The Study on Lucene Based IETM Information Retrieval

Jiaju Wu, Zhenji Liu, Xinglin Zhu and Rong Yu
Institute of Computer Application China Academy of Engineering Physics, Mianyang Sichuan, 621999 P.R. China

***Abstract*—With the intensive and large scale application of IETM in equipment integrated support, information retrieval technology becomes one of the most key technologies. This article discusses the full-text search technology and Lucene full-text retrieval engine, and combines them to develop a high-performance scalable IETM full-text retrieval system, this system can effectively deal with IETM unstructured data and structured data, significantly improving search efficiency and it is convenient to extend and easy to maintain. First, it briefly introduces information retrieval theory, and discusses the information retrieval requirement in IETM. Then, it thoroughly surveys the Lucene full-text retrieval engine, including its framework, retrieval process and index mechanism. Next this paper designs Lucene based IETM information retrieval process. Finally, the realization of Lucene based IETM information retrieval function.**

Keywords-IETM; lucene; information retrival; full-text retrieval

I. INTRODUCTION

Interactive electronic technical manual (IETM) appears in 1990s [1]. Due to its early performance ability, convenient query, updating and maintaining data sharing, real-time long-distance transmission, allowing multiple users to simultaneously read and many other advantages, IETM has become an important support of equipment integrated support [2], especially in weapon equipment support. IETM includes equipment's function, performance, run principium, program, operation, maintenance, detecting, inspecting, fault report and exception handles, disassembly program, assembly … etc all kinds of technique information, which has multifarious formats stored in database or magnetic disk folder. Information retrieval technology is the most IETM application in equipment maintain assistant. Traditional information retrieval technology commonly adopts keywords matching arithmetic to search useful information, which isn't impossible to deal with data effectively just like the search engines such as baidu and Google [3]. Moreover, keywords matching arithmetic can only search structured database data and helplessly deal unstructured data such as pdf, doc, xml, 2d/3d illustration, multimedia, picture…etc. The accuracy and high efficiency of IETM information retrieval is the key point for it sustaining equipment integrated support work. Most of the existing retrieval system requires the user with a complex query language to retrieve information. End-users not only need to be familiar with the syntax of the query language, but also need a comprehensive understanding of the diversity of IETM information document structure.

Full-text retrieval technology came into being in 80 years foreignness. And that is begun to research in 90 years inland [4]. Full-text retrieval technology [5] deals with character or text, audio frequency, video frequency, picture, various format documents…etc as data object to retrieve knowledge content, which is a full-text oriented new search technology, not a surface characteristic retrieval technology. Simple full-text retrieval can be used to character string match. Advanced full-text retrieval technology can used to structure data and unstructured data's knowledge search as IETM. Although there have much successful information retrieval engine tools such as baidu, Google…etc, IETM can not use these mature tools for its independent whole system application and equipment security problem. So we should design fit IETM terminal user's full-text retrieval system to satisfy IETM users' information retrieval requirement. Lucene is a mature Java open source full-text engine toolkit, which provides quadric development interface. It has realized query engine, index engine and text analysis engine in common use. This paper discusses the information retrieval requirement in IETM and surveys the Lucene full-text retrieval engine's structure, retrieval flow and index mechanism. Lucene based IETM retrieval course is designed. Lucene based IETM information retrieval function is realized.

II. IETM INFORMATION RETRIEVAL REQUIREMENT

IETM is a technical publication of content, which stocks in digital form with the form such as writing, graph, form, audio frequency, video, two/three-dimensional models and animation…etc, and offers equipment basic principle, operating use, maintenance, fault disposal, fight use and train with man-machine interactive mode [6]. IETM system has solved the problem of the handbook of paper mold, that is portable to carry , does not be easy to lose and can seek the needed information through the retrieval function of full text. According to Europe international specification for technical publications S1000D and Chinese specification GJB6600, IETM adopts the thought of modular design and unitary data source. IETM decomposes technical information content to DM (Data Module, DM). DM is the independence data unit of equipment's and its parts' data such as describe, program, operation …etc. It labels technical resource data according to special dada schema and generates SGML/XML document, which is stored in database and identified by DMC (Data Module Code, DMC). All data is stored without redundancy. It can be retrieved pass through code and information type. According to content structure, specification classifies DM into descriptive information, procedural information, fault

information, maintenance planning information, crew/operator information, Parts information, Battle damage assessment and repair information, wiring data, process data, technical information repository, container data, learning data, maintenance checklists and inspections…etc. DM includes IDStatus (Identification and Status section, IDStatus) and content section. IDStatus is used to document type's management, applicability management, retrieval and query management. Content is the main body of IETM DM, which is stored in XML document. DM content has cited figure, vector graph, audio frequency document, video frequency document, two/three-dimensional model, flash…etc except XML text. IETM information retrieval including equipment system, subsystem or sub-subsystem hierarchy catalog's technical information retrieval, keywords match search, user-defined logistic retrieval, content full-text retrieval, context-sensitive information search, applicability information retrieval… etc. IETM content full-text retrieval includes XML text retrieval and DM's citation documents.

III. LUCENE FULL-TEXT RETRIEVAL ENGINE

A. *Lucene's Framework*

Lucene is an excellent, mature, open source and free full-text retrieval engine tool, compiled with Java language compiled. It provides abundant API to interact with index information and be embedded in various application systems, for realizing application full-text index and retrieval. Lucene uses plentiful object oriented ideas. It has defined an index file form that is unconcerned with platform and designed abstract classes through abstracting system hard core t. Also, its platform realization part is designed to abstract classes' realization. Besides with these, the part related to specific platform is encapsulated to classes. So it forms a low couple, high efficient and easily twice development retrieval engine system.

Lucene retrieval engine includes four parts which are infrastructure component encapsulation, index center, external interface and query & analysis. Lucene framework [7] is like Figure I. Index center is Lucene's focal point, which includes index management and the data storage. Org. apache. lucene. index jar wrap realizes index's establishment, update, deleting operation [8], which improves retrieval efficiency through establishment participle index and searching index. Org. apache. lucene. store jar wrap realizes index file storage management. Infrastructure component encapsulation is foundation of Lucene, which includes document storage management and common classes. Org. apache. lucene. document jar wrap realizes document and domain information's storage management. Org. apache. lucene. util jar wrap is a common class which realizes some optimized data structure and arithmetic. External interface includes search management and Language analyzer. Org. apache. lucene. analysis jar wrap is language analyzer, which is used to cut participle mainly. Input text is divided into single word that is capable of being handled by index module. Org. apache. lucene. search jar wrap is search manager, which has provided user retrieval interface and realized query according to users' input condition. Org. apache. lucene. Queryparser jar wrap is a query analyzer, which is used to parse users' query sentence, analyze it and return a query object. Query analyzer can define user-defined rule also , realizing query conditions "and", "or", "not" like Google complex query.

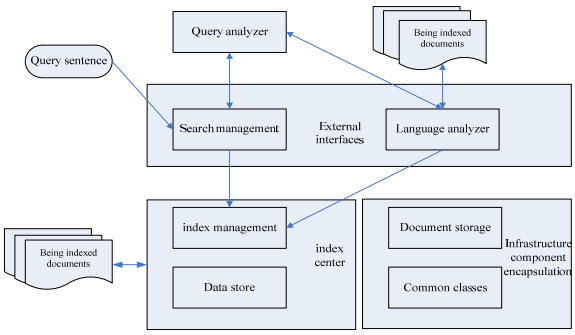

FIGURE I. LUCENE FRAMEWORK

B. *Lucene Information RetrievalFlow*

Lucene-based full text search contains index management and search index. Its data handling process is like Figure II.

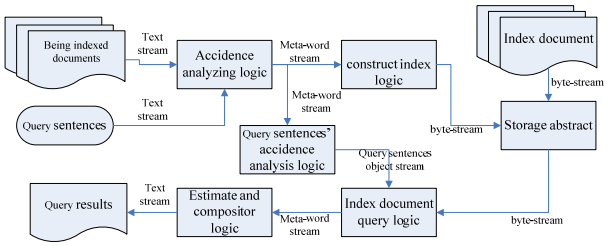

FIGURE II. LUCENE DATA HANDLING FLOW

When founding index, Lucene transfers being indexed content (the data to be searched) as document information to index writer object, firstly. And it names language analyzer. Then language analyzer handles document participle voluntarily to form etymon (token), through dividing it into individual words, which has wiped off punctuation and insignificant words like "a", "the", "or"…etc (These are used frequency as article, preposition, adverb or conjunction in English). Etymon (token) is transmitted to language processor to be handling, which forms "key word". For English meta-words' language handling, it changes word to lowercase. Then it cuts it short to etyma (such as "apples" conversion into "apple") or changes it into etyma form (such as "gave" conversion into "give"). Finally, index component forms dictionary according to words being produced by language processor. And it adapts reverse order index to form definite data structure index document.

Lucene inquire about information according to index. Firstly, it transmits user input query information to query parser object, that object adopts similar participle method and language processing technique of establishment index to get the query keyword and word. Then it carries analysis through parsing to establish syntax tree and form query objects. Finally, index searcher objects open and read index document in index catalog to find out word &document link table in reverse index table. It finds the result from "and", "differ", "or"… etc operations according to syntax tree.

C. *Lucene Index Mechanism*

Index is the foundation of Lucene to carry out the full-text search. It concerns the retrieval efficiency. In Lucene, analyzer can only carry out text data analysis, and produce index. For other type's data, we should submit them analyzer to handle after changing them into text data. Lucene manages index through index documents, which stores sequential information and reverse information all.

Lucene's index structure is a hierarchy structure which includes segment, document, field and term. The four are containing relation. Index is divided into some segments. Segment is divided into some documents. And document includes some fields. Fields includes some key words. Sequential information is core of search document that adopts sequential row index table, like Figure III.

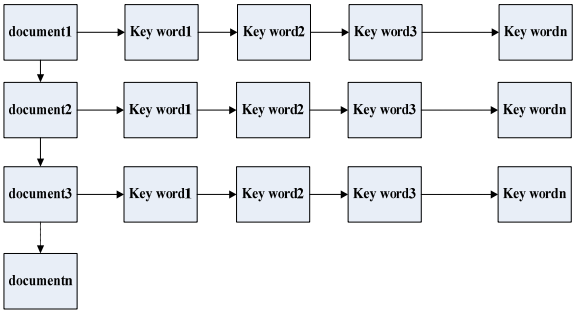

FIGURE III. SEQUENTIAL ROW INDEX TABLE

Reverse information is the core of Lucene index file, which records the reverse row index table. Reverse row index comes from searching records according to attribute value in practice application. Each item in index table includes the property value and the address of property value record. In Lucene, item of index table is key word, whose attributes include document no, frequency and address. Reverse row index table is like Figure IV.

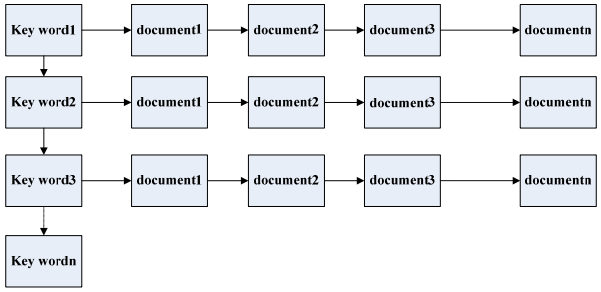

FIGURE IV. REVERSE ROW INDEX TABLE

IV. LUCENE-BASED IETM INFORMATION RETRIEVAL

IETM includes disk document data and relation database data. IETM information retrieval is divided into two courses approximately: index course (Indexing) and search course (Search), like following picture shows.

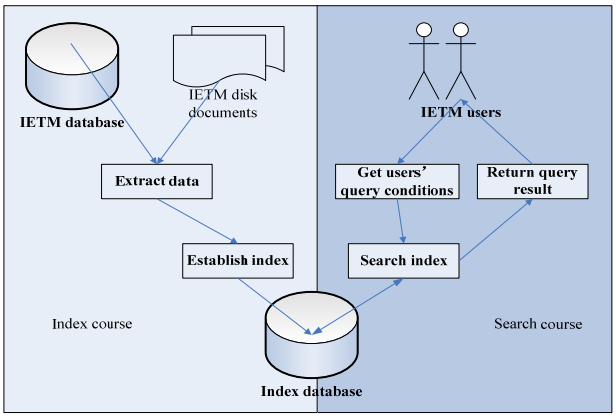

FIGURE V. IETM INFORMATION RETRIVAL COURSE

IETM index course is extracting information from IETM DM data, PM (Publication Module, PM) data, TIR (Technical Information Resource, TIR)data, and unstructured entity documents, such as picture, vector graph, audio frequency document, video frequency document, two/three-dimensional model, flash…etc and establish index. Index course's detail approaches is described at following.

(1) When we upload and parse IETM data package to write into database and disk folder, IETM data package is also transferred to language analyzer for voluntary participle. Some individual words are gotten. Analyzer has wiped off punctuation and insignificant words to form etymon.

(2) Etymon is transmitted to language processor to handle it into "IETM key word".

(3) IETM key words are transmitted to index module to establish IETM dictionary. Index module also queues up dictionary according to alphabetically order and merges identical word to form document reverse index table.

IETM search course is to get user query request, search established index and return query result. Its detail approaches are like following.

(1) IETM user inputs his query conditions to query parser object.

(2) Query parser object analyses IETM query conditions' accidence, grammar for carrying language processing through index. Accidence analyzer distinguishes keywords from query sentence mainly. Grammar analyzer form grammar tree according to syntax rule. Language processor is like index course.

(3) Search module uses IETM grammar tree to search index and get documents accord with grammar tree.

(4) Search module queues up order for result according to the correlation of document and query sentence. Then it returns the result to users. IETM software highlights these results.

V. LUCENE-BASED IETM INFORMATION RETRIEVAL REALIZATION

IETM index uses analyzer class object to get participles. And cn.smart.SmartChineseAnalyzer class object is used to get Chinese participles. Document class object realizes IETM data source identifier. Field class object describes the attributes of IETM data source. IETM index realization code is like following.

```
...
import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.cn.smart.SmartChineseAnalyzer;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;
import org.apache.lucene.document.StringField;
import org.apache.lucene.document.TextField;
import org.apache.lucene.index.DirectoryReader;
import org.apache.lucene.index.IndexReader;
import org.apache.lucene.index.IndexWriter;
import org.apache.lucene.index.IndexWriterConfig;
...

public static Analyzer analyzer = new SmartChineseAnalyzer(Version.LUCENE_40);
indexPath = initLuceneIndexPath();
Directory dir = FSDirectory.open(indexPath);
IndexWriterConfig iwc = new IndexWriterConfig(Version.LUCENE_40,analyzer);
iwc.setOpenMode(IndexWriterConfig.OpenMode.CREATE);
IndexWriter writer = new IndexWriter(dir, iwc);
doc = new Document();
DataModule dm = dmcIndexManager.parsFileIndexInfo(file);
doc.add(new StringField("dmc", dm.getDmc(), Field.Store.YES));
doc.add(new StringField("dmType", dm.getDmType(), Field.Store.YES));
doc.add(new StringField("icType", dm.getDmcIc(), Field.Store.YES));
doc.add(new StringField("infoName", dm.getInfoName(), Field.Store.YES));
doc.add(new StringField("techName", dm.getTechName(), Field.Store.YES));
doc.add(new TextField("contents", new BufferedReader(new InputStreamReader(fis,
"UTF-8"))));
writer.addDocument(doc);
writer.close();
```

FIGURE VI. IETM INDEX REALIZATION

IETM search adopts IndexSearcher and Query class object mainly. Query class object encapsulates inquire word, field and language analyzer, offer the rich inquiry such as vague inquiry, semantic inquiry, phrase inquiry, scope inquiry and combination inquiry way. It delivers IndexSearcher to query and return result into Hits class. When there have more retrieve results, IndexSearcher reserves the former 100 records to match with highest degree acquiescently.

```
...
import org.apache.lucene.index.DirectoryReader;
import org.apache.lucene.index.IndexReader;
import org.apache.lucene.search.IndexSearcher;
import org.apache.lucene.search.Query;
import org.apache.lucene.search.ScoreDoc;
import org.apache.lucene.search.TopDocs;
import org.apache.lucene.store.Directory;
import org.apache.lucene.store.FSDirectory;
import org.apache.lucene.util.Version;
...
indexPath = initLuceneIndexPath();
                IndexReader reader =
DirectoryReader.open(FSDirectory.open(indexPath));
        IndexSearcher searcher = new IndexSearcher(reader);
                System.out.println("Searching for: " + query.toString());
                TopDocs results = searcher.search(query, 100);
                int numTotalHits = results.totalHits;
                System.out.println(numTotalHits + " total matching documents");
```

```
ScoreDoc[] hits = results.scoreDocs;
        List<DataModule> dms = new ArrayList<DataModule>();
        for (int i = 0; i < hits.length; i++) {
                Document doc = searcher.doc(hits[i].doc);
                DataModule dm = new DataModule();
                dm.setDmc(doc.get("dmc"));
                dm.setDmType(doc.get("dmType"));
                dm.setDmcIc(doc.get("icType"));
                dm.setInfoName(doc.get("infoName"));
                dm.setTechName(doc.get("techName"));
                dms.add(dm);
        }
        reader.close();
        return dms;
```

FIGURE VII. IETM SEARCH REALIZATION

VI. CONCLUSION

Lucene based IETM information retrieval has efficient, succinct, easy used characteristic. Its retrieval speed is rapid. Its accuracy is high. IETM information retrieval result is to be able to queue up order according to related degree, which can raise apparently the knowledge guidance ability that IETM offers technical information. But for time limited, Chinese participle arithmetic's semantic veracity and precision need to be improved more. And retrieval result compositor arithmetic has some limitations also. Our future work will focus on arithmetic optimization. Then we will integrate improved arithmetic into IETM system.

REFERENCES

[1] ZHU Xindong. Weapon Equipment IETM[M]. National Defense Industry Press, 2009.

[2] XU Zongchang, Huang Yijia, YANG Hongwei. Supportability Engineering and Management for Equipment[M]. National Defense Industry Press, 2010.

[3] Wang Zhenfeng. the Research and applicaition of full-text retrieval technology based on Lucene[R], DongHua University Master Degree Paper, 2015.

[4] Wu yong. Research of Problems in WWW-based Chinese Serch Engine, Nanking University Master Degree Paper, 2000.

[5] Liu Tianyuan , Song Meina , Zhang Xiaoqi. Research of massive heterogeneous data integration based on Lucene and XQuery . In Web Society (SWS), 2010 IEEE 2nd 5(1) : 648~652.

[6] ATA , ASD, AIA. International Specification for Technical Publications S1000D [EB/ OL] . (available at http :// www.s1000d.org, 2012).

[7] Zhang Wubo, Shi Luhua, Li GuiRong. Research on Full-text Search Engine Lucene System Model and Application [J], Software Guides, June 2015, vol.14,No.6 127-129

[8] He Wei. Lucene-based Full-text Search Engine Research Design and Realization [J], Intelligence Journal, 2015 (9), vol.14,No.6 p88-90