OUKA
Osaka University Knowledge Archive

| | |
|---|---|
| Title | デスクトップ検索システムを用いたJavaソフトウェア部品リポジトリの自動更新機能の実現 |
| Author(s) | 市井, 誠; 木村, 和正; 松下, 誠 他 |
| Citation | 電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス. 2006, 106(201), p. 31-36 |
| Version Type | VoR |
| URL | https://hdl.handle.net/11094/26672 |
| rights | Copyright © 2006 IEICE |
| Note | |

*Osaka University Knowledge Archive : OUKA*

https://ir.library.osaka-u.ac.jp/

Osaka University

Institute of Electronics, Information, and Communication Engineers

社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SS2006-30 (2006-08)

# デスクトップ検索システムを用いたJavaソフトウェア部品リポジトリの自動更新機能の実現

市井 誠† 木村 和正†† 松下 誠† 井上 克郎†

† 大阪大学 大学院情報科学研究科
†† 大阪大学 基礎工学部
E-mail: †{m-itii,matusita,inoue}@ist.osaka-u.ac.jp, ††kazumasa@ics.es.osaka-u.ac.jp

**あらまし** ソフトウェア部品検索システムは過去に開発されたソフトウェア部品を検索および取得することによりその再利用や理解を支援するシステムであり，我々はJavaソフトウェア部品を対象としたシステムとしてSPARS-Jを提案している．SPARS-Jでは，高速かつ正確な検索の為にあらかじめ構築した部品リポジトリが利用される．部品リポジトリは検索対象の部品群の更新にあわせて更新される必要があり，開発者のクライアント計算機上の部品を検索する用途ではそのコストが問題となる．本研究では，この問題に対してデスクトップ検索システムを用いる．デスクトップ検索システムとはクライアント計算機上のファイルを索引付けして検索するシステムであり，計算機上に追加および更新されたファイルを自動的に検知して計算機の待機時間に登録することで，利用者が明示的な索引付けを行うことなくすべてのファイルの検索を可能としている．この機能を利用するシステムSPARS Desktopを構築し，SPARS-Jの部品リポジトリの更新の自動化を行う．また，適用事例によりその有効性を示す．

**キーワード** ソフトウェア部品検索，部品リポジトリ，自動更新，デスクトップ検索システム

# Automatic Updating System of Java Software Component Repository using Desktop Search System

Makoto ICHII†, Kazumasa KIMURA††, Makoto MATSUSHITA†, and Katsuro INOUE†

† Graduate School of Information Science and Technology, Osaka University
†† School of Engineering Science, Osaka University
E-mail: †{m-itii,matusita,inoue}@ist.osaka-u.ac.jp, ††kazumasa@ics.es.osaka-u.ac.jp

**Abstract** Component retrieval systems are proposed for effective software development and SPARS-J is one which targets Java components. SPARS-J builds and uses a component repository to search components quickly and precisely. Updating component repository costs much when SPARS-J is used on a client computer. In this paper, we developed SPARS Desktop, which updates component repository automatically using a desktop search system. A desktop search system is a search system for client computers, which builds index database automatically while the computer is in idle.

**Key words** Software Component Retrieval, Component Repository, Automatic Updating, Desktop Search System

## 1. はじめに

近年のソフトウェア開発現場では，大規模かつ高品質なソフトウェア製品を短期間で開発することが求められている．そのためにソフトウェア開発を支援するツールが必要とされており，そのひとつにソフトウェア部品検索システムがある．ソフトウェア部品検索システムとは，多数のソフトウェア部品を収集・蓄積することで部品リポジトリを構築し，キーワードなどの検索クエリによりリポジトリ中から部品を検索し，取得するシステムである．ソフトウェア部品検索システムを利用することで，ソフトウェア開発の効率化に有効であると言われている部品の再利用[1]やコード例の利用[2]を効率的に行うことができる．

我々はJavaを対象としたソフトウェア部品検索システムSPARS-Jの研究を行っている[3]．SPARS-Jは，類似や依存といったソフトウェア部品特有の性質を利用した効率的な検索を



NII-Electronic Library Service

Institute of Electronics, Information, and Communication Engineers

行う．SPARS-J では，検索に先駆けてファイルシステム上から対象となるソフトウェア部品を収集および解析し，部品リポジトリを構築することにより高速かつ効率的な検索を実現している．しかし，部品が新たに追加もしくは更新される度に部品リポジトリもまた更新される必要があり，運用にコストがかかる．SPARS-J が部品検索の為に設置されたサーバー計算機上で運用されているような場合には，部品の更新と部品リポジトリの更新を一括して行うことが可能であるため，更新にかかるコストは問題とならない．しかし，開発者個人が利用するクライアント計算機上にて利用される場合，無視できないコストが発生する．部品の更新が利用者の作業により発生し，部品リポジトリの更新の為に利用者の作業が妨げられるためである．

また，SPARS-J は Java 部品のみを対象とした検索システムであり，関連する情報を取得できない．例えば，WWW 上で配布されているライブラリ部品などはドキュメントが一緒に配布されていることが多いが，SPARS-J のみでは検索できない．そのため，一般の検索システムを併用する必要があり，同様の検索を繰り返す手間がかかる．

本研究では，これらの問題に対応するため，デスクトップ検索システムを利用する．デスクトップ検索システムとは，個人の計算機上のファイルを検索するシステムである．検索の高速化の為にあらかじめファイルの索引付けが行われてデータベースが構築されるが，索引付けは計算機の待機時間に行われる為に利用者の作業に影響を及ぼさない．また，様々な種類のファイルを検索可能であり，プラグインにより拡張可能である特徴をもつ．SPARS-J とデスクトップ検索システムをプラグインにより連携させるシステム SPARS Desktop を試作した．SPARS Desktop は，部品リポジトリの更新を計算機の待機時間に行うことで部品リポジトリの更新にかかるコストを削減し，また，デスクトップ検索システムの検索部から SPARS-J の表示部を利用することで関連情報の取得の問題を解決する．また，適用事例により，その有効性について確認する．

以降，2. 節でソフトウェア部品検索システムとその問題点について述べる．3. でデスクトップ検索システムについて述べ，4. 節で試作したシステム SPARS Desktop について述べる．5. 節で適用事例について述べ，最後に 6. 節でまとめと今後の課題について述べる．

## 2. ソフトウェア部品検索システム

### 2.1 概　　要

ソフトウェア部品検索システムとは，ソフトウェア部品を部品リポジトリに登録し，利用者の検索問い合わせに対して適合するソフトウェア部品を返すシステムである．ここでは，ソフトウェア部品（部品）は，クラスや関数，ソースコードやオブジェクトファイルなど，ソフトウェアの構成要素のことを指す．これまで，様々な形態のソフトウェア部品検索システムが提案されている．

ソフトウェア部品検索システムの代表的な利用目的としては，部品の再利用およびそのための理解や学習の支援が挙げられる．部品の再利用では，ライブラリ部品など，再利用するための部品を検索する．部品を再利用することで，新規開発やテストにかかるコストを削減できると言われている [1]．ソフトウェアまた，理解や学習では，コード例として利用できるソースコードをもつ部品を検索する．新しく利用するフレームワークの理解や学習には，実際の利用例としてのソースコードを参照することが役立つと言われている [2]．

これらの目的の為にはより多くの部品を正確かつ高速に検索する必要がある．そのため，多くのシステムではあらかじめ部品リポジトリを構築する．ソフトウェア部品を部品リポジトリに登録する方法は大きく分けて 2 通りある．

（1） 1 つ 1 つの部品を手作業により登録: 再利用を目的として設計・構築された部品を検索するシステムに多く，検索のためのメタデータなどと共に登録する

（2） 部品を自動的に収集して登録: ファイルシステム上を走査したり WWW 上を巡回することにより部品を発見して部品を登録する．ソースコード中の文字列などを基に部品の索引付けを行い，検索に用いる．

本研究では部品リポジトリの更新の自動化を目指しているが，(1) のタイプのシステムでは，検索の精度を向上する為に個々の部品の登録のコストを犠牲にしており，本質的に自動化が困難である．以降，ソフトウェア部品検索システムとしては (2) のタイプのみを考える．

### 2.2 システムへの要求

再利用もしくは理解や学習といった目的の為に，ソフトウェア部品検索システムには以下のことが要求される．

要求 1　必要な部品が部品リポジトリに含まれていること
世の中に存在するすべての部品を含む事が理想であるが，それは不可能であり，多くの場合はより限られた範囲の部品で十分である．例えば企業などの開発組織で運用する場合であれば，開発に利用できるライブラリ部品や，今までに開発したソフトウェアなどが含まれていれば良い．また，常に最新の部品に基づく情報で検索できることも必要である．

要求 2　部品リポジトリの更新にコストがかからないこと
登録処理やその手続きにより，利用者の本来の開発作業が妨げられてはならない．限りなく短い時間で更新処理を行うことが出来れば理想であるが，多少時間のかかる処理であっても，利用しない時間などで完了すれば問題ない．

要求 3　利用者が部品の検索条件を指定しやすいこと
検索条件として，利用者が必要とする部品に関して望む条件を的確に指定できることと，検索条件が複雑すぎず，容易に指定できることが同時に必要となる．

要求 4　検索結果が利用しやすいこと
検索結果が多数得られた場合に目的の部品に容易にたどり着けるように，適切に順位付けされていることが必要である．また，取得した部品を利用するときに必要な情報や他の部品を同時に取得できることが必要である．

### 2.3 SPARS-J

SPARS-J は Java のクラスを部品として，ソースコードに基づく全文検索を行うソフトウェア部品検索システムである．SPARS-J はライブラリ部品およびコード例としての部品の両



NII-Electronic Library Service

Institute of Electronics, Information, and Communication Engineers

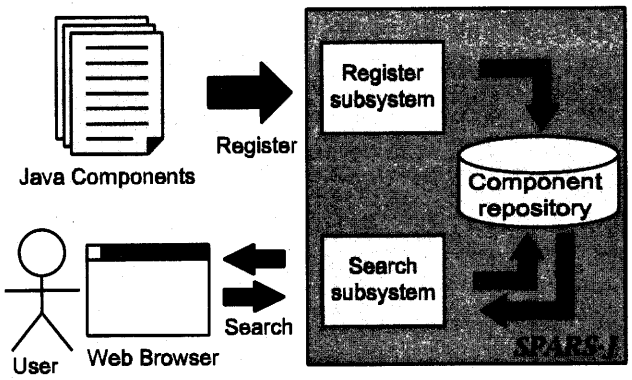


図 1　SPARS-J

方の検索および取得を目的としたシステムである．SPARS-J の概要を図 1 に示す．SPARS-J は大きく部品登録部 (Register subsystem)，部品検索部 (Search subsystem)，部品リポジトリ (Component repository) の 3 つに分かれる．SPARS-J では，部品登録部が，ファイルシステム上の指定したディレクトリ以下の Java 部品を部品リポジトリに登録する．利用者は，ウェブブラウザを用いて検索部にアクセスし，部品の検索および取得をおこなう．

SPARS-J は以下の 2 通りの利用形態が考えられる．

（1）　サーバー計算機上での利用

開発組織などに部品再利用の為のサーバーを設置し，そのサーバー上にて SPARS-J を運用する．部品リポジトリには，開発組織で利用可能な部品や，開発したソフトウェアを登録し，一元的に管理する．

（2）　クライアント計算機上での利用

開発者個人が利用する計算機上で SPARS-J を運用する．部品リポジトリには，過去に開発したソフトウェアや，利用する部品など，クライアント計算機上の部品を登録する．本研究ではこちらの利用形態に注目する．

#### 2.3.1　問　題　点

クライアント計算機で SPARS-J を利用する場合の問題点として以下の 2 点が挙げられる。

- 要求 1 と 2 を同時に満たせない，すなわち，利便性を落とさずに部品リポジトリを常に最新の状態に保つことができない．

クライアント計算機上で利用する場合，部品リポジトリに登録すべき部品はファイルシステム上に存在するすべての部品である．要求 1 を満たし，常に最新の状態で検索を行うためには，追加や変更が発生する度にファイルシステムを走査し，部品の登録処理を行う必要がある．しかし，登録処理により利用者の開発作業を妨げ，要求 2 を満たさなくなる．

これを避けるには，利用者がクライアント計算機を利用しない時間帯に更新処理を行えばよい．しかし，部品の追加や更新は利用者の開発作業により発生するため，部品リポジトリの更新の行われる時まで新しい部品を検索できない問題が発生し，要求 1 を満たさなくなる．また，クライアント計算機は利用しない時間帯は停止させる場合も多く，部品リポジトリを更新する時間を確保できない可能性もある．

- 要求 4 の一部を満たせない，具体的には，関連ドキュメントは別の手段で検索する必要がある

検索結果の順位付けに関しては，部品の再利用性尺度である Component Rank，キーワードと部品との適合度である Keyword Rank の 2 種類の手法で順位付けを行っている [3]．また，部品情報として，ソースコードの他に部品間の利用関係やパッケージ階層が表示され，効率的に部品を利用することができる．しかし，検索できるのは Java 部品のみであり，部品のドキュメントなどは検索できない．

なお，要求 3 に関しては問題無いと考えられる．SPARS-J は部品の検索手段としてキーワードによるソースコードの全文検索を提供している．検索時には必要に応じてキーワードの種類や Java パッケージの指定が可能であり，部品の的確な検索に必要な手段を備えている．また，キーワード検索はウェブ検索などにも用いられる一般的な手法であり，容易に利用できると考えられる．

### 2.4　その他のソフトウェア部品検索システム

本節では，SPARS-J 以外のソフトウェア部品検索システムについて述べる．

gonzui [4] は C, C++, Java, Ruby など，様々な種類の言語のソースコードの検索をおこなうシステムである．gonzui はファイルシステム上に存在する部品の他に，CVS や Subversion といったバージョン管理システムのリポジトリ上の部品を登録し，検索することができる．サーバー計算機上での管理の他，クライアント計算機上でも利用できるが，SPARS-J と同様，部品リポジトリの更新の問題や，関連ドキュメントが検索できない問題がある．

Koders [5] は様々なオープンソースソフトウェアから様々な言語の部品を収集し，ソースコードの検索を行うシステムであり，ウェブサイト上にて一般に公開されている．サーバー計算機上での管理のみを前提としていると考えられる．

JSearch [6] は Java を対象とした検索システムである．部品をコード例として利用することを目的としており，「“paint” という名前のメソッドを持つ」などの詳細な検索条件を指定することができる．部品リポジトリはサーバー計算機で管理され，利用者はクライアント計算機上の開発環境 (Eclipse) から部品の検索を行う．

Korders，JSearch は，いずれも部品リポジトリをサーバー上で管理する為に部品リポジトリの更新の問題は存在しないが，関連ドキュメントなどは別に検索する必要がある．

## 3.　デスクトップ検索システム

デスクトップ検索システムとは，クライアント計算機上のファイルを検索するシステムである．計算機上のファイルは検索の高速化の為にあらかじめ索引付けされ，データベースが構築される．デスクトップ検索システムとして代表的なものに Google Desktop [7], Windows Desktop Search [8], Yahoo! Desktop Search [9] がある．本研究では，他 2 つのデスクトップ検索システムに先駆けて開発された Google Desktop を用



NII-Electronic Library Service

Institute of Electronics, Information, and Communication Engineers

いる.

### 3.1 Google Desktop

Google Desktop は Windows 計算機上で動作する，デスクトップ検索システムである．Google Desktop の特徴としては以下の通りである．

- 更新および新たに追加されたファイルはクローラーと呼ばれるプログラムが自動的に検出し，計算機の待機時間を利用して索引付け処理が行う．このため，利用者の作業に影響を及ぼさず，常に最新の状態で検索が可能である．
- 様々な種類のファイルの索引付けを行い，キーワード検索の結果は様々な種類のファイルが混在して表示される．それぞれを選択するとファイルの種類に関連づけられたアプリケーションで表示される．検索結果の一覧は，更新時間やキーワードとの適合度に基づき順位付けされている．
- プラグインにより拡張可能であり，登録するファイルの種類の追加や，その登録処理や検索結果の表示のオーバーライドが可能である．
- 企業向けのバージョンでは，複数の計算機上の Google Desktop のデータベースを一元的に管理することが出来る．

## 4. SPARS Desktop

### 4.1 方　　針

本研究では，デスクトップ検索システムを用いた SPARS-J の拡張である SPARS Desktop を提案および構築する．2.3.1 節にて述べた，クライアント計算機上で SPARS-J を利用する時の問題点に対応するため，以下の機能を提供する．

（1） 利用者の利便性を落とさない形で，部品リポジトリを常に最新の状態に保つ

（2） 部品と同時に関連ドキュメントを検索する．

これらに対し，それぞれ以下の方針で対応する．

（1） Google Desktop の索引付けを利用して部品リポジトリの更新をおこなう．すなわち，Google Desktop のプラグインを作成することで，追加および変更された Java 部品を検出し，計算機の待機時間にそれらを SPARS-J に登録する．

ただし，プラグイン中で直接 SPARS-J への登録処理を行うと処理がブロックする可能性がある．そのため，ファイルパスを受け取り SPARS-J への登録を行うサーバーを作成し，プラグイン中での処理は更新されたファイルの情報をサーバーへ送信するのみとする．

（2） Google Desktop の検索画面を用いてファイル検索を行い，Java 部品と関連ファイルを同時に検索できるようにし，検索結果の Java 部品は SPARS-J を用いて表示する．

### 4.2 システムの概要

構築したシステム SPARS Desktop の概要を図 2 に示す．SPARS Desktop は登録部 (Register Subsystem)，表示部 (Display Subsystem)，登録サーバー部 (Register Server) の 3 つのサブシステムから成り立つ．登録部および表示部は Google Desktop のプラグインとして作成し，登録部は SPARS-J の部品リポジトリの更新を登録サーバー部を通じて行う．以下，それぞれについて説明する．

登録部（Google Desktop プラグイン） Google Desktop のファイル収集の対象に Java ファイルを追加する．Java ファイルの追加および更新をイベントとして Google Desktop から受け取り，そのファイルパスを登録サーバー部へ送信する．

表示部（Google Desktop プラグイン） Google Desktop の検索結果一覧画面において，Java ファイルが選択されたときの動作をオーバーライドし，利用者が Java ファイルを Windows 規定のアプリケーションで開くか，SPARS-J を用いて開くかを選択出来るようにする．

登録サーバー部 プラグインである登録部から Java 部品のファイルパスを受け取り，部品リポジトリの更新をおこなう．

### 4.3 処理の流れ

部品の登録および検索の処理の流れについて説明する．

#### 4.3.1 登　　録

クライアント計算機の利用者の作業によりファイルシステム上の Java ファイルが更新されると，Google Desktop のクローラーがそれを検知し，イベントとしてプラグインである SPARS Desktop の登録部に送る．登録部は受け取ったイベントからファイル情報を取得し，ファイルパスを登録サーバー部へ送信する．登録サーバー部は受信したファイルパスに基づき，Java 部品として部品リポジトリへ登録する．

#### 4.3.2 検　　索

利用者が Google Desktop の検索部に対して検索クエリを入力すると，Google Desktop は検索結果一覧を表示する．利用者は得られた結果から閲覧するファイルを選択する．利用者が選択したファイルが 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し，SPARS-J の部品表示部が選択されたファイルの部品情報を表示する．Java ファイルで無ければ，ファイルは規定のアプリケーションで開かれる．

## 5. 適 用 事 例

SPARS Desktop を利用することで Java ソフトウェア部品の検索に関する問題点が解決されていることを適用例により示す．適用事例では，部品検索のシナリオに関して利用者がとる手続きを確認する．

### 5.1 内　　容

適用事例は，利用者が Java 言語での Telnet プロトコルの実装についての情報を得るために Java ソフトウェア部品を入手し，ファイルシステム上に展開した状態から開始する．このときの部品としては，Java Telnet Daemon(注1) を用いる．

まず，利用者は展開した部品を部品リポジトリへと登録する．続いて，検索により Telnet に関係する情報および Java 部品を得る．

この時，SPARS Desktop を利用する場合と利用しない場合を比較していく．具体的には，以下の 2 環境での適用を比較しながら見ていく．

有り Windows XP 上に Google Desktop, SPARS-J, および SPARS Desktop を導入した環境

(注1)：http://telnetd.sourceforge.net/



NII-Electronic Library Service

Institute of Electronics, Information, and Communication Engineers

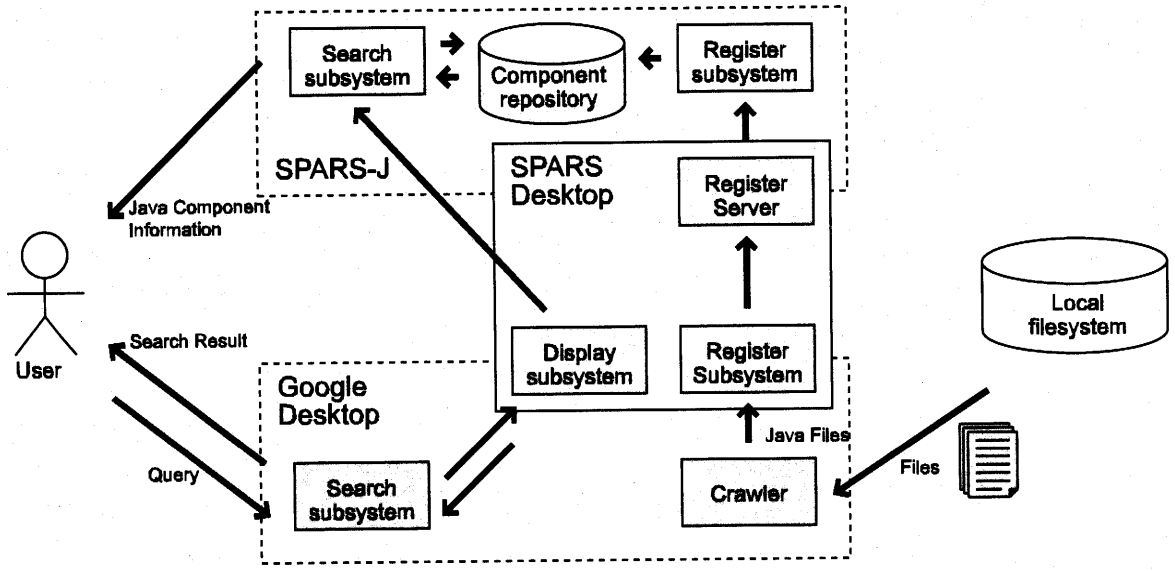


図 2　SPARS Desktop

無し　Windows XP 上に Google Desktop, SPARS-J, および Java ファイルを索引付けする Google Desktop プラグインを導入した環境.

### 5.2 結　　果

a) 部品リポジトリへの登録

有り　アーカイブを展開すると，Google Desktop のクローラーが十数秒程度で検知し，アーカイブ中の検索対象のファイルを索引データベースに対する登録を行った．そのうち Java 部品については，SPARS Desktop により，同時に部品リポジトリに対して登録された．利用者の作業は必要とされない．

無し　同様に Google Desktop のクローラーにより検知され，Java 部品は Google Desktop のデータベースにのみ登録された．利用者は SPARS-J の部品リポジトリへ登録するため，SPARS-J の登録部を用いて部品を展開したディレクトリを指定することで，部品の追加登録をおこなった．

b) 検　　索

有り　利用者は，Google Desktop の検索画面に対して検索語 “telnet” を入力することで検索結果を得た．検索結果には，Java ファイルやそのほかのファイル，ウェブページの履歴が同時に含まれる（図 3）．利用者は，その中から一般のファイルを選択して開くことにより，Telnet に関する情報を得ることができた．また，同じく一覧の中の Java 部品を選択すると，SPARS-J を用いて表示するかどうかの選択ダイアログが開き，利用者は SPARS-J を用いて表示することを選択する．結果として，部品のソースコードおよびパッケージ情報，利用関係にある部品を SPARS-J を用いて得ることが出来た（図 4）．

無し　利用者が，Google Desktop の検索画面に対して検索語 “telnet” を入力して検索結果を得て，一般のファイルを開くことで Telnet に関する情報を得るところまでは，SPARS Desktop の有る場合と同様である．しかし，Google Desktop の検索結果の Java ファイルを選択すると，Windows 上で規定のアプリケーションである統合開発環境によりそのファイルが開かれ，

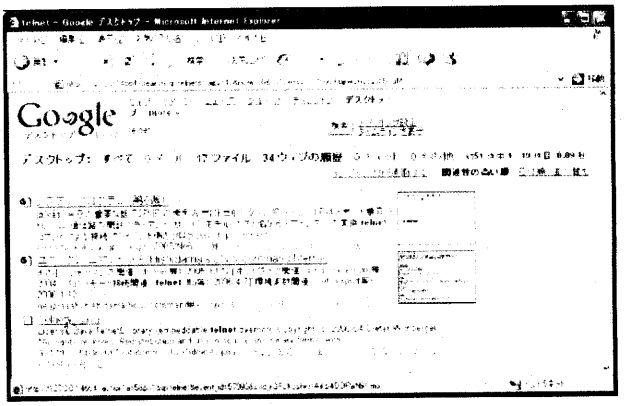


図 3　検索結果画面

ソースコード以外の情報，すなわちパッケージ情報や利用関係を得ることが出来なかった．そこで，利用者は SPARS-J の検索画面に対して同様に検索語 “telnet” を入力して得た検索結果から同じ部品を探し，部品情報を得ることができた．なお，SPARS-J は部品名やファイルパスにより，検索結果一覧を経由せず部品を得ることも可能であるが，本質的な手間は変わらない．

### 5.3 考　　察

SPARS Desktop を用いる場合と用いない場合を比較することで，SPARS Desktop によりソフトウェア部品の検索および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た．

まず部品リポジトリの更新の問題に関して，これまで Java ファイルが更新される度に必要であった部品リポジトリの更新作業が不要となる．クライアント計算機上のソースコードは利用者の作業により頻繁に更新されるため，手作業での更新作業は大きなコストとなる．また，SPARS Desktop を用いない場合には，部品リポジトリの更新の時には部品の存在する場所を正確に指定する必要がある．このときに指定に誤りがあれば登録漏れが生じてしまう．確かに登録時にファイルシステムすべ



NII-Electronic Library Service

Institute of Electronics, Information, and Communication Engineers

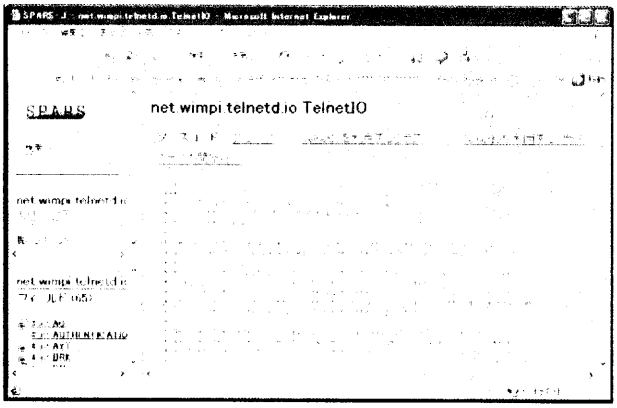


図 4　部品情報画面

てを指定すれば登録漏れは生じないが，ファイルシステムの走査は時間のかかる処理であり，作業に影響を及ぼしてしまう．これに対して，SPARS Desktop を用いる場合には更新されたファイルのみを検出して登録を行うため，時間のかかる処理を行うこと無く登録漏れを防ぐことができる．

また，関連情報の検索に関してもコストを削減出来ることがあきらかになった．これまで，Java 部品の検索と関連情報の検索で，同様の検索を複数回行う必要があったが，SPARS Desktop を用いることで 1 回の検索で両方を得ることができた．

このように，SPARS Desktop によりソフトウェア部品検索に関わるコストが大きく改善されたが，以下の様な問題が残されていると考えられる．まず，Google Desktop のクローラーは新規追加および更新されたファイルのみを検出し，削除されたファイルについては検出しない．そのため，削除されたファイルが検索されてしまう問題がある．何らかのタイミング，例えば検索によりヒットした時や定期的な処理によりファイルの有無を確認し，ファイルシステム上から削除されたファイルを部品リポジトリからも削除することが必要である．

また，関連情報の検索の問題に関して Google Desktop の検索画面を用いることで対応したが，Google Desktop の検索結果の順位付けでは SPARS-J の特徴である Component Rank が用いられず，キーワードとの適合性や更新時間のみが考慮されている．現在，Google Desktop では順位付けをプラグインにより操作することは不可能であるが，SPARS-J 側へ Google Desktop の検索結果を取り込むなどの方法により，順位付けの統合を行うことが望ましいと考えられる．

## 6. ま と め

本研究では，デスクトップ検索システム Google Desktop を用いて，Java ソフトウェア部品検索システム SPARS-J の部品リポジトリの自動更新機能を実現した．また，Google Desktop の検索結果から SPARS-J の部品表示部を利用するための実装をおこなった．

また，適用事例を示し，実装したシステムを用いることで，SPARS-J の部品リポジトリの更新時のコストが軽減され，また，部品情報と関連情報を同時に取得できることを確認した．

今後の課題としては，5.3 節で述べた，ファイルシステム上にて削除されたファイルを部品リポジトリからも削除すること，および，部品と関連情報の混在させた検索にて，部品の Component Rank を考慮した順位付けを実現することが挙げられる．

**謝辞**　本研究の一部は文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託に基づいて行われた．ここに記して謝意を表す．

### 文　献

[1] I. Jacobson, M. Griss and P. Jonsson: "Software Reuse" (1997).
[2] F. Shull, F. Lanubile and V. R. Basilli: "Investigating reading techniques for object-oriented framework learning", IEEE Transactions on Software Engineering, **26**, 11, pp. 1101–1118 (2000).
[3] K. Inoue, R. Yokomori, T. Yamamoto, M. Matsushita and S. Kusumoto: "Ranking significance of software components based on use relations", IEEE Transactions on Software Engineering, **31**, 3, pp. 213–225 (2005).
[4] "gonzui". http://gonzui.sourceforge.net/.
[5] "Koders". http://www.koders.com/.
[6] R. Sindhgatta: "Using an information retrieval system to retrieve source code samples", In Proc. of the 28th International Conference on Software Engineering (ICSE'06), pp. 905–908 (2006).
[7] "Google desktop". http://desktop.google.com/.
[8] "Windows desktop search". http://desktop.msn.com/.
[9] "Yahoo! desktop search". http://desktop.yahoo.com/.



NII-Electronic Library Service