

特別研究報告

題目

Java ソフトウェア部品検索システム SPARS-J のための
リポジトリ自動更新機能の実現

指導教員

井上 克郎 教授

報告者

木村 和正

平成 18 年 2 月 20 日

大阪大学 基礎工学部 情報科学科

Java ソフトウェア部品検索システム SPARS-J のための
リポジトリ自動更新機能の実現

木村 和正

内容梗概

近年のソフトウェアの大規模化と複雑化に伴い、ソフトウェア部品の再利用の為に技術が必要とされており、その1つとしてソフトウェア検索システムがある。我々の研究チームでは、Java のソースコードを対象としたソフトウェア部品検索システム SPARS-J の構築を行っている。SPARS-J では、あらかじめ検索に利用するための部品情報を保持したリポジトリを構築する。このとき、リポジトリに登録するソフトウェア部品は、ユーザが Java ソースコード集合を明示的に指定することで決定される。そのため、Java ソースコードが新規作成・更新された際には、その都度ユーザがリポジトリへの登録作業を行う必要がある。また、検索対象が Java ソースコードに限定されているため、ソースコードに関連するドキュメントなどの情報を得るためには、別の検索システムを利用する必要がある。

そこで本研究では、SPARS-J のリポジトリの自動更新機能を実現する、SPARS Desktop を開発した。SPARS Desktop は、デスクトップ検索ツール Google Desktop を利用した、SPARS-J の拡張である。Google Desktop とは、Google 社が開発した、ローカルディスクやネットワークドライブ内に存在するファイルのインデックス化を自動的に行い、キーワード検索を可能にするデスクトップ検索ツールである。このインデックス作成機能を利用して、SPARS-J のリポジトリの自動更新を実現した。これにより、常に最新の状態に保たれたリポジトリからの、ソフトウェア部品検索が可能となる。さらに、Google Desktop を用いて検索した Java ソフトウェア部品を SPARS-J を用いて表示する機能を、SPARS Desktop に実装した。この機能により、Java ソフトウェアの部品情報と関連情報を同時に検索・閲覧することができる。SPARS Desktop を利用することで、ソフトウェア部品の検索・再利用コストを軽減することを、実装したシステムを適用実験により評価することで確認した。

主な用語

自動更新

ソフトウェア部品検索

デスクトップ検索

リポジトリ

Java

目次

1	まえがき	4
2	背景	6
2.1	Java ソフトウェア部品検索システム SPARS-J	6
2.1.1	SPARS-J の概要	6
2.1.2	システム構成	7
2.1.3	使用法	10
2.2	デスクトップ検索ツール Google Desktop	10
2.2.1	デスクトップ検索	10
2.2.2	Google Desktop の特徴	10
2.3	本研究との接点	12
3	システムの機能と実装	13
3.1	システムの機能	13
3.2	システムの構成	13
3.3	実装	15
3.3.1	Google Desktop 登録部	15
3.3.2	SPARS-J 登録部	16
3.3.3	表示加工部	17
4	評価	21
4.1	方針	21
4.2	SPARS Desktop 導入前後の作業工程	21
4.2.1	リポジトリの更新	21
4.2.2	関連情報の取得	22
4.3	作業工程の比較と考察	25
5	まとめと今後の課題	28
	謝辞	29
	参考文献	30

1 まえがき

ソフトウェアの大規模化と複雑化に伴い、高品質なソフトウェアを一定期間内に効率良く開発することが重要になってきている。これを実現するために、近年のソフトウェア開発において、再利用を用いた開発が頻繁に行われている。再利用とは、既存のソフトウェア部品を同一システム内や他のシステムで利用することを指し、開発期間の短縮や品質向上が期待できるといわれている [1, 2, 4, 5]。ソフトウェアの再利用による効果を最大限に引き出すためには、開発者が開発しようとするソフトウェアに必要な部品及びライブラリに関する知識を持つことが重要であるが、知識の共有が満足になされていないために、同種のプログラムが別々の場所で、独立して開発されている事も多い。そこで既存の部品の中から、開発者の必要としている機能を持つ部品、その機能の使い方を示している部品のような、再利用に有益な情報を提供する検索システムを実現する事で、知識の共有が実現でき、再利用を促進する事ができると考えられる。

我々の研究チームでは、Java ソースコード集合を対象としたソフトウェア部品検索システム SPARS-J (*Software Product Archive, analysis and Retrieval System for Java*)[10] の開発を行っている。SPARS-J は、依存や類似といったソフトウェア部品特有の性質を考慮しながら、大規模なライブラリの分類・検索を自動的に行なうシステムである。検索時には、キーワードとトークン種類を検索キーとした全文検索を行い、依存や類似を考慮することでソフトウェアのソースコードを効率良く検索することが可能である。さらに、検索結果表示の際にソフトウェア部品に関する詳細な情報を併せて提供する。このシステムを用いることで、ライブラリの知識が無い開発者も有用なソフトウェア部品やそれに付随する有益な情報を容易に入手することができる。

SPARS-J では、あらかじめ検索に利用するための部品情報を保持したリポジトリを構築する必要がある。このとき、リポジトリに登録するソフトウェア部品は、ユーザが Java ソースコード集合を明示的に指定することで決定される。そのため、ソースコードが新規作成・更新される度に、ユーザによるリポジトリへの登録作業が発生するという問題がある。また、検索対象が Java ソースコードに限定されているため、プログラムのドキュメントや、開発に関するメールなどの情報を得るためには、別の検索システムが必要となる。そのため、同様の検索作業を異なるシステムに対して行わなければならないという問題もある。

そこで本研究では、リポジトリの自動更新を実現する、SPARS Desktop を開発した。SPARS Desktop は、デスクトップ検索ツール Google Desktop[3] を利用する、SPARS-J の拡張である。Google Desktop とは、Google 社が開発した、ローカルディスクやネットワークドライブ内に存在するファイルのキーワード検索を可能にするデスクトップ検索ツールである。このツールでは、CPU がアイドル状態の時に既存ファイルのインデックス化を自

動的に行う。そのため、計算機のパフォーマンスに影響することなく、検索候補となるファイルの情報を常に最新の状態に保つことができる。SPARS Desktopはこのインデックス作成処理を利用して、新規作成・更新されたJavaファイルの情報を取得することで、SPARS-Jのリポジトリの自動更新を実現する。また、Google Desktopでは多様なファイルを混在したまま検索することができる。そこで、Google Desktopの検索機能により得られる検索結果に対して、SPARS-Jのソフトウェア部品情報を提供することで、Javaソフトウェア部品情報と関連情報を一度の検索で取得することが可能となる。

以下、2節でSPARS-J及びGoogle Desktopについて説明する。そして、3節で提案するシステムの実装について述べ、4節で実装したシステムの評価を行い、その有効性を検証する。最後に5節でまとめと今後の課題について述べる。

2 背景

本節では，Java ソフトウェア部品検索システム SPARS-J と，デスクトップ検索ツール Google Desktop について述べる．

2.1 Java ソフトウェア部品検索システム SPARS-J

2.1.1 SPARS-J の概要

SPARS-J は，Java の 1 つのクラスやインタフェースのソースコードを部品として扱うソフトウェア部品検索システムである．依存や類似といったソフトウェア部品特有の性質を考慮しながら，大規模なライブラリの分類・検索を自動的に行う．検索時には，キーワードとトークン種類を検索キーとした全文検索を行い，ソフトウェアのソースコードを効率良く検索することが可能である．さらに，検索結果表示の際にソフトウェア部品に関する詳細情報を併せて提供する．このシステムを用いることで，ライブラリの知識が無い開発者も有用なソフトウェア部品やそれに付随する有益な情報を容易に入手することができる．

検索システムとしての形態は自然言語文書を対象とした全文検索システムと似ているが，ソフトウェア部品間の利用関係や類似関係を可視化する部品群グラフの構築や，利用関係に基づく順位付けにより，ソフトウェア部品検索に特化した機能を提供する．

以下では，ソフトウェア部品における利用関係や類似部品，検索結果の順位付け手法など，SPARS-J の根幹を成す概念について簡単に説明する．

利用関係と類似部品

ソフトウェアは構成要素の部品間で相互に属性や振る舞いを利用し合うことで，1 つの機能を提供する．ここで，ある部品がある部品を利用するとき，この部品間に利用関係が存在するという．

ある部品をコピーした部品や，コピーして一部を変更しただけの部品を類似部品と呼ぶ．類似部品の集合を部品群と呼ぶ．SPARS-J では部品の集合を部品群に分類するために，任意の部品間の類似度をメトリクスを用いて定量的に評価し [9]，閾値以上の類似度をもつ部品を同一部品群として分類する．

また，ある部品群に属する部品が他の部品群に属する部品を利用している場合には，その 2 つの部品群間に利用関係が存在するという．

検索結果の順位付け

ソフトウェア部品の検索を行う際，検索キーと適合度が高い部品を上位に位置付けることが重要であると同様に，よく利用されている部品を上位に提示することも重要である．

なぜなら，そのような部品は利用例も多く，部品の再利用や理解をスムーズに行いやすいと考えられるからである．そこで SPARS-J では，検索キーと部品の適合度を表す指標と共に，よく利用されている部品であるかを定量的に評価するための指標を用いて検索結果を順位付けする．以下では，適合度に基づく順位付け手法である KR 法 (Keyword Rank 法) と，利用関係に基づく順位付け手法である CR 法 (Component Rank 法) についての詳細を述べる．

KR 法

索引語の中には部品の内容と密接に関係したものもあれば，関係の薄いものも存在する．抽出された索引語が部品の内容を表す上でどれだけの重要度を持っているか測ることができれば，より精度の高い検索を実現できると考えられる．そこで，索引キーと検索キーの適合度に基づいて検索結果を順位付けする．このときの適合度の算出には，任意の部品中における特定の索引キーの出現頻度 TF (*Term Frequency*)，及び特定の索引キーを含む部品数の逆数 IDF (*Inverse Document Frequency*) の値をもとにして適合度を算出する TF-IDF 法 [8] を用いる．TF-IDF 法は自然言語文書検索システムで用いられる一般的な手法である．

SPARS-J では，索引キーの重みを算出する際にそのトークン種類に応じて異なる重みを与えている．本手法を KR 法と呼び，測定した適合度の値を KR 値，さらにその順位付けの結果を KR と呼ぶ．

KR 値は次のように求める．まず，検索キー毎に，索引キーのトークン種類の重みと出現回数を乗算し正規化した値を求める．次に，その正規化した値に，ヒットした部品数を総部品数で割った値の逆数を乗算する．この総和が部品の KR 値となる．

CR 法

我々の研究チームではこれまでに，ソフトウェア部品間の利用関係に基づいて順位付けする手法 (CR 法) を提案している [11]．CR 法では，部品の集合に対して，各部品間に存在する利用関係に基づいてグラフ及び行列を構築し，構築された行列に対して繰り返し計算を行うことで各部品を評価する．求められる値は，開発者が利用関係に沿って参照を行うと仮定した場合の各部品の参照されやすさを表しており，よく利用される部品や，重要な部品から利用される部品の順位は高くなる．

2.1.2 システム構成

SPARS-J は，部品登録部と部品検索部という 2 つの処理部と，登録された情報を蓄積しているリポジトリから構成される．構成図を図 1 に示す．

以下，図中のシステムの各構成要素について説明する．

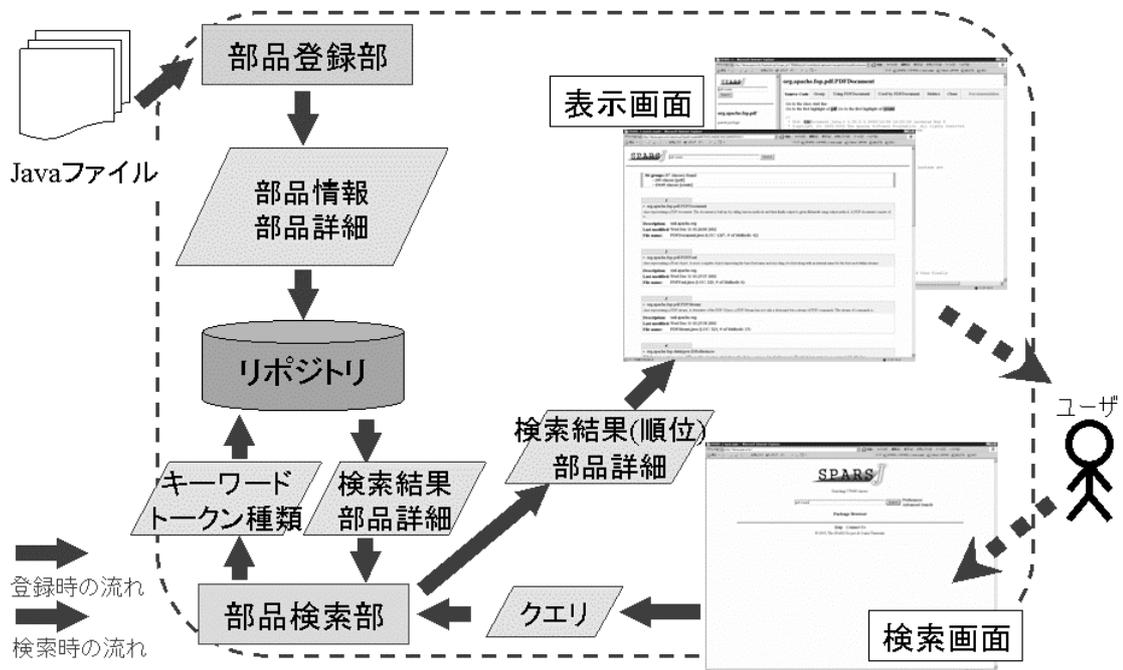


図 1: SPARS-J のシステム構成

リポジトリ

ライブラリ中のファイル情報，登録された部品に関する情報，部品間の利用関係，索引情報の管理を行う．ファイル，部品，索引キーには一意な ID が割り振られている．各リポジトリは，ID からファイル名や部品名，ファイル名や部品名から ID を対応付けるために正引き，逆引き索引を持つ．

- ファイル情報リポジトリ

ライブラリ中の Java ファイルのファイルパス，ファイル ID，各ファイルに存在する部品の ID，更新時刻などの情報が格納されている．

- 部品情報リポジトリ

部品名，部品 ID，部品が存在するファイル ID，ファイル中の部品の場所など，部品とライブラリ中のファイルを対応付けるための情報の他に，各部品中で定義されたメソッド名やフィールド名，類似判定のためのメトリクス，部品が属する部品群 ID，CR

値，部品中に現れた索引キー ID や出現頻度，場所など部品に関する全情報が格納されている．

- 利用関係リポジトリ

部品間の利用関係と，その種類が格納されている．また利用関係とは別に，未解決名情報も格納されている．登録過程でまだ登録されていない部品への利用関係があった場合，その利用部品名を未解決名として格納する．その後の解析において，未解決名に相当する部品が登録された場合，利用関係を追加する．

- 索引情報リポジトリ

登録された部品中で出現した全索引キーとその ID が格納されている．

部品登録部

部品登録部はライブラリ中のファイルを解析して，索引，メトリクスの計算，利用関係の抽出，さらに類似度の測定を行い，結果をリポジトリに格納する．部品登録時の流れは次のようになる．

1. ユーザが Java ファイルを登録
2. Java ファイルを解析し，各種情報をリポジトリに格納
3. 部品群グラフを構築
4. CR 法による順位付け

部品検索部

ユーザに指定されたクエリを解析し，検索キーを抽出してそれをもとに検索を行う．得られた結果を KR 法で順位付けし，部品登録時に算出した CR と KR の順位の統合を行う．検索結果と併せて，部品の詳細情報としてソースコードや利用関係，メソッド一覧などをウェブインタフェースを介して提供する．部品検索時の流れは次のようになる．

1. 入力されたクエリを解析
2. リポジトリから部品を検索
3. KR 法による順位付け
4. CR と KR の順位の統合

2.1.3 使用法

SPARS-J には 2 通りの使用法がある。

SPARS-J は、大規模資産からの高速な検索が可能である。そこで、大量の Java ソースコードを集めた公開サーバを設置し、ユーザがそのサーバにアクセスすることで、Java ソフトウェア部品の検索・再利用を行うという使用法がある。

2 つ目は、個人で Java ソースコードを管理し、再利用するために、各計算機に SPARS-J をインストールして使用するという方法である。

本研究では、2 つ目に挙げた、個人の計算機に SPARS-J をインストールして使用することを想定する。

2.2 デスクトップ検索ツール Google Desktop

2.2.1 デスクトップ検索

デスクトップ検索とは、ウェブブラウザを通じてコンピュータに保存された様々な種類のファイルを全文検索する技術、またその機能を実装したソフトウェアを指す。

ウェブ検索エンジンの技術を応用し、コンピュータに保存されたウェブページ (HTML ファイル) やテキストファイル、PDF ファイル、ワープロソフトや表計算ソフトで作成されたファイルなどを全文検索することができる。結果は日付順や適合度順などで並べ替えて表示することができる。ソフトウェアによっては、ウェブ検索の結果と統合して表示することも可能である。

代表的なツールとして、Google Desktop、Windows Desktop Search[6]、Yahoo! Desktop Search[7] などがある。次節では、その中から Google Desktop の特徴について述べる。

2.2.2 Google Desktop の特徴

Google Desktop とは、Google 社が開発、提供しているデスクトップ検索ツールである。Google Desktop では、各種ファイルの全文検索機能に加えて、パーソナライズしたウェブ上の情報をサイドバーに表示することが可能である。

以下にその主な特徴を示す。

- 各種ファイルの検索

ローカルディスク、ネットワークドライブ内の以下のファイルを検索できる。

- Gmail、Outlook 2000 以降、Outlook Express 5 以降、Netscape Mail 7.1 以降、Mozilla Mail 1.4 以降、Thunderbird のメール。

- テキスト , Word , Excel , Powerpoint , PDF , 画像 (JPEG , GIF , PNG , BMP , Photoshop など) , オーディオ , ビデオ などのパソコン内のファイル . メディアファイルのメタタグも検索可能 .
 - Internet Explorer 5 以降 , Netscape 7.1 以降 , Mozilla 1.4 以降 , Firefox で閲覧したウェブページ .
 - AOL 7 以降 , AOL インスタント メッセンジャー 5 以降 , MSN メッセンジャーのチャット .
- インデックス作成
CPU がアイドル状態のときに各ファイルからインデックスの作成を行う . インデックスとは , キーワードとファイルの内容を関連付けた索引のことである . システムは , この事前に作成されたインデックスをもとに検索を行う . Google Desktop では , インデックスに反映される単語の抽出には形態素解析が用いられている .
 - 検索結果の表示方法
検索結果は , 日付順及び関連性の高い順序により表示される . ここでの関連性は , 検索されるファイル中のキーワードの数や位置によって決定される .
また , ファイルのタイプによって特定のタイプの検索結果のみを表示したり , メールであれば差出人や宛先で検索結果を表示することも可能である .
 - ウェブ検索結果との統合
Google によるウェブ検索の結果と , デスクトップ検索の結果を併せて表示することができる .
 - プラグインによる機能拡張
Google 社から提供されている , Google Desktop SDK を利用することで , 機能拡張を行うためのプラグインを開発することができる . プラグインを開発 , 使用することで , インデックスの作成やコンテンツの保存 , 検索できるファイル形式やアプリケーション (他のインスタントメッセージプログラムやウェブブラウザなど) の追加などを行うことができる .
 - サイドバー
デスクトップ上にサイドバーを表示し , ユーザが設定した様々な情報に簡単にアクセスすることが可能となる . 以下にサイドバーの機能の一部を示す .
 - Gmail などの新着メールの確認

- サイドバーやウェブブラウザで閲覧したニュース記事に基づく、カスタマイズされたニュースの見出しの表示
- ウェブページの RSS/Atom フィードの表示
- よく使用するウェブページやファイルのリストの表示

2.3 本研究との接点

2.1.2 節で述べたように、SPARS-J では、あらかじめ検索に利用するための部品情報を保持したりリポジトリを構築する必要がある。このとき、リポジトリに登録するソフトウェア部品は、ユーザが Java ソースコード集合を指定することで決定される。そのため、新たなソースコードを入手・作成した場合、あるいは既存のソースコードに修正を加えた場合などには、ユーザが手作業でリポジトリに追加登録する必要がある。この登録作業が随時行われなければ、SPARS-J が正確な情報をユーザに与えることが保証されない。そのため、このようリポジトリの更新作業を自動化することで、ソフトウェア部品の検索・再利用にかかる作業コストを軽減することができると考えられる。また、SPARS-J では検索対象が Java ソースコードに限定されているため、ソースコードに関連するドキュメントなどの情報を得るためには、別の検索システムが必要とされる。これらの情報を一度に取得することができれば、検索作業のコストを軽減することができると考えられる。

そこで本研究では、リポジトリの自動更新を行う方法として、デスクトップ検索ツール Google Desktop を利用し、SPARS-J の拡張を行う。2.2.2 節で述べたように、Google Desktop は CPU がアイドル状態のときにインデックスの作成を行い、検索候補となるローカルディスクやネットワークドライブ内のファイル情報を最新の状態に保つことができる。また、プラグインによる機能拡張が比較的容易であるため、プラグインを開発することで、SPARS-J のリポジトリの自動更新機能を実現できると考えられる。さらに、デスクトップ検索機能により得られる検索結果に対して、SPARS-J のソフトウェア部品情報を提供することで、Java ソフトウェア部品情報と関連情報を一度の検索で取得することも実現できる。

3 システムの機能と実装

本研究では，リポジトリの自動更新を実現する，SPARS Desktop を開発した．SPARS Desktop は，2.2.2 節で取り上げたデスクトップ検索ツール Google Desktop を利用して，Java ソフトウェア部品検索システム SPARS-J の機能拡張を行う．

本節では，実装したシステム，SPARS Desktop の詳細について述べる．

3.1 システムの機能

本システムは，SPARS-J 及び Google Desktop と連携することで，以下の機能を実現する．

- リポジトリの自動更新
ローカルディスクやネットワークドライブ内の Java ファイルが新規作成・更新されると，そのファイルを SPARS-J のリポジトリに登録する．その際，部品の利用関係情報の登録や，検索結果の順位付けも適宜行われる．
- Google Desktop の検索結果からのソフトウェア部品の詳細な情報の取得
Google Desktop を利用して検索を行った際に，検索結果として Java ファイルが得られることがある．このとき，そのファイルに関する Java ソフトウェア部品の詳細な情報を参照することが可能である．この情報は SPARS-J により提供される．これにより，1 度の検索クエリの入力から，詳細な Java ソフトウェア部品の情報とともに，これに関連する資料を閲覧することが可能となる．

3.2 システムの構成

本システムは，登録部と表示加工部の 2 つの処理部から構成される．SPARS Desktop を含めたシステムの構成図を，図 2 に示す．

SPARS Desktop は，SPARS-J 及び Google Desktop との間に立ち，データの受け渡しを行いながら処理を進めていく．これら 3 つのシステムにおける処理の流れは次のようになる．

登録処理

- クローラーが更新された Java ファイルを発見
- Google Desktop がイベントを生成
- 登録部がイベントを受け取る
- 登録部が Java ファイル情報を取得

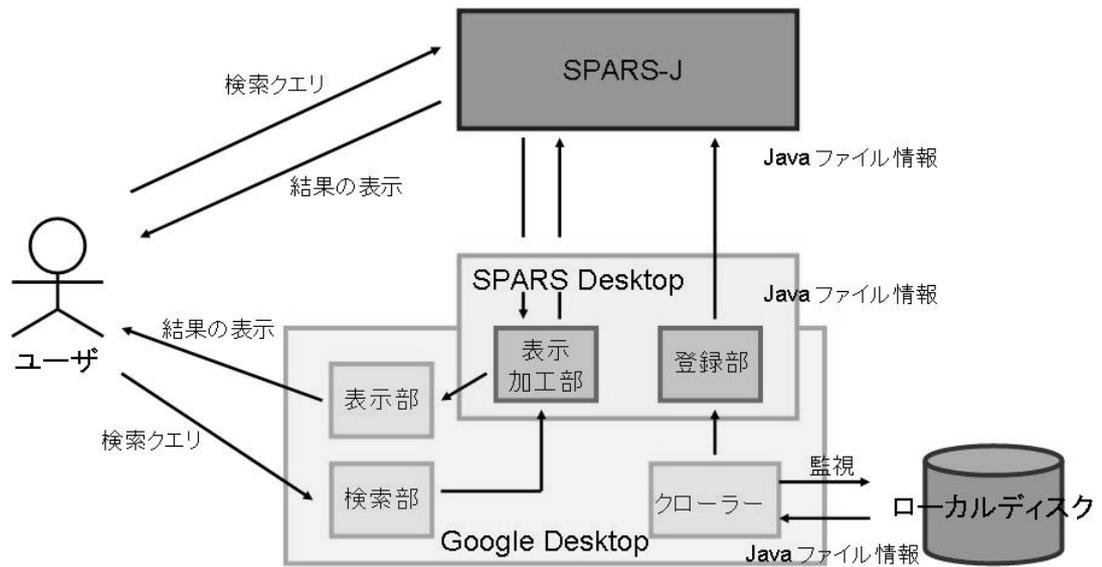


図 2: システムの構成

- 登録部が Google Desktop へファイル情報を登録
- 登録部が Java ファイル情報を SPARS-J へ登録

検索結果表示処理

- ユーザが Google Desktop へ検索クエリを入力
- Google Desktop が検索結果表示
- ユーザが閲覧するファイルを選択
- ▶ 選択されたファイルが Java ファイルであれば
 - 表示加工部が SPARS-J へファイル情報を送信
 - SPARS-J が部品情報を表示

▶ そうでなければ

- 規定のアプリケーションでファイルを開く

3.3 実装

以上の機能を実現するシステムの実装を行った。

今回実装したシステムでは、実装上、登録処理部が Google Desktop 登録部と SPARS-J 登録部の 2 つの処理部に分かれている。システム全体を Google Desktop のプラグインとして実装した場合、プラグインから直接 SPARS-J への登録処理を行うこととなる。このとき、複数の Java ファイルが同時に更新されると、全てのファイルの登録作業が完了するまでプラグインの処理が中断されてしまうという問題がある。そこで本システムでは、Google Desktop 登録部及び表示加工部を Google Desktop のプラグインとして実装し、SPARS-J のリポジトリへの登録処理を行う SPARS-J 登録部をプラグインから切り離して実装することとする。プラグインの実装にあたっては、Google 社から提供されている Google Desktop SDK を利用した。

以下、各処理部についての詳細を述べる。

3.3.1 Google Desktop 登録部

新規作成・更新されたファイルの情報を取得し、Google Desktop へ登録する。また、SPARS-J 登録部へもファイル情報を送る。

拡張子の登録

本システムにおいて、ローカルディスク及びネットワークドライブ内の Java ファイルの監視を行うのは、Google Desktop である。しかし、Google Desktop は Java ファイルに対するインデックスの作成及び検索をサポートしていない。そこで、Google Desktop SDK に用意されている、`IGoogleDesktopRegisterIndexingPlugin` インタフェースを用いて、拡張子“java”を Google Desktop に登録し、Java ファイルの検索を可能にする。

SPARS-J 登録部の起動

Google Desktop が発見した Java ファイルは、SPARS-J 登録部を通して SPARS-J のリポジトリへと登録される（SPARS-J 登録部の詳細は 3.3.2 節）。Google Desktop 登録部は、初回ファイル登録時に SPARS-J 登録部の起動を行う。

ファイル情報の取得

Google Desktop 及び SPARS-J への登録のためのファイル情報を取得する。

Google Desktop が登録された拡張子のファイル（本システムでは Java ファイル）を発見すると、HandleFile メソッドが呼び出される。HandleFile メソッドは、引数として、取得したファイルパスと Google Desktop に送信するためのイベントオブジェクトへのポインタを持つ。このファイルパスをもとに、ファイルの最終更新時刻、ファイルの内容を取得する。

Google Desktop へのファイル情報の登録

Google Desktop での検索結果に反映させるためのファイル情報を登録する。

Google Desktop SDK に用意されている、IGoogleDesktopEventFactory インタフェースを用いて、ファイルパス、最終更新時刻、ファイルのフォーマット、ファイルの内容を登録する。

SPARS-J 登録部へのファイル情報の送信

発見した Java ファイルを SPARS-J へ登録するために、SPARS-J 登録部にファイル情報を送信する。

SPARS-J 登録部との通信ではソケットを用いる。SPARS-J 登録部に対して接続要求を出し、接続が受理されれば、ファイルパス、最終更新時刻を送信する。その後、SPARS-J 登録部より、SPARS-J へのファイル情報の登録が完了したことを知らせるメッセージを受信すると、通信を停止する。

3.3.2 SPARS-J 登録部

Google Desktop 登録部から送られてきたファイル情報をもとに、SPARS-J への登録作業を行う。SPARS-J 登録部はデーモンとして実装し、初回起動以降 Google Desktop 登録部からのファイル情報の送信を待つ。

ファイル情報の受信

Google Desktop 登録部からの接続要求を受けて接続を受理した後、受信バッファ上のファイル情報を受信する。

SPARS-J への登録

受信したファイル情報をもとに、更新された Java ファイルを SPARS-J へ登録する。登録処理の流れは以下のようになる。

1. SPARS-J のリポジトリのオープン
2. リポジトリへの登録

3. 部品，キーワードの総数のカウント
4. 部品の順位付け
5. 検索順位のソート
6. リポジトリのクローズ

この中で，3～6 は，初回登録時以降，一定のファイル数が登録されるごとに行われる．また，1 については，初回登録時及び，クローズ処理が行われた後の Java ファイルの登録時に行われる．これらの登録作業は，SPARS-J の各メソッドを利用することで行われる．

3.3.3 表示加工部

Google Desktop の検索結果中の Java ファイルに対して，SPARS-J が提供する，利用関係や類似部品，メトリクス情報などの Java ソフトウェア部品に関する情報の表示を可能にする．

カスタムアクションコンポーネントの登録

表示加工部は，Google Desktop の規定の表示部の処理の上書きを行う．本システムでは，Google Desktop の検索結果中の Java ファイルに対して，SPARS-J での表示を行う．そこで，Google Desktop SDK に用意されている，`IGoogleDesktopRegisterCustomAction` インタフェースを用いて，`ACTION_OPEN_SEARCH_RESULT` を上書きできるように設定する（カスタムアクションコンポーネントの登録）．これにより，検索結果の表示画面において，ユーザがファイル名をクリックしたときの処理を上書きすることが可能になる．

SPARS-J による表示

Google Desktop の検索結果として表示されたファイルのうち，Java ファイルに対して SPARS-J による情報を参照できるようにする．

まず，ユーザが Google Desktop 上で検索クエリを入力すると，Google Desktop によってクエリに対する検索結果が表示される（図 3）．

その検索結果の中に，ユーザの望む Java ファイルが表示されたとする．ユーザがそのファイル名をクリックすると，ダイアログを表示する（図 4）．

ダイアログ中の「はい」ボタンをクリックすると，Google Desktop SDK に用意されている，`IGoogleDesktopNotifyEvent` インタフェースの `GetProperty` メソッドを用いて，選択したファイルのファイルパス取得する．そして得られたファイルパスをもとに，対応し

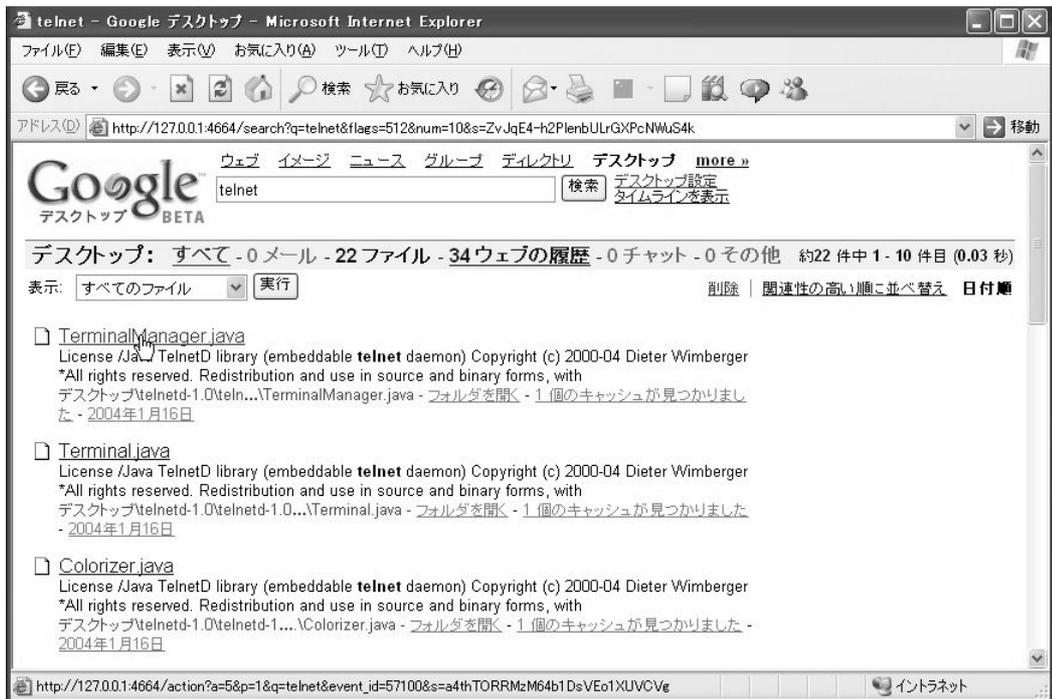


図 3: 検索結果の表示

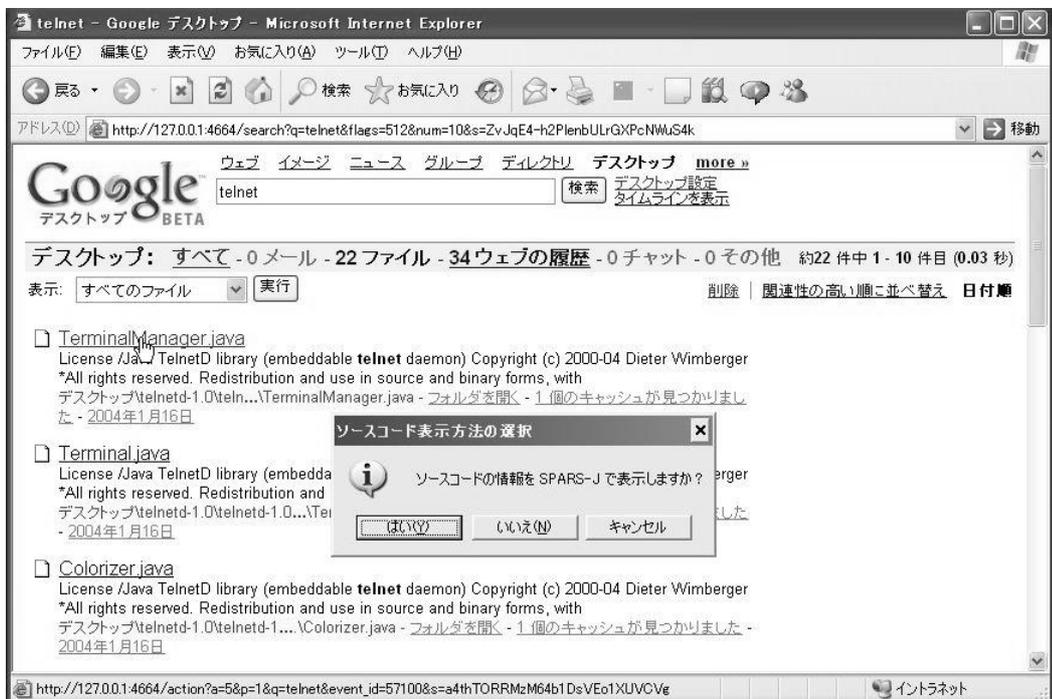


図 4: ダイアログの表示

た SPARS-J の部品情報ページを開く (図 5) . 「いいえ」 ボタンをクリックすると , 通常の Google Desktop と同様に , 規定のアプリケーションにより , ソースコードを表示する .

4 評価

本研究において実装したシステム，SPARS Desktop の有効性を検証する．

4.1 方針

SPARS Desktop を利用することで，Java ソフトウェア部品の検索・再利用にかかる作業コストが軽減されるかどうかを評価する．評価は，SPARS Desktop 導入時と未導入時において，以下の観点から作業工程を比較することで行う．

- リポジトリの更新
- 関連情報の取得

ここで，SPARS Desktop 未導入時の場合には，大規模な Java ソースコード集合を保持する公開サーバを設置して SPARS-J を利用するのではなく，個人の計算機中の Java ソースコード資源に対して，SPARS-J を利用できるものとする．また，Google Desktop は，Java ファイルに対するインデックスの作成・検索をサポートするプラグインを導入した状態で利用できるものとする．

4.2 SPARS Desktop 導入前後の作業工程

4.2.1 リポジトリの更新

SPARS-J による Java ソフトウェア部品の検索・再利用を正確に行うためには，Java ファイルを新規作成・更新した際に，同時にリポジトリの情報も更新する必要がある．そこでここでは，Java ファイルを新規作成・更新した際の，リポジトリの更新作業について考える．

未導入時

ユーザが新たに Java ファイルを入手・作成したり，既存の Java ファイルを更新したとする．するとユーザは，Java ファイルの更新を SPARS-J のリポジトリに反映させるために，SPARS-J を起動する．ユーザは，SPARS-J のインタフェースから登録する Java ソースコード集合を指定し，登録処理を実行する（図 6）．

導入時

ユーザが新たに Java ファイルを入手・作成したり，既存の Java ファイルを更新したとする．すると，SPARS Desktop が自動的にバックグラウンドで登録処理を行う．ユーザは，登録処理を意識する必要は無い．

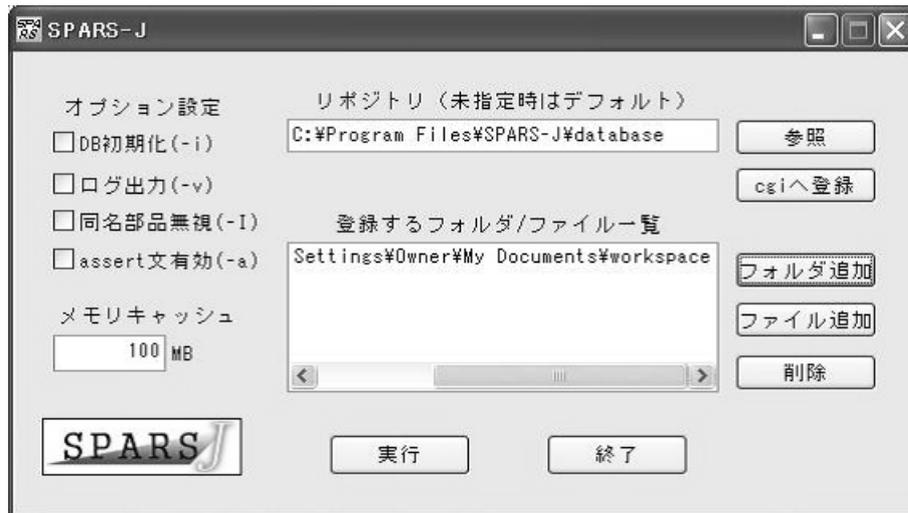


図 6: SPARS-J の登録画面

4.2.2 関連情報の取得

ソフトウェアを再利用する際に、ソースコードに基づくソフトウェア部品そのものの情報だけでなく、開発に関するメールやプログラムのドキュメントなど、関連するファイルからの情報も有用であると考えられる。そこでここでは、再利用したい Java ソースコードファイルと共に、これに関連する情報を持つファイルも同時に検索する場合として、telnet プロトコルを実装したソフトウェア部品とそれに関連する情報の検索作業を想定する。

未導入時

ユーザは Java ソフトウェア部品を再利用するために、Java ソフトウェア部品の検索に特化したシステムである、SPARS-J を利用する。そこで、SPARS-J の検索インタフェースに検索クエリ “telnet” を入力し、検索結果を得る (図 7)。検索結果から、目的のソフトウェア部品を発見すると、その部品を選択し (図 7)、詳細な部品情報を閲覧する (図 8)。

しかし、これらの情報だけではユーザにとってソフトウェア部品の理解が難解であるため、この部品の理解に役立つ他の情報を取得する必要がある。そこで、ユーザは一般の全文検索ツール (ここでは Google Desktop を用いる) を用いて再度検索を行う (図 9)。すると、この検索結果から、目的の Java ソフトウェア部品に関連する Java ファイル以外のファイルを得ることができる。

このように、SPARS-J を用いた検索では、検索対象が Java ソースコードに限定されているため、次に、ユーザが初めから Google Desktop を用いて検索を行うことを考える。そこで、Google Desktop の検索インタフェースに検索クエリ “telnet” を入力し、検索結果を得



図 7: SPARS-J による検索

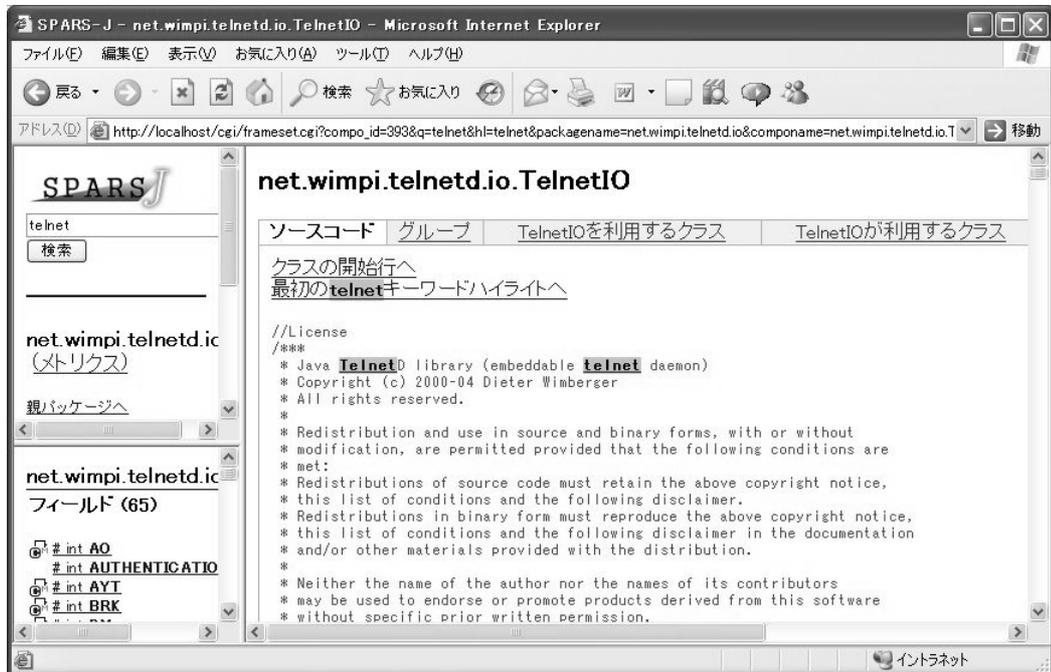


図 8: SPARS-J からのソフトウェア部品情報の表示

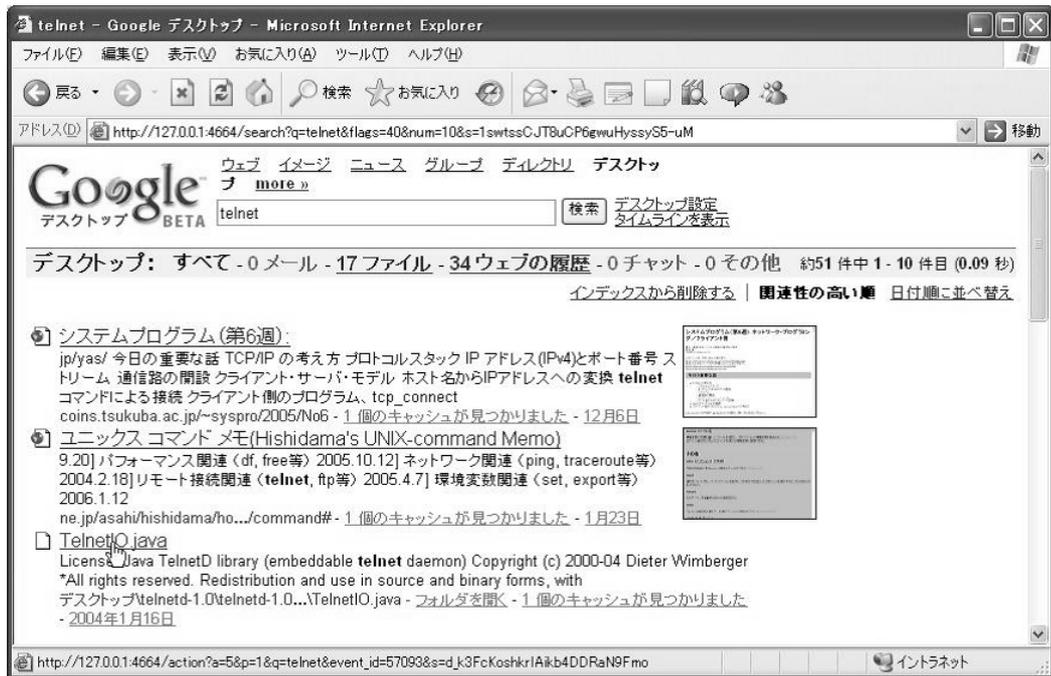


図 9: Google Desktop による検索

る(図9)。すると、この検索結果から、あらゆる種類のファイルを閲覧することができる。ユーザは、実際に telnet プロトコルが実装されているソフトウェア部品を再利用したいため、検索結果から Java ファイルを選択する(図9)。すると、規定のアプリケーションによって、Java ファイルのソースコードが表示される(図10)。

しかし、この状態ではソースコードそのものしか見ることができず、ユーザがこのソフトウェア部品を理解し、再利用することは困難である。そこで、この部品に関するより詳細な情報を得るために、SPARS-J を利用することとする。ユーザは SPARS-J の検索インタフェースに検索クエリ“telnet”を入力し、検索結果を得る(図7)。検索結果から、目的のソフトウェア部品を発見すると、その部品を選択し(図7)、詳細な部品情報を閲覧する(図8)。2.1 節で述べたように、SPARS-J は類似部品や部品の利用関係などの、ソフトウェア部品の理解に役立つ情報を提供するため、ユーザはソースコードからは得られなかった情報を十分に補うことができる。

導入時

ユーザは Java ソフトウェア部品を再利用するために、Java ソフトウェア部品の検索に特化したシステムである、SPARS-J を利用することを考える。しかし、前述のように SPARS-J では検索対象が Java ソースコードに限定されているため、様々なファイルから広く情報を

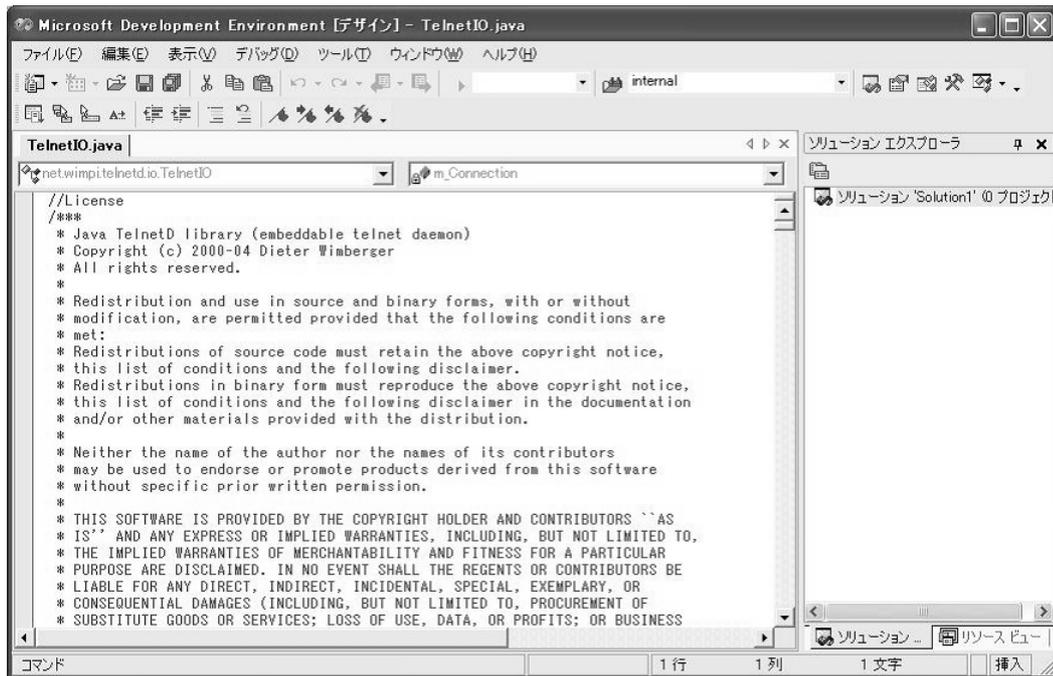


図 10: ソースコードの表示

集めることができない。そこで、ユーザは Google Desktop を用いて検索を行うことを考える。Google Desktop の検索インタフェースに検索クエリ “telnet” を入力すると、様々な種類のファイルが検索結果として得られる（図 9）。ユーザは、実際に telnet プロトコルが実装されているソフトウェア部品を再利用したいため、検索結果から Java ファイルを選択する（図 9）。すると、SPARS Desktop によってダイアログが表示される（図 11）。このダイアログ中の「はい」ボタンをクリックすることで、SPARS-J による表示を選択すると、この Java ファイルに関する詳細なソフトウェア部品情報が得られる（図 12）。

4.3 作業工程の比較と考察

作業工程の比較により、SPARS Desktop を利用することで、ソフトウェア部品の検索・再利用における作業コストが軽減されることが確認できる。

SPARS Desktop を利用することで、大きく分けて 2 つの問題点を解決することができる。まず 1 点目は、リポジトリの更新作業における問題点である。4.2.1 節の作業工程から分かるように、SPARS Desktop を導入しない場合には、Java ファイルが新規作成・更新される度に、ユーザがリポジトリへの登録作業を行う必要がある。個人の計算機中のソースコードは頻繁に更新されるため、この登録作業も頻発することになる。一方導入時では、新規作成・更新された Java ファイルに対して SPARS Desktop が自動的に登録処理を行うため、ユー

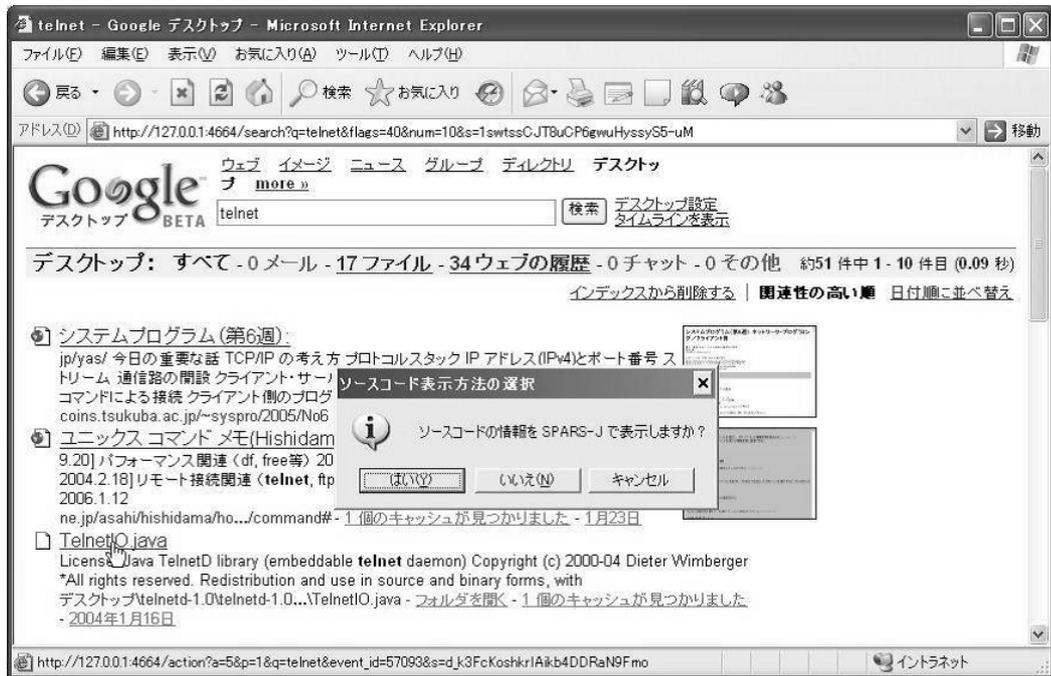


図 11: ダイアログの表示

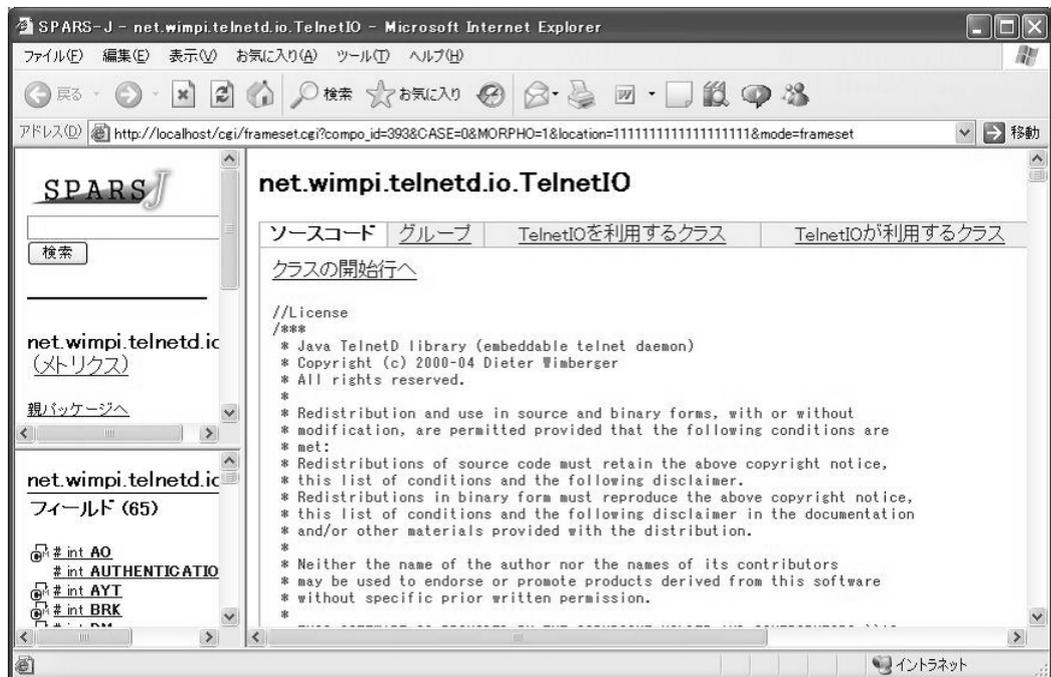


図 12: SPARS Desktop からのソフトウェア部品情報の表示

が登録処理を行う必要が無くなる。さらに、未導入時ではユーザが登録する Java ファイルを明示的に指定する必要があるために、検索候補とすべきファイルの登録漏れが生じる可能性もある。しかし導入時では、ローカルディスク内に保持する全ての Java ファイルをリポジトリに登録するため、このような登録漏れの問題も解消される。

2点目は、情報の検索における問題点である。4.2.2節の作業工程から分かるように、SPARS Desktop を導入しない場合には、SPARS-J での検索から Google Desktop での検索、あるいはその逆といったように、いずれの場合でも同様の検索を繰り返し行う必要がある。一方導入時では、一度検索クエリを入力すれば、その後は検索結果からユーザが望むファイルを選択することで、関連するあらゆるファイルの情報を得ることができる。

以上のような問題点を解決したことから、SPARS Desktop の導入によって、ソフトウェア部品の検索・再利用におけるユーザの作業コストを軽減できるといえる。

5 まとめと今後の課題

本研究では、デスクトップ検索ツール Google Desktop を利用し、Java ソフトウェア部品検索システム SPARS-J のためのリポジトリの自動更新機能を実現した。また、実装したシステムの導入前後のソフトウェア部品の検索・再利用にかかる作業工程を比較することで、実際に作業コストが軽減されることを確認した。

今後の課題としては以下のことが挙げられる。

- 削除されたファイルの処理

ファイルが削除された場合でも、そのファイルのインデックスや、SPARS-J におけるリポジトリの情報は残ったままである。これらの情報を削除するには、ユーザが削除処理を行わなければならない。リポジトリを自動で管理するためには、削除処理を自動化する必要がある。

- 順位付け手法の統合

現在、Google Desktop の検索インタフェースを利用した場合、検索結果の表示順序は日付順または関連性の高い順序により表示され、2.1.1 節で述べた、ソフトウェア部品間の利用関係に基づく順位付け手法である CR 法は適用されていない。Java ファイルに対して、CR 法を順位付け手法の 1 つとして統合することで、Google Desktop による検索において、より効率的な検索が可能になると考えられる。

謝辞

本研究において、常に適切な御指導および御助言を頂きました 大阪大学大学院情報科学研究科コンピュータサイエンス専攻 井上克郎教授 に深く感謝致します。

本研究において、常に適切な御指導および御助言を頂きました 同 松下誠助教授 に深く感謝致します。

本研究において、常に適切な御指導および御助言を頂きました 同 市井誠 氏 に深く感謝致します。

最後に、その他様々な御指導、御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 ソフトウェア工学講座 井上研究室の皆様 に深く感謝致します。

参考文献

- [1] B. Keepence and M. Mannion: “Using patterns to model variability in product families”, *IEEE Software*, Vol. 16, No. 4, pp. 102-108, (1999).
- [2] C. Braun: “Reuse, in John J. Marciniak, editor”, *Encyclopedia of Software Engineering*, Vol. 2, John Wiley & Sons, pp. 1055-1069, (1994).
- [3] Google Desktop Download: “<http://desktop.google.com/>”.
- [4] S. Isoda: “Experience report on software reuse project: its structure, activities, and statistical results”, *Proceedings of the 14th international conference on Software engineering (ICSE1992)*, pp. 320 - 326, Melbourne, Australia, (1992).
- [5] V. R. Basili, G. Caldiera, F. McGarry, R. Pajerski, G. Page and S. Waligora: “The software engineering laboratory - an operational software experience”, *Proceedings of the 14th international conference on Software engineering (ICSE1992)*, pp. 370-381, Melbourne, Australia, (1992).
- [6] Windows Desktop Search: “<http://desktop.msn.com/>”
- [7] Yahoo! Desktop Search: “<http://desktop.yahoo.com/>”
- [8] 北, 津田, 獅々堀: “情報検索アルゴリズム”, 共立出版, (2002).
- [9] 小堀, 山本, 松下, 井上: “類似度メトリクスを用いた Java ソースコード間類似度測定ツールの試作”, 電子情報通信学会技術研究報告, SS2003-2, Vol. 103, No. 102, pp. 7-12, (2003).
- [10] 横森, 梅森, 西, 山本, 松下, 楠本, 井上: “Java ソフトウェア部品検索システム SPARS-J”, 電子情報通信学会論文誌, D-I, Vol. J87-D-I, No. 12, pp. 1060-1068, (2004).
- [11] 横森, 藤原, 山本, 松下, 楠本, 井上: “利用実績に基づくソフトウェア部品重要度評価システム”, 電子情報通信学会論文誌, D-I, Vol. J86-D-I, No. 9, pp. 671-681, (2003),