

9 エージェントによる生産支援技術の開発（第4報）

村河亮利, 藤原義也, 岩谷 稔, 相原玲二^{*1}, 甲斐雅貴^{*2}, 西田俊彦^{*3}

Development for manufacturing support technology with agent (4th Report)

MURAKAWA Akitoshi, FUJIWARA Yoshinari, IWATANI Minoru, AIBARA Reiji^{*1},
KAI Masaki^{*2} and NISHIDA Toshihiko^{*3}

We have developed a system that collects information from the Internet automatically, once users set conditions what they want. Since this system uses RDBMS (Relational Database Management System), search time is shortened compared to DB using text file, and new functions that are difficult by using text file are realized.

New functions we add and improve are as follows.

1. Using collected information of DB shortens the search time.
2. This system can analyze and compare complicated units.
3. By assigning a priority to items, users can sort results.
4. We apply multithreading on the server program in order to process requests simultaneously.

In addition, this system can connect many RDBMSs (e.g. PostgreSQL, Oracle, DB2, MySQL) via JDBC.

キーワード：エージェント，ネットワーク，インテリジェント，情報収集

1 緒 言

製品開発時に検索エンジンを利用した情報収集は主に手動で行われている。そのため利用者は時間と労力を要しているのが現状である。そこで、利用者が必要とする情報の条件設定を行えばインターネット上から自動的に情報収集・提示することにより、利用者の負担を軽減するシステム開発を行ってきた。本年度は、収集したデータを RDBMS (Relational Database Management System) に保存し再利用により検索時間の短縮を行った。また、単位及び結果表示方法等の機能改善を行った。

2 システム

昨年度のシステムから Web サーバの代わりに利用者に専用のソフトを用意した。システム概要及びシステムの構成について述べる。

2.1 システム概要

現在、分散 DB (データベース) から目的とするデータを収集するシステム^{1,2)}や検索エンジン³⁾は開発されている。しかし、時々刻々と更新されるインターネット上のデータを収集し、利用者が望む情報を整理し提示するシステムは見当らない。そこで、昨年度作成したデータ収集システム⁴⁾を使用し、インターネット上で広く使用されている HTML 中から目的のデー

タを抽出・加工し、利用者に提示する手法について検討する。具体的には、まず利用者が検索エンジンを用いてキーワードを入力し検索結果アドレスを対象に検索を行うか、検索対象の URL の指定を行う。利用者はキーワード、要求項目等を入力する。入力データは検索サーバに送信され、インターネットや独自 DB から情報収集を行う。収集した情報から、利用者の要求項目とデータが一致するものを選択し利用者に提示する。

2.2 システム構成

システム構成を図 1 に示す。昨年度のシステムには Web サーバを使用していた。Web サーバの使用により、システム導入後は各端末のブラウザにて直ちに本システムの利用が可能である。しかし導入時に Web サーバの設定作業を行わなければならない。そこで、この作業負担を軽減するためにクライアント部分を Java 言語で作成した。この場合利用者の各端末に

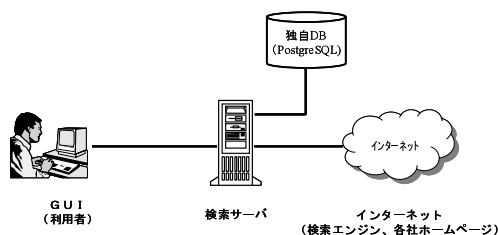


図 1 システム構成

^{*1} 広島大学, ^{*2}(有)広島情報通信研究所, ^{*3}(有)インターコミュニケーションズ

JVM (Java Virtual Machine) をインストールする必要が出てくるが、Web サーバを設定するよりも作業負担が少ない。また、Java 言語で作成しているため、ブラウザと同様マルチプラットフォームに対応するという特徴は共通である。

3 検索機能

昨年度の検索システムに機能追加及び機能の改善を行った。RDBMS を使用することにより検索時間の短縮が図られた。また複雑な単位の組合せ (例: km/h, kg・m/s²) に対応するよう機能改善を行った。インターフェースについても利用しやすいよう GUI 及び結果表示方法の改善を行った。

3.1 RDBMS によるデータ管理

昨年度までは検索対象となる URL 情報や Web からダウンロードした HTML ファイルなどを全てテキストファイルに保存してデータ管理を行っていた。しかし、この場合検索条件や検索 ID (検索開始時に発行される ID) を利用者が全て管理する必要がある。また検索総数が多い場合、ファイルアクセスに時間がかかるため、検索終了までに非常に時間がかかるという欠点があった。

そこで、検索条件、リンク情報などの管理を RDBMS で行うことにより検索時間の短縮を図った。

さらに RDBMS の利用者認証機能を用いることで、利用者ごとに検索情報を DB で管理可能となるため、利用者は検索 ID や設定した検索条件などを管理する必要がなくなり、使い勝手の向上が図られた。

RDBMS を用いる場合の欠点として導入時及び設定変更時に DB の管理コストがかかる点があげられるが、利用者の追加・削除・DB 管理 (初期化、DB 初期設定)などをすべて GUI 画面で操作可能とし、DB 管理の省力化を図った。

使用した RDBMS はオープンソースである PostgreSQL で、Linux、Windows、MacOS X などさまざまなプラットフォームをサポートしている。

3.2 独自 DB 利用による検索効率の向上

Web から HTML ファイル取得時に URL と HTML を DB に保存し、再利用可能とした。これにより Web へのアクセスが減少するためさらに検索効率の向上が図られる。DB の利用方法は、『全て利用』、『利用しない』、『指定日以降を利用』から選択可能とした。また、HTML ファイル取得クラスをマルチスレッドで行うことでさらに検索効率の向上を図っている。

3.3 単位解析

単位解析方法を $12.3 * 10^2 \text{kg} * \text{m}/\text{sec}^2$ を例に手順を

表 1 単位を含む数字の格納構造

数字	0			
基本単位	g	m	sec	...
階乗	0	0	0	0

示す。

1. 各単位を含む数字に対して、表 1 の様な数字、基本単位に対応する階乗を表す配列を初期化する。
2. 数字 (12.3) を解析し表 2 の数字に登録する。
3. 演算文字である * の記号があるので、次の数字を解析し、102 を 2 で登録した数字に掛け、再び表 2 の数字に保存する。
4. 次の文字が数字及び演算文字ではないので、あらかじめ登録してある接頭語 (k) と基本単位 (g) の組合せを比較し、合致するものを得る。また合致した単位に対する階乗を解析する。
5. 接頭語 (k の場合 3) と階乗の数字 (1) を掛け合わせ、表 2 の数字に 10^3 を掛け合わせ再び表 2 の数字に保存する。
6. 基本単位 (g) の接頭語が k であるので、g 内の配列に 3 を加える。
7. 次の文字は掛け算の記号なので、分母を示す符号 (+ 1) にする。割り算の場合は分子を示す符号 (- 1) にする。
8. 同様に単位の塊が終了するまで 4 から 7 まで繰り返す。解析終了時の状態を表 2 に示す。

表 2 単位解析終了時の状態図

数字	1 230 000			
基本単位	g	m	sec	...
階乗	3	1	- 2	0

3.4 単位比較

$12.3 * 10^2 \text{kg} * \text{m}/\text{sec}^2$ と $5 \text{kg} * \text{m}/\text{sec}^2$ を比較する場合の例を示す。

1. 比較元 ($5 \text{kg} * \text{m}/\text{sec}^2$) の単位を 3.3 の手順と同様に解析する。
2. 比較元と比較先 ($12.3 * 10^2 \text{kg} * \text{m}/\text{sec}^2$) の基本単位の階乗パターンが一致するか調べる。
3. 階乗パターンが一致すれば、数値の大小比較を行いその結果を返す。今回の場合 1 2300 000 と 5 との比較する。もし階乗パターン異なれば、比較不可の結果を返す。

3.5 検索対象リンクの絞込み

一般に検索サイトを利用する際、利用者は検索結果に表示される URL とそのサイトの抜粋を見て、サイトを調べるかどうかを判断することが多い。そのた

め、この作業と同等の機能として、検索エンジンの検索結果から得られる URL 情報とサイトの抜粋を自動で抽出・利用者に提示し、利用者が調べたいリンク先を対話的に選択可能とした。この機能により、利用者が調査対象としていない URL を除外することができ、検索効率の向上を図れる。

またニュースサイトなどに代表されるように、ページから張られているリンク数は多いが、リンク先をさらに調べても有効な検索結果を抽出できる可能性は小さく、検索効率が低下する場合がある。そこでサイトに含まれるリンク先の数が多い場合、検索を保留して、そのページの検索を続けるかどうかを利用者が選択できる機能を加えた。

3.6 ユーザーインターフェースの改善

検索機能の追加に伴い本システムの入力画面（図2）に以下の機能を追加した。

（テキスト DB, RDBMS 共通）

- ①AND 検索, OR 検索の選択
- ②ページごとの登録 URL の最大値
- ③イベント発生時にメッセージの送信先メールアドレスとメールサーバの登録
- ④同義語の内容の変更・追加
（RDBMS のみ）
- ⑤検索 ID, 検索条件の表示
- ⑥利用者の追加・削除・DB 管理（初期化, DB 初期設定等）

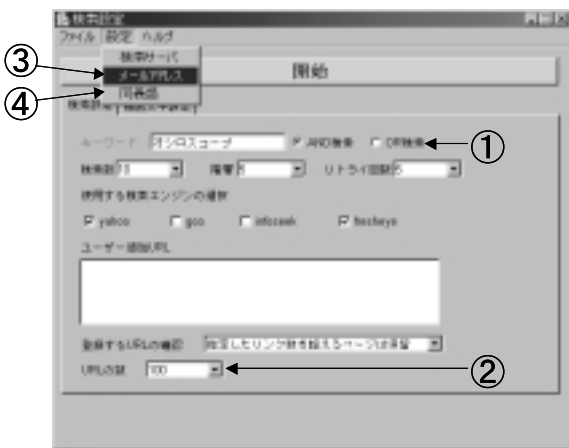


図2 入力画面（テキスト DB）

3.7 検索制御及び状況確認

検索開始, 検索状況照会, 検索結果表示, 検索停止及び検索再開の基本機能に加え, Web から収集済みのデータを再利用し, 項目データの抽出条件のみを変えて検索を行う機能を追加した。

3.8 検索サーバのマルチスレッド化

検索サーバプログラムをマルチスレッドで実行可能

にすることにより, 同時に複数の検索要求を処理可能とした。それぞれの検索要求に対して, 検索サーバが固有の ID を割り当て, 検索開始時に利用者に通知する仕様とした。また, 同義語についても各 ID 毎に設定可能とした。

4 結果及び考察

実行環境, 検索条件の設定及び実行結果を示す。

4.1 実行環境

実行環境を表3に, 使用した検索エンジンやその他の設定条件を表4に示す。

表3 実行環境

	検索サーバ	DB サーバ
CPU	Pentium IV 2.8GHz	Pentium III 866MHz
RAM	2 Gbyte	512Mbyte
OS	Windows2000	Red Hat Linux 9
アプリケーション	Java 開発言語	PostgreSQL 7.3.4
備考	WAN : 100Mbps	LAN : 100Mbps

表4 実行条件

使用した検索エンジン	yahoo, fresheye
検索件数	各検索エンジン結果の上位100件
リンク階層	5階層
再試行回数	5回

4.2 検索条件及び結果

4.2.1 検索結果

電子回路設計に必要な機器として, オシロスコープの仕様表の情報収集を例にとり, 本システムを利用した結果を表5に示す。

表5 検索結果

キーワード	オシロスコープ				
項目	帯域	100MHz 以上			
	チャンネル数	2 以上			
	価格	2,000,000以下			
時間	自動 : Web : 65分, 解析 : 6分 手動 : 2,310分 (推定)				
ページ数	総数	自動	手動	見落	
	9,131	565	50	-	
内訳	0階層	121	34	12	評価せず
	1階層	823	85	25	
	2階層	1,385	175	14	
	3階層	2,385	209	12	
	4階層	2,919	56	1	
	5階層	1,498	6	0	

0階層を目視で調べた結果表5に示すように、見落としが1個ある。このページの該当表をソースコードで確認すると、1行目とその他の行が分離した複数の表から成り立っており、今回のシステムでは想定していない形式であった。しかし、一般的にこのような記述は行わない例外である。また、1階層以降は数が多いため目視での評価を行わなかった。

正解率は階層が深くなるにつれて減少する傾向が見られる。これは検索エンジンの結果が目的のページである確率が高いからと考えられる。また検索対象のリンクを絞り込まなかったことも正解率が低下した原因と考えられる。そのため対話的に検索ページの絞り込みを行えば、より正解率が上がり、かつ検索時間も短縮されると予想される。

手動時間の推定は、0階層の121件を実際に検索した総ページ数9,131ページから推定した。

よって従来の手動検索に比べ、本システムの有効性があるといえる。

4.2.2 検索時間の比較

表4の条件でのホームページを検索した場合の検索時間を表6に示す。

表6 テキストDB, RDBMS 利用時の検索時間

DBの種類	独自DB	検索時間(分)
テキストDB	-	131
RDBMS	利用しない	71
	利用する	43

表6の結果から、実験結果は検索を行う時間帯や検索条件により変わる可能性はあるが、RDBMSを利用することで検索時間を約46%短縮することができた。さらに独自DBを利用すればテキストDBに比べて約67%検索時間の短縮ができた。

4.3 結果表示

結果表示画面を図3に示す。表示画面には抽出内容に加えて、抽出元のURL、ページの取得日時と更新日時を表示した。従来は検索して抽出順に表示をしたため、利用者は要求する内容を検索結果全体から探す必要があった。そこで利用者が表示の順序を選択できるように要求項目に優先順位を付けることで、優先度の高い順に検索結果の並べ替えを可能とした。また、検索結果の取得元を確認するため簡易ブラウザ機能を追加し、URLをクリックすることでリンク先を表示可能とした。



図3 結果表示画面

5 結 言

昨年度までテキストDBを用いていたが、RDBMSを用いることにより時間短縮及びテキストDBでは実現困難であった機能の追加が可能となった。

本年度に機能追加及び改良した機能は以下の4点である。

1. インターネットからの収集済みDBを用いることにより、検索時間の短縮を図った。
2. 複雑な単位の解析及び比較が可能となった。
3. 要求項目に優先順位を付けることで、優先度の高い順に検索結果の並べ替えを可能とした。
4. 同時に複数の検索要求を処理可能とするため、マルチスレッドに対応した。

また、検索サーバからRDBMS操作にJavaのクラスであるJDBCを使用しているため、PostgreSQL以外のRDBMS(Oracle, DB2, MySQLなど)へも対応可能である。

今後、本システムは電子回路以外の分野にも応用可能であるため、適用例を増やす予定である。

文 献

- 1) 西野, 赤坂, 小泉: 情報処理学会論文誌40(1999)1, 253.
- 2) 西野, 木室, 赤坂: 情報処理学会研究報告98(1998)8, 201.
- 3) 森川, 深澤, 吉岡, 大須賀, 本位田: 情報処理学会講演論文集3(2000), 367.
- 4) 村河, 藤原, 岩谷, 田尾, 吉野: 広島県立西部工業研究報告, No46(2003).