論　文

Java ソフトウェア部品検索システム SPARS-J

横森　励士[†]　梅森　文彰[†]　西　秀雄[†]　山本　哲男[††]
松下　誠[†]　楠本　真二[†]　井上　克郎[†]

Java Software Component Retrieval System SPARS-J

Reishi YOKOMORI[†], Fumiaki UMEMORI[†], Hideo NISHI[†], Tetsuo YAMAMOTO[††], Makoto MATSUSHITA[†], Shinji KUSUMOTO[†], and Katsuro INOUE[†]

あらまし　大規模で複雑な大量のソフトウェアが開発され，様々な場所において様々な目的で利用されている．これらのソフトウェア資産の中には新たな開発作業において活用することができるアイデアや，少しの修正を加えるだけで開発に用いることが可能なソフトウェア部品が存在していると考えられる．現在は自然言語文書用に開発された全文検索システムを用いて開発者が望む情報や部品を適宜検索するという方法が主であるが，ソフトウェアは自然言語文書とは違い，依存や類似といった部品間の関係の利用なしでは有効な検索は難しい．
本論文では 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 の構築を行い，その有効性を評価した．SPARS-J は，依存や類似といったソフトウェア部品特有の特性を考慮しながら，大規模なライブラリの構築を自動的に行う．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，部品および関連する詳細情報を併せて提供する．有効性評価では，既存の全文検索システムとの比較や，企業における実際のソフトウェア開発現場への適用によりシステムの有効性を確認した．

キーワード　ソフトウェア部品，ソフトウェア検索，ソフトウェア再利用，Java，実験的評価

1. はじめに

近年，大規模で複雑な大量のソフトウェアが開発され，様々な場所において様々な目的で利用されている．プログラムソースの公開と再配布を許可しているオープンソースソフトウェア開発コミュニティは，ソフトウェアプロダクトが増加するにつれて開発基盤としての価値だけでなく，ソフトウェア資産を有する大規模なライブラリとしての価値も高まっている．例えば SourceForge [9] ではソースコードのストレージやコミュニケーションに必要なメーリングリストなどソフトウェア開発に必要とされる物的資源を提供しており，2004 年 2 月現在 75,000 以上ものプロジェクトを保持し，開発者として登録されているユーザ数は 78 万を越える．これらのソフトウェア資産の中には新たな開発作業において活用することができるアイデアや，少しの修正を加えるだけで開発に用いることが可能なソフトウェア部品が存在していると考えられる．

そこで，大規模なライブラリの可視化によるソフトウェア資産の有効活用が重要となる．有効活用の一例としてソフトウェア部品の再利用 [2] が挙げられる．一般にソフトウェア部品 (Software Component) は再利用できるように設計された部品とされ [4]，再利用の単位として用いられる．再利用は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ために有効な技術の一つとして知られているが，その効果を最大限に得るためには，部品を含むライブラリに関して十分な知識を再利用者が持たねばならない．しかしながら，多数の開発者が参加する開発プロジェクト内でライブラリに関する知識の共有を行うことは非常に困難でコストのかかる作業である．現在は Namazu [7] などの自然言語文書用に開発された全文検索システムを用いてライブラリを構成し，開発者が望む情報や部品を適宜検索するという方法が主である．

しかし，ソフトウェア部品においては，部品間のメ

[†] 大阪大学大学院情報科学研究科〒 560–8531 大阪府豊中市待兼山町 1–3
Graduate School of Information Science and Technology, Osaka University 1–3, Machikaneyama-cho, Toyonaka-shi, Osaka 560–8531, Japan
[††] 独立行政法人科学技術振興機構〒 332-0012 埼玉県川口市本町 4–1–8
Japan Science and Technology Agency 4–1–8, Honmachi, Kawaguchi-shi, Saitama 332–8531, Japan

ソッド呼び出しや継承などの利用関係や，コピーによる類似部品が多いなどの特性から，部品間の関係が非常に複雑になりやすい．そのため，ソフトウェア部品の検索を目的として，単に全文検索システムを利用するだけでは，適切な検索結果を得ることや，検索結果を参考にして効率的に情報を取得することは難しい．

本論文では 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*) を構築し，その有効性を評価した．SPARS-J は，依存や類似といったソフトウェア部品特有の特性を考慮しながら，大規模なライブラリの構築を自動的に行うシステムである．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．

さらに，SPARS-J についてソフトウェア部品検索システムとしての実験的評価を行う．実験ではまず，全文検索システム Namazu と SPARS-J との比較実験を行う．実際の利用方法を想定した検索を行い，上位で取得された部品の適合率をもとに評価する．次に，SPARS-J で用いられている各順位付け手法の特徴や，最適な順位付け手法を調査する．SPARS-J では，利用関係に基づく順位付け手法や，各部品における検索キーの出現頻度に基づく順位付け手法をもとに順位付けを行っており，上位で取得された部品の適合率や，ユーザの最適順位と実際に得られた順位の差を比較することで評価を行う．最後に，SPARS-J を企業における実際のソフトウェア開発現場に適用し，被験者に対して実施したアンケート結果に関する考察を行う．SPARS-J が検索システムとしてだけでなく，部品の管理や理解を支援するシステムとしても有効であることを確認する．

以降，2. 節で SPARS-J の概略を説明し，3. 節で構築したシステムについて説明する．4. 節では，各評価実験の概要および結果を紹介し，考察を行う．最後に，5. 節でまとめと今後の課題について述べる．

2. SPARS-J の概要

本節では Java ソフトウェア部品検索システム SPARS-J の概要と，そこで用いる手法について簡単に説明する．

2.1 SPARS-J の特徴

ソフトウェア部品検索システムとは，登録時に部品を用途や信頼度に応じて分類してライブラリとして蓄積し，検索時には検索者の指定に応じて部品を探し出す自動化システムを指す．

SPARS-J は Java の一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．再利用並びにプログラム理解，保守といった広範な用途を想定しているため，部品の詳細の把握や，目的に応じたカスタマイズをすることが可能なソースコードを部品とした．支援対象は，設計者やプログラマなど部品を検索し使用するソフトウェア開発者と，ライブラリの部品を分類し維持管理にも責任を持つライブラリ管理者である．

部品は登録時に索引付けされ，検索時には全部品の索引キーの集合から成り立つ索引をもとに部品の検索を行う．索引キーは Java ソースコード中に現れる予約語以外のキーワードとトークン種類の組である．また，索引キーに用いられるキーワードのことを特に索引語と呼ぶ．同様に，検索時に用いられるキーワードを検索語と呼び，検索語と検索対象のトークン種類の組を検索キーと呼ぶ．

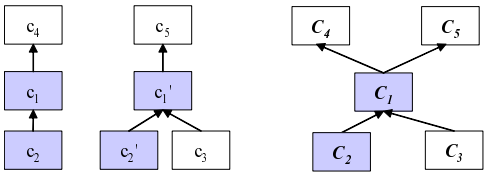
検索システムとしての形態は自然言語文書を対象とした全文検索システムと似ているが，ソフトウェア部品間の利用関係や類似関係を可視化する部品群グラフの構築や，利用関係に基づく順位付けにより，ソフトウェア部品検索に特化した機能を提供する．

2.2 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一つの機能を提供する．いま，ある部品がある部品を利用する時，この部品間に利用関係が存在すると言う．

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．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言う．

部品を頂点，利用関係を有向辺としたグラフを部品グラフと呼ぶ．さらに各部品群を頂点，その間の利用関係を有向辺としたグラフを部品群グラフと言う．類似部品のない部品は単一要素集合の部品群をなす．図 1(a) は部品グラフで，この場合，部品 c_2 は c_1 を，



(a) 部品間の関係　　(b) 部品群間の関係

図 1　部品グラフと部品群グラフ
Fig. 1　Clustered Component Graph

表 1　利用関係
Table 1　Use Relation

Relations
Class Inheritance(継承)
Implementation of Abstract Class(抽象クラス実装)
Implementation of Interface(インターフェース実装)
Declared Variable Type(変数宣言の型)
Instantiate(インスタンス生成)
Field Reference(フィールド参照)
Method Invocation(メソッド呼び出し)

c_2', c_3 は c_1' を，c_1 は c_4 を，c_1' は c_5 をそれぞれ利用していることを表している．また，部品 c_1 と c_1'，部品 c_2 と c_2' はそれぞれ類似した部品とする．c_3, c_4, c_5 には類似部品はない．図 1(b) は，この部品群グラフである．部品群に属する部品はそれぞれ
$C_1 = \{c_1, c_1'\}$，　$C_2 = \{c_2, c_2'\}$，　$C_3 = \{c_3\}$，
$C_4 = \{c_4\}$，　$C_5 = \{c_5\}$ である．

SPARS-J における利用関係

SPARS-J では表 1 に示す 7 種類の部品間の関係を利用関係と定めている．ライブラリに登録される部品について静的依存関係解析を行い利用関係を求める．

SPARS-J における類似部品の判定

SPARS-J における類似部品の判定では，部品のソースコードを複数のメトリクスで測定した値を用いて判定する．この類似部品の判定手法の詳細は [6] において紹介されている．

この手法ではまずサイクロマチック数などの複数のメトリクスによって測定された値を，部品の構造的複雑さを表す複雑性類似度メトリクスとする．次に，部品の表層的特徴の指標として，Java 全 96 種類のトークンそれぞれの種類別出現頻度を求め，構成トークン類似度メトリクスとする．これら複雑性類似メトリクスと構成トークン類似度メトリクスに関して経験的に閾値を定め，二つの部品間でそれぞれの差が閾値内の場合，類似部品と判定する．

2.3　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づけることが重要であるのと同様に，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と考える．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易いと考えられるからである．そのため，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．

KR 法

索引キー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．抽出された索引キーが部品の内容を表すうえ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*)，およ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 [5] 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．現在は表 2 のように，経験的にクラス定義名やメソッド定義名など部品の概念を表すものに対して大きな重みを設定している．本手法を *KR* **法** (*Keyword Rank* **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．

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．その総和を部品の KR 値とする．例として，検索語に “quick sort”，検索対象のトークン種類としてフィールド変数名，コメントが指定された AND 検索の場合，表 3 に示す索引を持つ部品の KR 値は次のようになる．ここで，ライブラリの総部品数は 1000，“quick” でヒットする部品数が 10，“sort” でヒットする部品数が 30 とする．

$$log_e(200{\cdot}1){\cdot}\tfrac{1000}{10} + log_e(200{\cdot}1 + 30{\cdot}8 + 10{\cdot}2){\cdot}\tfrac{1000}{30}$$

CR 法

我々はこれまでに，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 (*Component Rank* **法，***CR* **法**) を提案している [3], [11]．CR 法では，部品群グラフを表す隣接行列における固有値 1 の固有ベクトルを求め，ベクトルの各要素の値を CR 値，CR 値で

表 2　トークン種類とその重み
Table 2　Tokens and their Weights

Token	WT	Token	WT
クラス定義名	200	メソッド定義名	200
インタフェース名	50	パッケージ名	50
インポート名	30	呼出しメソッド名	10
参照フィールド変数名	10	生成したクラス名	10
変数の型名	10	参照する変数名	1
コメント (/* */)	30	文書コメント (/** */)	50
行末コメント (//)	10	文字列リテラル	1

表 3　部品の索引
Table 3　Indexed Words

Index	Token	Count
add	コメント	3
⋮	⋮	⋮
quick	メソッド定義名	1
sort	メソッド定義名	1
sort	コメント	8
sort	行末コメント	2
⋮	⋮	⋮
vector	クラス定義名	1

表 4　Borda の手法
Table 4　Borda's Method

	CR	KR	CR+KR
1	A	C	C
2	E	D	A
3	C	E	E (2nd)
4	D	A	D
5	B	B	B

表 5　評　価　点
Table 5　Evaluation Value

	CR	KR	CR+KR
A	1	4	5
B	5	5	10
C	3	1	4
D	4	2	6
E	2	3	5

整列した順位を CR と呼ぶ．CR 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 CR 値は高くなる．

順位の統合

CR 法，KR 法はそれぞれ別の観点から部品の順位付けを行っている．この順位付けを統合することで，それぞれが持つ検索アルゴリズムの特徴を考慮した結果が得られると考えられる．そこで Borda の手法 [1] を利用して CR 法，KR 法の順位の統合を行う．この手法では，各順位に対して評価点を割り当て，その合計値をもとに統合された順位を求める．表 4 は CR 法，KR 法，および統合された結果を，表 5 はそれに対応した各部品の評価点を表しており，表中の A，⋯，E はそれぞれ部品である．部品 A は CR で 1 位，KR で 4 位であり，評価点は 5 点となる．よって部品 A は順位を統合した結果 2 位となる．以下，この CR 法，KR 法の出す順位の統合によって順位を求める手法を *CR+KR* **法**と呼び，得られた順位を CR+KR と呼ぶ．

3. システム構成

本節では，構築したシステムについて説明する．システムは部品登録部と部品検索部という二つの処理部と，登録された情報を蓄積しているデータベースから構成される．構成図を図 2 に示す．

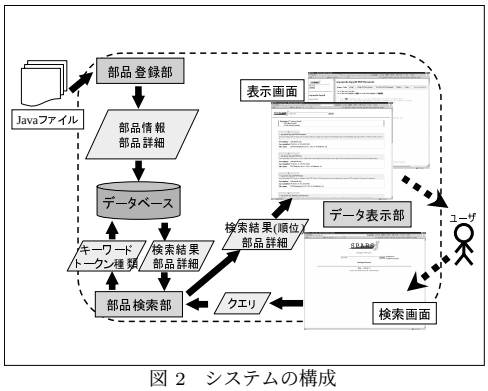


図 2　システムの構成
Fig. 2　Architecture of SPARS-J

3.1　システムの機能

本システムが持つ機能は以下の通りである．

- キーワード検索

日本語を検索語とすることも可能である．また，検索対象とするパッケージや，トークン種類の指定ができる．検索結果の順位付けは，CR，KR，および CR+KR のいずれかを検索者が選択可能である．

- 利用関係や類似部品の提示

部品間の利用関係や類似部品を提示することで，部品の使用方法に関する情報を取得できる．

- パッケージブラウザ

パッケージ階層構造をハイパーリンクで表現しており，パッケージ内に含まれる部品の一覧を取得できる．

- メソッド一覧表示

部品内で定義されているメソッドの一覧が表示される．メソッド名はメソッド定義行へのリンクになっており，辿ると定義行へジャンプできる．

- キーワードハイライト

ヒットしたキーワードがハイライトされて表示される．ハイライトをクリックすることで，次のハイライト部分へジャンプできる．

- アーカイブダウンロード

部品が含まれる jar ファイルなどのアーカイブをダウンロードして使用することができる．

3.2　システムの構成

データベース

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．検索時には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．

- ファイル情報データベース

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，更新時間などの情報が格納されている．

- 部品情報データベース

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．

- 利用関係データベース

部品間の利用関係と，その種類が格納されている．このデータベースには利用関係とは別に，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．

- 索引情報データベース

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，結果をデータベースに格納する．部品登録時の流れは次のようになる．

（1） ライブラリ管理者が Java ファイルを登録
（2） Java ファイルを解析して各種情報をデータベースに格納
（3） 部品群グラフ構築
（4） CR 法による順位付け

部品検索部

検索者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なう．得られた結果を KR 法で順位付けし，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 Web 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） データベースから部品を検索
（3） KR 法による順位付け
（4） 順位の統合

4. 評 価 実 験

4.1 実 験 目 的

一般的な検索システムとの順位付け性能の比較

この実験では，一般的な検索システムと比較して，SPARS-J が適合する部品をどれだけ十分に開発者に提供できるかを評価する．比較においては，ある目的に応じた部品の取得を想定した検索を行い，上位 10 件で取得された部品の適合率をもとに評価を行う．SPARS-J を用いることで要求するソフトウェア部品を迅速に検索でき，ソフトウェアの再利用に SPARS-J が有効であることを示す．

SPARS-J 内で利用される各順位付け手法の比較

2.3 節で示したとおり，SPARS-J は CR 法と KR 法，更にそれらを統合した CR+KR 法という 3 種類の順位付け手法を実現している．この実験では，適合率やユーザが最適だと思う順位と実際に得られた順位との差を求めることで，各順位付け手法の特徴や，最適な順位付け手法を調査する．

企業内のソフトウェアに対する SPARS-J の適用

SPARS-J は部品検索機能に加えて，データベース内における部品間の利用関係や類似部品等，部品に関する様々な情報を提供することができ，部品の管理や理解を支援するシステムとしても有効であると考えられる．この実験では，SPARS-J を実際のソフトウェア開発現場に適用し，実際の開発者・管理者に対してアンケートを実施することで，それらの情報が実際の開発現場ではどのように役立てられるのかを確認する．

4.2 実験における評価尺度

実験では，以下の評価尺度を用いる．

- 適合率

検索された部品中の適合部品数の割合を示す指標で，適合率が高いほど検索結果に不必要な部品を含んでおらず，よい検索結果を示しているといえる．

- ndpm 値 [10]

ndpm(*Normalized Distance-based Performance Measure*) 法とはユーザプリファレンスの概念を用いて，二つの順位付けの差を定量的に評価する手法である．ユーザプリファレンスとは，そのユーザの好みを明示的に表現したものである．ここでは部品におけるユー

ザプリファレンスを，部品を二つ一組にして比較した場合の,「どちらの部品がより有用である（，あるいは目的に適合する）か」を記述したものとする．

この時，D を部品の有限集合とすると，ユーザプリファレンスとは，$d, d' \in D$ について，D 上の二項関係 $\succ_u$，すなわちユーザの相対的な順序付けとなる．

npdm 法では，ユーザプリファレンスとシステムの順位付けの距離に基づいてシステムの性能を計測する．

D の各要素の全ての組み合わせに対して，ユーザプリファレンス ($\succ_u$) とシステムの順位付けにおける順位関係 ($\succ_s$) が異なる，すなわち $d \succ_u d'$ かつ $d' \succ_s d (d, d' \in D)$ となる d, d' の組数を求める．この組数を m とし，n を部品集合 D の要素数とすると，ndpm は以下の式で求められる．

$$ndpm(\succ_u, \succ_s) = \frac{m}{{}_nC_2}$$

値が小さいほど，ユーザプリファレンスとシステムの順位付けの差が少なく，システムの順位付けが妥当で，ユーザの理想的な順位付けに沿ったものであるといえる．[10] では，この手法が順序的尺度の測定手法として妥当であることを示している．

4.3 評価実験 1:一般的な検索システムとの順位付け性能の比較

本実験では，幅広い導入実績を持ち，信頼性の高い全文検索システムである Namazu [7] と SPARS-J との検索結果の比較実験を行った．

一般の検索において，検索者は最初に表示されるページに該当部品が見つからなかった場合，次ページを見るよりはクエリを変更して再検索する傾向がある．そこで本実験では，実際の利用方法を想定した検索を行い，上位 10 件の部品に対する適合率を求め，比較を行う．SPARS-J を用いることで迅速にソフトウェア部品を検索でき，ソフトウェアの再利用に SPARS-J が有効であることを確認する．

実験においては，JDK1.4 や，SourceForge.net [9] などのオープンソース開発コミュニティなどで公開されているソースコード，約 14 万ファイル (約 15 万クラス) から SPARS-J のデータベースを構築した．

なお，SPARS-J の検索結果は，類似部品を一つにまとめたグループ毎に順位付けを行っている．それに対応した処置として，Namazu の検索結果では，手作業で類似部品をまとめ一件として扱った．

上記の検索対象データベースに対して，表 6 に示した検索キーワードで検索した結果を評価した．各キーワードの検索結果に対する各検索システムの上位 10 件の適合率の計算結果を表 7 左に示す．

表 6　検索キーワード
Table 6　Query Keywords

	Keywords	Purpose	Hits(class)
K1	quicksort	Algorithm	47(89)
K2	binarysearch	Algorithm	23(46)
K3	clock applet	Tool	16(55)
K4	applet textarea	Example	186(344)
K5	random number generate	Tool	39(51)
K6	stack push pop	Algorithm	133(219)
K7	chat server client	Tool	61(124)
K8	classfile dump	Tool	35(57)
K9	zip deflate	Example	29(43)
K10	write read inputstream outputstream	Example	21(25)

適合率について，SPARS-J の各順位付けと Namazu とで，対応のある平均値の差の検定 [8] を有意水準 5%で行ったところ，有意な差が存在した．Namazu は SPARS-J の KR 法と同様に評価値に TF-IDF 法を用いている．しかしながら，上位 10 件の適合率に有意差が見られたのは，SPARS-J では構文解析を行い索引語の種類によって重み付けを行っているためであると考えられる．このことから SPARS-J は，ソフトウェア検索において，文書検索システムと比較して優れており，部品検索システムとして機能を十分に果たすと期待できる．

4.4 評価実験 2:SPARS-J 内で利用される各順位付け手法の比較

本実験では SPARS-J における順位付け手法である，CR 法，KR 法，CR+KR 法それぞれの比較を行う．SPARS-J では，順位付け手法にかかわらず検索結果に含まれる部品の集合は同じであるため，各部品の順位付けのみが変化する．

本実験では，評価実験 1 と同じ検索を行い，各順位付け手法における検索結果上位 10 件の適合率を求める．さらに，検索結果全体についてユーザにとっての最適な順位を作成し，ndpm 法を用いて SPARS-J の各順位付け手法で実際に得られた順位と最適な順位との比較を行い，順位の差を計算する．これらの指標をもとに，各順位付け手法の特徴や，最適な順位付け手法を調査する．検索対象となるデータベースおよび検索キーワードは，評価実験 1 と同じものを利用した．

SPARS-J システムの各順位付け手法に対して，表 6 に示された各キーワードの上位 10 件の検索結果に対する適合率の計算結果と，順位全体における ndpm 値の計算結果を表 7 に示す．表 6 中の HITS 欄は，

表 7　適合率・ndpm 値の評価結果
Table 7　The result of Precision and Ndpm

	precision				ndpm		
	CR	KR	CR+KR	Namazu	CR	KR	CR+KR
K1	1	1	1	0.9	0.036	0.048	0.037
K2	1	1	1	0.6	0.194	0.261	0.221
K3	0.5	0.5	0.5	0.4	0.133	0.117	0.092
K4	0.4	0.9	0.8	0.6	0.123	0.200	0.189
K5	0.4	0.4	0.4	0.3	0.208	0.192	0.194
K6	0.2	0.2	0.2	0.1	0.184	0.184	0.160
K7	0.9	1	1	0.4	0.081	0.103	0.080
K8	1	0.8	1	0.2	0.047	0.109	0.052
K9	0.6	0.7	0.7	0.4	0.190	0.305	0.257
K10	0.5	0.7	0.7	0.7	0.210	0.324	0.267
Ave	0.65	0.72	0.73	0.46	0.143	0.178	0.141

SPARS-J において検索できた件数を部品群単位で表している．またカッコ内は検出した総クラス数を表す．

適合率について対応のある平均値の差の検定を行ったところ，KR 法と CR+KR 法は，CR 法と比較して有意水準 5%で優れているという有意差が存在した．また，ndpm 値の比較についても同様に対応のある平均値の差の検定を行ったところ，CR 法と CR+KR 法は，KR 法と比較して有意水準 5%で優れているという有意差が存在した．

CR 法は，よく利用され中心的な役割を持つような部品が上位に順位付けされる．そのため，検索結果として得られる件数や部品数は表 6 で示したように違いはあるが，件数の多少にかかわらず，検索結果全体の順位はユーザの想定する順位付けに近い．しかし，クエリとの適合性は低い中心的な役割を持つ部品も上位に順位付けされやすいため，上位の適合率では劣ることがわかる．

一方，KR 法は基本的にキーワードがよく現れる部品を上位に配置するため，検索結果上位は適合部品である可能性が高い．しかし，キーワードが頻繁には出てこないがクエリとの適合性は高い部品も少なからず存在する．KR 法では，その部品を低く順位付けしてしまうため，ユーザの想定する順位付けとの差が出やすくなる．

また，CR+KR 法は CR 法と KR 法の両方の性質を受け継ぎ，CR 法と KR 法の両方で高順位に順位付けされた部品のみが上位に順位付けされることで，CR 法，KR 法単独の順位付けより性能が改善していると考えられる．

4.5　評価実験 3:企業内のソフトウェアに対する SPARS-J の適用

ここでは，企業の協力を得て SPARS-J を実際の開発現場において適用した結果を紹介する．企業内のソフトウェアの一部 (約 2500 クラス) を用いて SPARS-J のデータベースを構築し，ソフトウェア開発者・管理者合わせて 7 名の被験者に実際の業務におけるソースコード管理支援を目的として本システムを用いてもらう．2 週間の運用後，その 7 名に対して，表 8 のような本システムについての 11 項目のアンケートを行った．

表 8　アンケート内容とその回答
Table 8　The Results of Questionnaire

Respondent	A	B	C	D	E	F	G	Mode
パッケージブラウザ	5	4	4	5	5	3	3	5
類似部品一覧	5	2	4	5		4	3	5,4
被利用クラス一覧	5	5	5	5		5	5	5
利用クラス一覧	5	5	5	1		5	5	5
メトリクス一覧	4	2		1	1	4	5	4,1
ソースコード取得機能	3	5		1	5	2	5	5
検索結果一覧の見やすさ	5	3	4	4	5	3	5	5
ハイライト表示	5	5	3	5	5	5	5	5
時間的コストの削減	5	3		3	5	4	1	5,3
ソフトウェア品質の向上	3	3		5	3	4	1	3
ソフトウェアの把握	5	5	3	1	3	2	1	5,3,1

表 8 にその回答結果を合わせて示す．回答は 1 から 5 までの 5 段階評価であり，各機能の有効性，利用しやすさや，各被験者にとっての利用頻度をもとに，総合的に評価したものである．5 が一番良く，1 が一番悪い評価，また空白は無回答であったことを意味する．また，表の一番右の列はその項目において最も回答が多かった評価値 (最頻値) を表す．

表 8 のアンケート結果から，SPARS-J における「パッケージブラウザの利用」,「類似部品のグループ化」,「被利用クラスの参照」,「利用クラスの参照」,「検索結果一覧表示」,「ハイライト表示」は，実際の開発現場の開発者や管理者から高い評価を得ていることがわかる．また，具体的な感想として，SPARS-J を用いることで,「部品を利用しているソフトウェアの把握」ができ,「部品改訂時の影響範囲の調査」がしやすいという意見もあった．これらの意見とパッケージブラウザと利用クラスの参照，被利用クラスの参照という機能が高評価を得ていることを併せると，ソフトウェアシステムの全体像の把握に非常に適していると考えられる．加えて，CR 法による順位付けでは，ソフトウェアの中心となる部品が上位に順位付けされるため，利用関係をさらに有効活用できるという意見もあった．

5.　まとめと今後の課題

本論文では Java ソフトウェア部品検索システム

SPARS-J の構築を行い，実験的評価によってその有効性を確かめた．評価の結果，SPARS-J は全文検索システム Namazu と比較して検索結果上位 10 件の適合率に有意差が存在した．また，SPARS-J 内における各順位付け手法の比較では，検索結果上位 10 件の適合率や，検索結果全体においてのユーザの理想順位との差において，2 つの順位付け手法にそれぞれ特徴が見られ，それらを統合した順位付け手法は一番有効であることを確認した．さらに，アンケートの結果から，SPARS-J はソフトウェアの再利用を目的とした検索だけでなく，ソフトウェアシステムの保守などを目的とした，部品の管理や把握に非常に有効であることを確認した．

これらを総合すると，SPARS-J はソフトウェア部品検索システムとして高度な順位付けの性能を持ち，ソフトウェア開発時の部品再利用の促進・支援や保守時における部品の把握を支援するのに有効で，その結果，ソフトウェア品質の向上支援に利用でき，過去の資産の共有に役立つと考えられる．今後の課題として，他のソフトウェア部品検索システムとの比較，順位付け性能以外の定量的な評価，などが挙げられる．なお，現在 http://demo.spars.info で構築したシステムを公開中である．

謝辞 本研究は，一部，独立行政法人科学技術振興機構計算科学技術活用型特定研究開発推進事業 (ACT-JST)，日本学術振興会科学研究費補助金基盤研究（B）（課題番号：14380144）の支援を受けている．

文　献

[1] J. C. Borda: “M’emoire sur les ’elections au scrutin”, *Histoire de l’Acad’emie Royale des Sciences*, (1781).
[2] C. Braun: “Reuse, in John J. Marciniak, editor”, *Encyclopedia of Software Engineering*, Vol. 2, John Wiley & Sons, pp. 1055-1069, (1994).
[3] K. Inoue, R. Yokomori, H. Fujiwara, T. Yamamoto, M. Matsushita, S. Kusumoto: “Component Rank: Relative Significance Rank for Software Component Search”, Proceedings of the 25th International Conference on Software Engineering (ICSE2003), pp. 14-24, Portland, Oregon, U.S.A., (2003).
[4] I. Jacobson, M. Griss and P. Jonsson: “Software Reuse”, *Addison Wesley*, (1997).
[5] 北, 津田, 獅々堀: “情報検索アルゴリズム”, 共立出版, (2002).
[6] 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).
[7] “Namazu Project”, http://www.namazu.org .
[8] 芝, 渡部, 石塚 編: “統計用語辞典”, 新曜社, (1984).
[9] “SOURCEFORGE.net”, http://sourceforge.net/ .
[10] Y. Y. Yao: “Measureing Retrieval Effectiveness Based on User Preference of DocumentS”, Journal of the American Society for Information Science, Vol. 46, No. 2, pp. 133-145, (1995).
[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).

（平成 xx 年 xx 月 xx 日受付）

横森　励士　（正員）

平 11 阪大・基礎工・情報卒．平 15 同大大学院博士課程了．同年同大産学官連携研究員．博士 (工学)．プログラム構造解析の研究に従事．情報処理学会，IEEE 各会員．

梅森　文彰

平 14 阪大・基礎工・情報卒．現在同大大学院情報科学研究科博士前期課程在学中．プログラム構造解析の研究に従事．

西　　秀雄

平 14 阪大・基礎工・情報卒．現在同大大学院情報科学研究科博士前期課程在学中．プログラム構造解析の研究に従事．

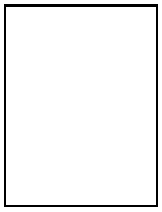
山本　哲男

平 9 阪大・基礎工・情報卒．平 14 同大大学院博士課程了．同年科学技術振興機構研究員．博士 (工学)．ソフトウェアプロセスの研究に従事．情報処理学会，IEEE 各会員．

松下　　誠

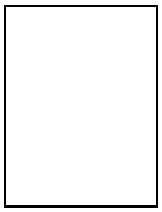
平 5 阪大・基礎工・情報卒．平 10 同大大学院博士課程了．同年同大・基礎工・情報・助手．平 14 阪大・情報・コンピュータサ

イエンス・助手．博士 (工学)．ソフトウェアプロセス，オープンソースソフトウェア開発の研究に従事．



楠本　真二　（正員）

昭 63 阪大・基礎工・情報卒．平 3 同大学院博士課程中退．同年同大・基礎工・情報・助手．平 8 同学科講師．平 11 同学科助教授．平 14 阪大・情報・コンピュータサイエンス・助教授．博士 (工学)．ソフトウェアの生産性や品質の定量的評価，プロジェクト管理に関する研究に従事．情報処理学会，IEEE，IFPUG 各会員．



井上　克郎　（正員）

昭 54 阪大・基礎工・情報卒．昭 59 同大大学院博士課程了．同年同大・基礎工・情報・助手．昭 59〜昭 61 ハワイ大マノア校・情報工学科・助教授．平 1 阪大・基礎工・情報・講師．平 3 同学科・助教授．平 7 同学科・教授．平 14 阪大・情報・コンピュータサイエンス・教授．博士 (工学) ソフトウェア工学の研究に従事．情報処理学会，日本ソフトウェア科学会，IEEE，ACM 各会員．