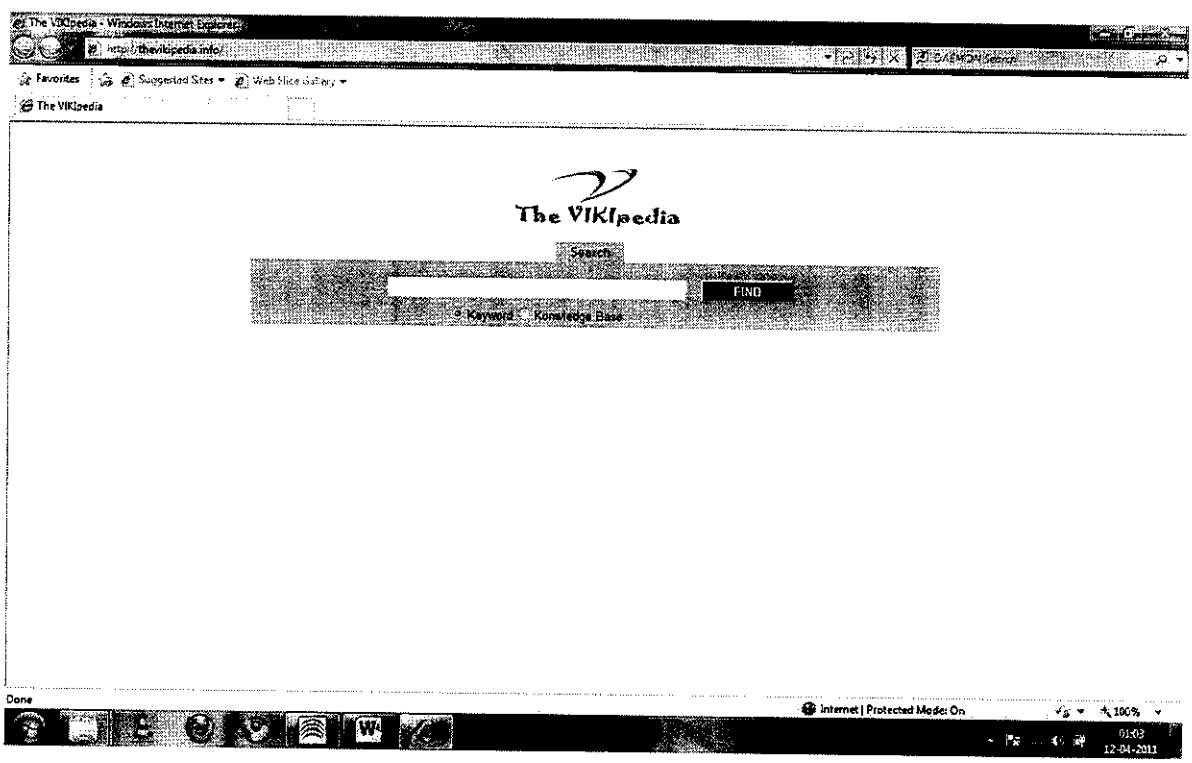
EXPORTING KEYWORD DATAS



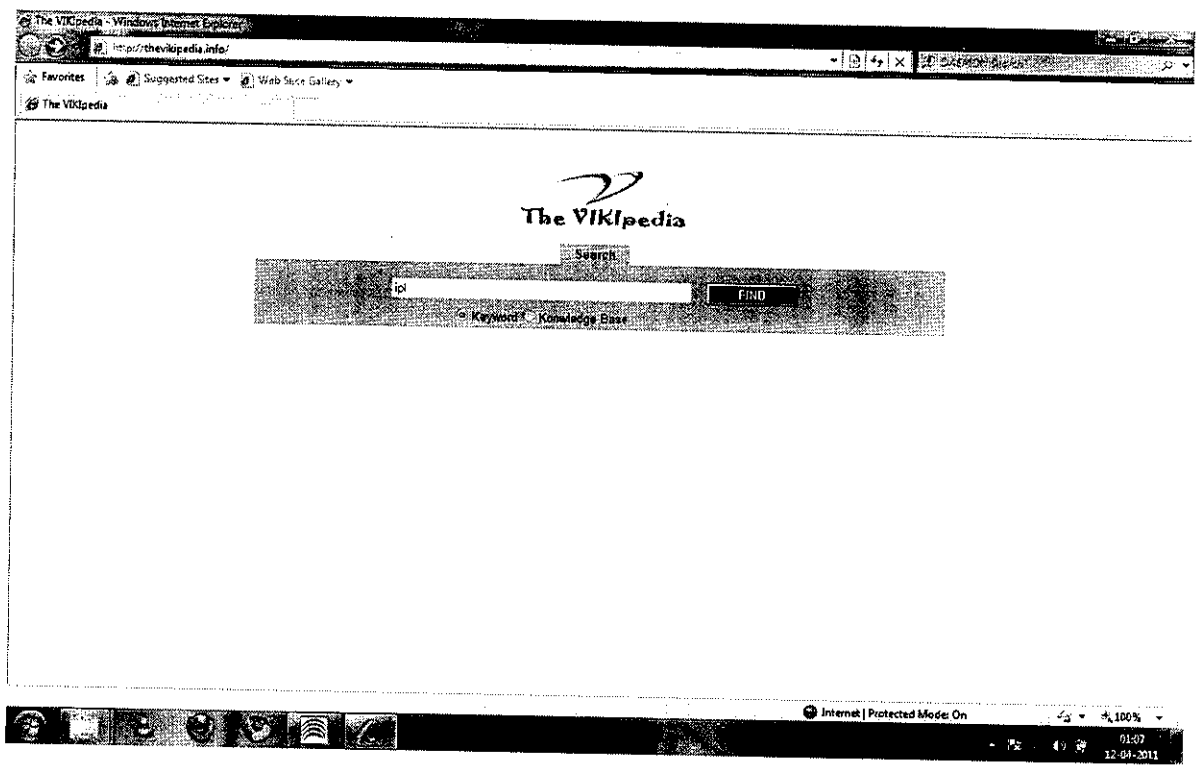
KNOWLEDGE BASE DATA EXPORTING



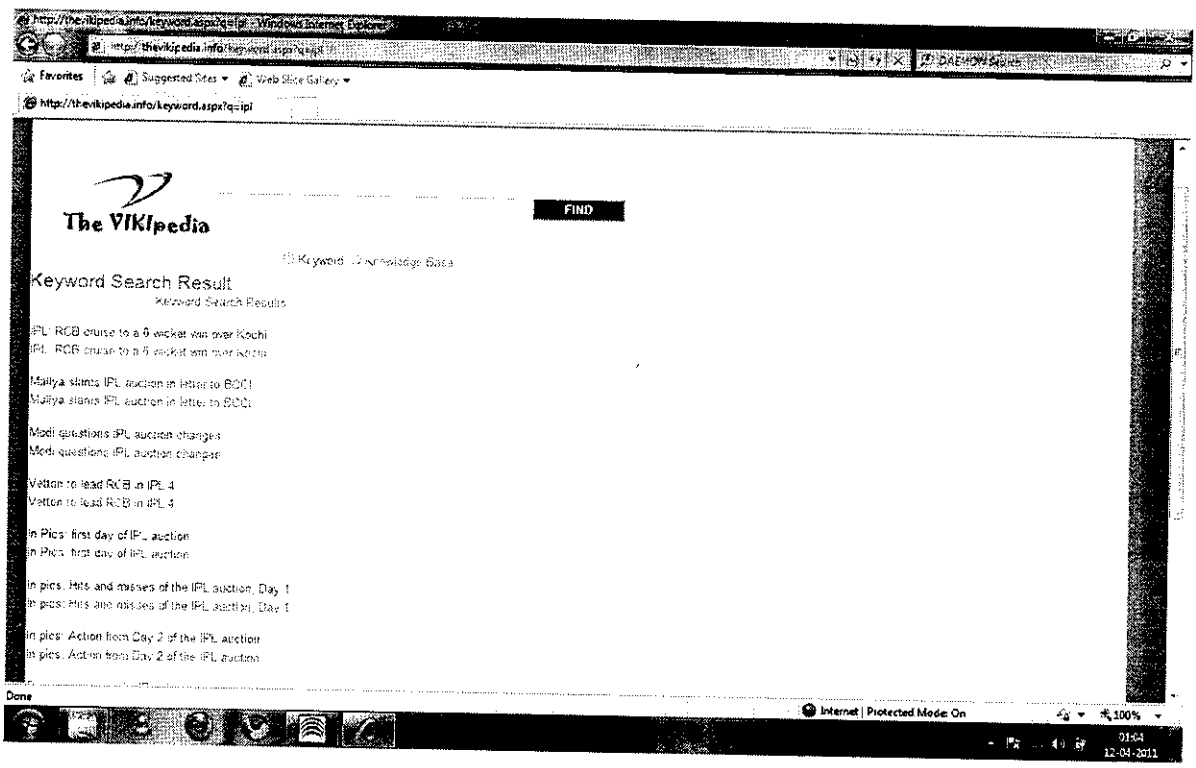
SEARCH HOME


Favorites
Suggested Sites
Web Slice Gallery
The VIKIpedia
The VIKIpedia
Search
FIND
Keyword
Done
Internet | Protected Mode: On
100%

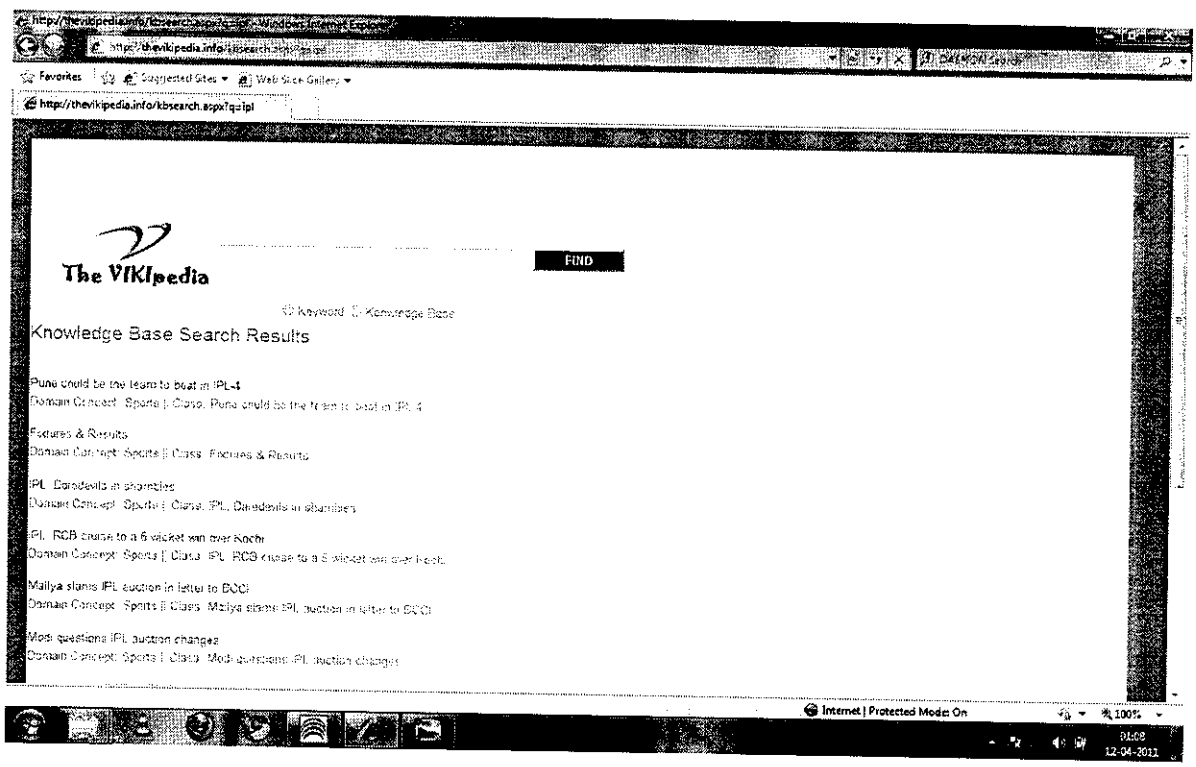
SEARCH QUERYING



KEYWORD RESULT



KNOWLEDGE BASE RESULT



REFERENCES

[1] T.Berners-Lee, J.Hendler, and O.Lassila, "The Semantic Web", Scientific Am., vol.284, no.5, pp.34-43, 2001.

[2] D.Beckett, "RDF/XML Syntax Specification (Revised)", http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/, 1994.

[3] S.Bechhofer, F.Van Harmelen, J.Hendler, I.Horrocks, D.L. McGuinness, P.F.Patel-Schneider, and L.A.Stein, "OWL Web Ontology Language References", http://www.w3.org/TR/2004/REC-owl-ref-20040210/, 2004.

[4] A.Go'mez-Pe'rez and O.Corcho, "Ontology Languages for the Semantic Web". IEEE Intelligent Systems, vol.17 , no.17, pp. 54-60, Jan.-Feb.2002.

[5] The Google.com search engine, http://www.google.com/, 2011.

[6] The AltaVista.com search engine for GUI, http://www.altavista.com/, 2011.

[7] Web hosting by BigDaddy.com , http://www.bigdaddy.com/, 2011.