マルチメディア通信と分散処理 78－5
（1996．9．27）

アーカイブ検索システムDartsの実装と評価

広瀬 雄二
大駒 誠一
慶應義塾大学理工学研究科管理工学専攻

概略

Internet 技術の進歩により、多くの有用なソフトウェア、データなどのパッケージがいながらにして瞬時に入手できるようになった。あまりに多いパッケージの中から欲しいものを探すためのユーティリティとして現在 archie がある。しかし、archie は検索キーとしてパッケージのファイル名しか指定できないため、探しているもののファイル名を知らない場合や、漠然とこういったものが欲しいと言うような場合には無力である。そこで本稿では、ファイル名だけではなく、カテゴリ、OS、マシン、用途などパッケージの持っている属性を検索キーとして指定できるコンテンツベースの検索システムを提案する。

Construction and Evaluation of Archive Searching System Darts

HIROSE Yuuji and OKOMA Seiichi
Department of Administration and Engineering,
Faculty of Science and Technology, Keio University

Abstract

The growth of the Internet technology has enabled us to obtain various software packages from the network. We currently use *archie* to search the enormous number of packages for the proper one. However *archie* allows only partial or entire file name of the packages for the searching key. Therefore we cannot search any package with *archie* when we have no information about file name or we are looking for some package without the firm conscious of what we actually need. In this paper, we propose the contents-based package retriever which allows attributes and characteristics such as category, OS, machine, and their use, for searching keys.

1 背景

通信技術の発達、とくにインターネットの普及により我々はおびただしい量の情報を簡単に入手できるようになったばかりでなく、情報の発信者となることも容易になった。とくにHTTP(Hyper Text Transfer Protocol) を利用したWWW(World Wide Web) は、その構築のコストの低さと視覚的効果の高さにより瞬く間に世界中に広まった。これにともないWWW により提供される情報を検索するための技術も広く研究の対象にされ、実用的なシステムが林立する今日の状況を迎えた。

しかし、目を転じて、インターネット上で流通している資源の一つ「ソフトウェア」の検索性はどうかというと、1992 年頃から普及したArchie 以来ほとんど進歩を見ていない。

一般的に、ソフトウェア、データ、あるいはドキュメントなどのパッケージは、`tar+gzip` や `zip` などのツール用いてアーカイブと呼ばれる一塊のファイルに納められ anonymous ftp 上で公開され、配布される。

archie は、そうしたアーカイブのファイル名リストを、一ヶ所集中の archie server に登録しておき、archie 専用検索クライアント (archie コマンドなど) にファ

イル名をキーとして与えることにより、目的アーカイブの位置情報を解として得るシステムである。

archie server は、各 anonymous ftp サービスホストのファイル名リストを定期的に回収し自己のデータベースに登録する。どのホストのリストを集めるかは各 archie server により決まっているので、当然のことながら archie server に登録していない anonymous ftp サービスホストにしか存在しないファイルは検索することができない。

また、検索時に与えるキーの性質を比較した場合、一般の、文書/文献検索では、探したい文書には探したいことがらに関することが含まれていることが明らかなので、

探すためのキーワード $\in$ 知りたいこと

という概念上の関連が約束できる。ところが、アーカイブ検索においてはこのことは全く保証されない。なぜなら、

- 検索のためのキーがファイル名に限られている
- ファイル名がアーカイブの中身の性質を表しているとは限らない
- ファイル名長等 (8+3 文字等) の制限があり、意味を持たせることが困難である

のような制約から、検索者がどのようなキーワードを選択すれば良いかを決定するのが非常に困難となっているからである。ところが実際、検索のためのキーワードを容易に選定できないこの状況で、いまだに archie が有効に利用されているのは、NetNews の力に負うところが大きい (後述)。

2 パッケージの検索

前節で述べたように、通常テキストの検索と、アーカイブ検索では検索者の発すべき問い合わせを導出する過程に差異がある。ここでは、パッケージ検索者がなんらかのパッケージを必要とする状況に至る経緯を四つの層に分類し、各層に対して異なる解発見アプローチを取るべき必要があることについて述べる。

2.1 検索者の四層分類

特定のパッケージがないかどうか探す時に検索者が、パッケージを要求する動機のレベルを分類すると以下のようになる。

1. ある問題を解決したい。
2. ある機能を提供するパッケージが欲しい。
3. 名前の分からないパッケージが欲しい。
4. 名前の分かるパッケージが欲しい。

レベル 1は、現在計算機上で対象としている問題を解決するうえにおいて、不便であるとか、トラブルがあるだとかの問題をかかえてはいるもののその具体的解決策は知らない、または思い付かない場合を指す。

通常このレベルの動機を持っている検索者は、手がかりが得られないので、曖昧な質問をするしかなく、回答を人間に求めて、問い合わせを身近にいる詳しい人、あるいは NetNews などに発する。

レベル 2は、パッケージ自体に求める機能ははっきりしているが、そのような機能を有するパッケージが存在するかどうかは定かでない状態である。例えば、あるユーザが Sというソフトウェアを O_1という OS 上で有効に利用している場合に、別の OS で Sの移植版、もしくは同様の機能を持ったソフトウェアを利用したいと思ったが、実際にそれが存在するかどうかが分からない、といった場合などがこれにあたる。

ただし、どのような機能を必要とするか自体の発想が誤っている可能性もあり、たとえ検索が成功して何らかの解が得られたとしてもそれが本当に当該問題を解決するために有効なパッケージを指しているどうかかは保証されない。

レベル 3は、パッケージ自体に求める機能、およびそれを有するパッケージが存在することが分かっていて、その名前のみを失念している状態である。

レベル 4は上の状態に加え、パッケージの名前、またはその一部があらかじめ分かっている場合で、既にそのパッケージの古いバージョンなどを持っているような場合に相当する。

2.1.1 各レベルの検索に必要なもの

上述した各レベルの検索者に解を与えるためには以下に示す通り、全く別の検索機構が必要である。

レベル 1 レベル 1の検索者は、発生した問題を解決してくれるようなパッケージを求める。このような場合同様な問題を解決した人からの回答を集めた症例データベースが必要である。

レベル2 レベル2の検索者は、パッケージの持っている機能からそのパッケージのアーカイブ名が分かれば良い。したがって、パッケージの持っている機能/ 性質を端的に表したファイル (以後 short description) を用意し、それらの中から検索をしてマッチするキーワードを含む Description を持つものが求めるパッケージであるという解を返せば良い。

ただし、検索者が期待した解決方法を提供するパッケージが存在することもあるが、それとは全く違ったアプローチで解決方法を提供するものが存在することがある。例を示そう。たとえば UNIX オペレーティングシステムを用いている場合に誤ってファイルを消してしまった場合に、「消去してしまったファイルを復活するツールはないのか」という動機から検索を試みる。ところが現在普及している UNIX 系 OS ではファイルを復活することは事実上不可能で、通常は誤って消去した場合に備えてこまめにバックアップを取ったり、ファイルを消去するコマンドの代わりに一時的なゴミ箱ディレクトリに移動するツールを用いる。

したがって、UNIX を利用している人が、ファイルを復活するためのツールが欲しいという問い合わせを発した場合は、バックアップを容易に取るためのツールを解として返す、といった変換が必要である。

レベル3 このレベルの検索者は単にパッケージの名前が分からないだけで、求める機能もはっきりと表現できるレベルにある。したがって、short description を用意すればそれらの中から適切なキーワードで検索させることが可能である。

レベル4 すでにパッケージの名前が分かっているので既存ツール archie を使えば良い。

レベル1～レベル3いずれの場合も、現状では検索者が適切な検索手段を見出すことができず、NetNews に質問を投稿し、解を知っている別の読者がレベル4の問題にまで変換した上で、回答を提示するという過程をとっている。archie が現状でも頻繁に使われている理由はまさにここにあり、NetNews 上のやりとりが archie の適切なフロントエンドとして有効に機能しているからである。しかし時には逆に NetNews に質問を投げかけること自体の心理的障害から検索を断念してしまうこともある。

3 *Darts*の実装

全てのレベルの検索者の問い合わせを名前ベースの検索ツールである archie に頼っている現状に対し、コンテンツベースの検索を可能にするシステム *Darts* を提案する。

3.1 基本思想

アーカイブ検索システムを構築する上での基本理念として、

- 4 レベルいずれに属する検索者に対しても解を与えられること
- データ構築のコストがかからないこと

という事を掲げた。

このために、インターネット利用者の activity を積極的に *Darts* のデータに取り込む方針で設計した。具体的には、

- NetNews で交わされる Q/A 記事の利用
- *Darts* 利用者のインタラクションからの short description の抽出

という手法を取り込むことにより、管理者のデータ構築コストを低減させることを目指している。

3.2 Q/A 記事の特性

先に触れたように NetNews 上にはあらゆるレベルの検索者からの質問とそれに対する回答が展開されている。これを検索用データベースとして利用することで同様の問題を抱えている人の問題解決の補助とすることができる。さらに、次の特性により、全 NetNews 記事のなかからアーカイブ検索のために有効な Q/A 記事を自動抽出することが容易になっている。

一般に、NetNews に対して発せられる質問記事と、回答記事には 図.1 のような参照木が形成される。図における最初の質問に直接関わる回答記事は網かけで示した部分である。慣習的に NetNews 上でのパッケージに関する質問に対する回答記事は、

1. 元記事の質問文の引用

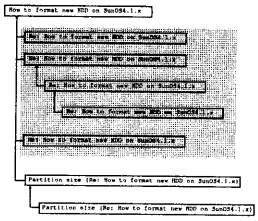

図 1: NetNews に投稿された記事のスレッド

2. パッケージの所在を表す記述 (URL など)

が含まれることがほとんどである。言い換えるならば、回答記事のみに Q と A の要素両方が含まれているということで、これはすなわち、全 NetNews 記事の中から、2のロケーションパターンを含む記事のみを抽出することで、検索に必要な情報ほぼ全てをカバーできることになるということである。また、ロケーションパターンは正規表現で記述できるのでそれ自身を含むファイルの抽出も容易である。

以上の理由により *Darts*で利用する検索用データとしては、参照木の根 (すなわち質問文そのもの) は採り入れず、回答としてロケーションパターンを含む記事のみ採り入れることとした。

3.3 システム構成

*Darts*では 4 つのレベル全ての検索に対する解を与えられることを目的としているが、レベル 4の解に関しては既存のツールである archie を呼ぶためのフロントエンドとして機能させる。レベル 2～レベル 3 に属する検索は *Darts*固有の検索部が解を与える。

以下に、*Darts*を構成するモジュールについて示す。

Location Extractor NetNews 記事からパッケージのロケーション情報を抽出して Q/A データベースを構築

Archie Interface Level 4 の検索者へのサービス

Searching Board Q/A, Desc. file データベースからの検索

Interaction Tracer ユーザのアーカイブ閲覧時の動作から各種情報を取得しフィードバックを行なう

Desc. file Extractor Interaction Tracer の取得した Desc. file を既存データベースに追加

Reporter 無効データファイルを管理者に報告

Reorganizer Q/A データベースの分類木の再構成

上記のモジュール群のうち、archie へのインタフェースを除いた部分は 図 2のような構成をとってい

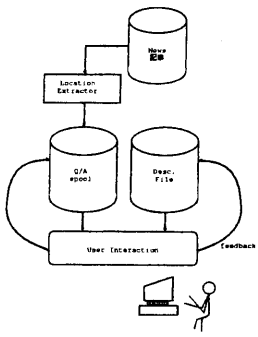

図 2: *Darts*概念図

る。`Q/A spool` は、Location Extractor が NetNews の記事から、パッケージの在処の質問と回答を検索用データとして集積したもので、おもにレベル 1～3の問い合わせをカバーすることを想定している。また、*Darts*利用者からの報告により分類が不適切と思われる記事があった場合には Reorganizer により自己再編成が行なわれる。

`Desc. file` は、Interaction Tracer と Desc. file Extractor が、*Darts*利用者の検索時の行動から各種パッケージの short description を集めたもので、おもにレベル 2～3の問い合わせをカバーすることを想定している。ユーザが与えたキーワードをそれぞれのデータから検索し、マッチする単語を含む文書をユーザに提示する。提示文書には該当パッケージへのリンクが張られており、ユーザはそのリンクを辿ることで目的パッケージに到達できる。

3.3.1 Q/A spool

NetNews から抽出された Q/A 記事群は図 3 のように、あるカテゴリ固有の概念を表す単語をもつ文書が、同一グループに属するように分類配置される。各文書は複数のグループに属すこともあり、複数グルー

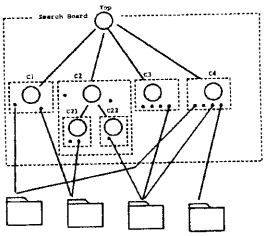

図 3: 分類木

プに存在する文書であってもファイルシステム上では同一ファイルとして存在する。各グループには、グループ内の記事のみに範囲を限定したサーチボードが付属し、検索者があらかじめ検索対象を選ぶことができるようになっている。

この分類木を生成する Location Extractor は、分類表と記事中に含まれる単語を照合し分類表に適合するパターンがあった場合にその記事を対応するカテゴリに分類する。分類表は、左辺にカテゴリ名、右辺にそのカテゴリ固有の概念を表す典型的な単語群を正規表現で記述したものである。

Location Exctractor は図 4のような二段階の 分

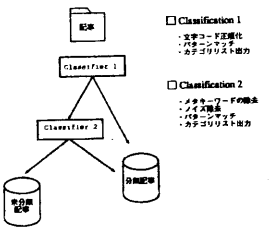

図 4: 分類の流れ

類を行っている。分類の第一段階では、記事に含まれる Newsgroup ヘッダと、Subject ヘッダのみを照合の対象として分類を行う。一般的に二つのヘッダには記事の内容にふさわしい単語が含まれていることが期待でき、仮にふさわしくない Newsgroup に投稿されていたり、ふさわしくない Subject が付けられていた場合には、回答する人がふさわしいものに変えてから投稿することが多く見受けられる。

第一段階の過程でどのカテゴリにも分類されなかったものに限り、第二段階の分類にかける。ここでは照合の範囲を記事本文にまで広げて分類を行うが、その際にメタキーワードとノイズの除去を行う。メタキーワードとは、例えば「詳細はメイルで質問して下さい」の「メイル」のような伝達手段を表すキーワードで、言及しているパッケージの属性には関係なく登場する性質のものであるため、メタキーワードになり得る単語は全て除外してから照合を行っている。ノイズとは、signature 部に含まれたロケーションパターンや、例えば「# Wnn のちょうしがわるいのでひらがなでしつれい」などのような、文脈とは全く関係のない文章のことである。これらも同様に照合時に除外する必要があるが、文脈を完全に把握することは非常に困難であるため、*Darts*では signature 部の除去のみ行なっている。

3.3.2 Desc. file

これは各パッケージの short description を収集する部分であるが、Internet 上に存在する全てのパッケージの short description を用意することは事実上不可能と言って良い。そこで *Darts*では、*Darts*利用者のインタラクションから short description を生成する手法をとっている。*Darts*が提示した全ての解からユーザがパッケージを取得する際に、希望があればアーカイブ中のファイルリストを表示して任意のファイルを閲覧させる。そしていずれかのファイルを読んでいる時点でそのパッケージが所望のものだと判断した場合、そのとき読んでいたファイルがパッケージの性質を良く表しているとみなし、これをそのパッケージの short description として Desc. file データベースに追加登録し、それを次回以降の検索用データとして利用する。

4 検討と評価

4.0.3 Q/A 記事分類木生成

1995 年 12 月から 1996 年 8 月までの fj 全 NewsGroup のロケーションパターンを含む全記事を対象に分類を試みた結果を以下に示す。

表 1の Subject と NewsGroup による分類率を見て分かる通り、ヘッダ部分の照合だけで全体の 87%の Q/A 記事の分類が可能である。通常ヘッダにはノイズやメタキーワードが含まれにくいことを考えると、Q/A 記事のかなりの部分が適切なカテゴリに分類さ

表 1: 分類成績

全記事数	4715	—
Subject による分類	2759	58.5%
NewsGroup による分類	1353	28.7%
記事内容による分類	391	6.8%
分類不能	212	4.5%

表 2: アクセスデータ

全アクセス	トップページ			1257	
	全ページ			26858	
名前不明 (Level 1,2,3)	カテゴリ別メニュー参照	分類済		2511	
		未分類		102	
	Q/A 記事ファイル参照	分類済		4829	
		未分類		75	
archie 起動 (Level 4) 752	検索成功			570	
	検索失敗			182	
Q/A Desc. からの検索 893	topmenu から	37	検索成功	686	
	分類済 menu から	828			
	未分類 menu から	28	検索失敗	207	
目的アーカイブに到達 4283	Darts DB から	3101	内容確認あり	1445	
			内容確認なし	1656	
	archie から	1182	内容確認あり	729	
			内容確認なし	453	

れるということが言える。

4.1 アクセス状況

*Darts*を一般アクセスを可能にした1995年末より現在までのアクセスデータについて分析する。

表2によれば、名前ベースの archie 経由の検索よりも、コンテンツベースの検索のほうが高頻度で動作していることがわかる。このことより、レベル1〜3に属する検索者の潜在的な多さが明らかになった。

さらに、目的アーカイブ到達の項を見ると、archie 経由でアーカイブを取得した場合の内容確認率が61.7%であるのに対し、*Darts*の DB を検索の礎としてアーカイブを取得した場合の内容確認率が46.6%であることから、コンテンツベースでの検索が、より目的アーカイブへの到達ステップを削減していることが伺える。

5 今後の課題

*Darts*所期の目的からすると有効性をあまり見出せなかった点がある。ユーザインタラクションからの short description の抽出である。現時点までに抽出できた description file はわずか 32 件であり、目的アーカイブに到達した延べ件数4283件から見ると極めて少ない数であることから、ユーザになんらかの contribution を課すことの難しさが理解できた。今後は、ユーザがアクセスしたアーカイブの中から、short description をシステムが自主的に抽出してデータベースに加えるといった機構が望まれる。

また、本システムのアクセス許可を与えて以来、一般の検索ロボットのデータ収集はブロックしなかった。このため、*Darts*自身のサーチボードを経由せず、Alta Vista や Lycos などに代表される検索エンジンからのリンクを辿ってアクセスされることが多かった。このことは*Darts*のもつデータベース再構成の機会を増やすと言う意味では有効なのだが、検索者が*Darts*の持つデータベースにどのようなキーワードで検索を行ない、どのようなパッケージを解として得たかの情報を得るためには不利であった。今後は、*Darts*固有の検索システムからの検索のみを許可し、検索キーワードと目的アーカイブの対応関係を明らかにしたい。

6 まとめ

本稿ではコンテンツベースのアーカイブ検索システムの有用性を述べた。今後は、本システムをインターネットコミュニティ全体で実用的に使える規模と強度に改善して行きたいと考えている。

参考文献

[1] Shirley Browne, Jack Dongarra, Stan Green, Keith Moore; Location-Independent Naming for Virtual Distributed Software Repositories; ACM-SIGSOFT Symposium on Software Reusability (SSR'95)

[2] C. Mic Bowman, Peter Danzig, Udi Manber, and Michael Schwartz; Scalable Internet Resource Discovery: Research Problems and Approaches; Comm. of the ACM, Vol. 37, no. 8, pp. 98-107, 114(Aug 1994).

[3] K. Sollins; Functional Requirements for Uniform Resource Names; RFC1737, MIT/LCS, Dec.1994