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

信学技報
TECHNICAL REPORT OF IEICE.

ソフトウェア部品推薦のための協調フィルタリング手法の提案と実現

市井　誠†　山本　哲男††　横森　励士†　井上　克郎†

† 大阪大学 大学院情報科学研究科
†† 立命館大学 情報理工学部 情報システム学科
E-mail: †{m-itii,yokomori,inoue}@ist.osaka-u.ac.jp, ††tetsuo@cs.ritsumei.ac.jp

あらまし　ソフトウェア部品検索システムは，ソフトウェア開発における再利用および理解の支援に有効なシステムで，我々の研究チームでは，Java を対象としたソフトウェア部品開発システム SPARS-J を提案している．本研究では，ユーザの検索効率の向上を目的として，協調フィルタリングを利用したソフトウェア部品の推薦手法を提案する．提案手法では，ユーザの検索履歴を取得し，蓄積されたユーザの検索履歴に対して協調フィルタリングを適用することで，ソフトウェア部品の推薦を行う．また，提案手法を用いて SPARS-J 上に推薦システムを実現し，そのシステムへの適用実験の結果により，推薦機能の利用により検索効率が向上する事を確認した．
キーワード　ソフトウェア部品検索，協調フィルタリング，推薦システム

Collaborative filtering method for software component recommendation

Makoto ICHII†, Tetsuo YAMAMOTO††, Reishi YOKOMORI†, and Katsuro INOUE†

† Graduate School of Information Science and Technology, Osaka University
†† Department of Computer Science, College of Information Science and Engineering, Ritsumeikan University
E-mail: †{m-itii,yokomori,inoue}@ist.osaka-u.ac.jp, ††tetsuo@cs.ritsumei.ac.jp

Abstract　A software component search system is a support tool for reuse and understanding in software development. We have proposed SPARS-J, which is a software component search system for Java. In this paper, we propose a method for recommendation of software component for the purpose of improvement of users' search efficiency. In the proposed method, a system collects users' search histories and recommends components using the histoies based on collaborative filtering. We implemented a recommendation system on SPARS-J based on the proposed method, and evaluated it. Experimental result shows that our recommendation system helps improvement of users' search efficiency.
Key words　Software Component Search, Collaborative Filtering, Recommender System

1. ま え が き

近年，ソフトウェアの開発現場においてソフトウェア部品の再利用の必要性が高まってきている．その支援ツールとしてソフトウェア部品検索システムがあり，我々の研究チームでは Java のソースコードを対象としたシステム SPARS-J [5] を提案している．SPARS-J では，索引語の抽出やパッケージ階層，利用関係の解析などの，ソースコードの静的な解析結果を基にしてユーザに検索手段を提供する．ユーザは，これらの解析から得られた情報を利用して検索を行い効率的にソフトウェア部品を取得することができる．しかし，実際の検索効率はユーザによって大きく異なる事や，同様の目的の検索がすでに他のユーザによってなされている場合も考えられる．この場合，ユーザに対して過去のユーザの検索行動をフィードバックすることで，より効率的に検索や部品の取得ができると考えられる．

そこで本論文では，ユーザの検索履歴に対して協調フィルタリングの手法を適用し，ユーザに必要と思われる部品を推薦する手法を提案する．提案手法では，まずユーザからアイテムに対する評価を取得して蓄積し，その評価の傾向からユーザの目的を推測する．その後，推薦対象のユーザと目的が類似しているユーザが高く評価したアイテムを推薦する．我々は，一般的な協調フィルタリングでおいて利用される「好み」という概念を，検索システムにおける適用においては「検索目的」と置き換えることで，有効な推薦を行うことができると考える．

さらに，本提案手法を SPARS-J 上で実現し，適用実験を行うことで，部品検索システム上においても協調フィルタリング手法が有効であるかどうかを検証する．

以下，2. 節で協調フィルタリングに関する研究について説

明する．次に，3. 節で提案する手法の詳細を説明し，4. 節で SPARS-J に対する実装について説明する．さらに，5. 節では実装したシステム上での適用実験の内容を説明し，その実験結果について考察する．最後に，6. 節でまとめと今後の課題について説明する．

2. 協調フィルタリング

協調フィルタリング *(Collaborative Filtering)* とは，そのシステムにおける各ユーザの嗜好をアイテムに対する評価という形で記録し，そのユーザと似たような評価をしているユーザの嗜好情報をもとに，ユーザの嗜好を推測するシステムである．この協調フィルタリングは，リコメンデーションサービスを提供する際に使用される代表的な手法で，大量のアイテムの中から，個々のユーザの好みに合うアイテムを推薦する際の要素技術として研究が進んでいる．ここでいうアイテムとは，記事，映画，音楽，商品などの，推薦の候補となるものを指す．本研究の場合は，ソフトウェア部品となる．

協調フィルタリングと相対する概念として，**コンテンツベースフィルタリング** *(Content-based Filtering)* [3] がある．これはアイテム自体の性質，例えば含まれる文字列や，あらかじめ入力された属性などを基にユーザにアイテムを提示する手法である．コンテンツベースフィルタリングの例として，検索システムにおけるキーワード検索や，ユーザが閲覧した文書から特徴的な単語を抽出し，それをもとに提示する手法などが挙げられる．SPARS-J における利用関係の表示なども，コンテンツベースフィルタリングの一例である．

これに対し，協調フィルタリングでは他のユーザの評価を基にユーザにアイテムを提示する．初期の協調フィルタリングを使用したシステムとしては，E-mail や Netnews 等の推薦システムである，Tapestory [1] がある．Tapestory の場合，ユーザによって明示的に選択された推薦者が評価したアイテムを掲示するという，推薦を得る相手を明示的に指定する手法をとっている．

これに対して，GroupLens [7] は推薦者や推薦する情報をシステムが自動的に決定する．GroupLens の場合，ユーザは Usenet の記事に対する評価を 5 段階で明示的に入力する．GroupLens はその情報をもとに，そのユーザと評価の傾向が似ているユーザを自動的に抽出し，ユーザに推薦を行う．現在，協調フィルタリングとして広く認識されている推薦メカニズムはこの手法を基にしており，GroupLens の手法は協調フィルタリングの基礎となっている．本研究でも，この手法を基にした手法を提案する．

さらに，アイテムへの評価をユーザに明示的に入力させず，システムが暗黙的に取得するシステムもある．たとえば Phoaks [4] では，Usenet の記事中から，URL の付属した投稿を収集し，それらをその URL に対する投票と見なしている．また，大杉 [6] のシステムでは，ソフトウェア機能の使用履歴を取得して暗黙的な投票とし，ユーザに対してソフトウェア機能を自動的に推薦する．

明示的な投票を利用するシステムでは，ユーザの評価が当人の嗜好を正確に表すため，精度の高い推薦を行うことができる利点がある．しかし，ユーザに投票の操作を要求する必要があり，ユーザが非協力的な場合は，評価の絶対数を減らしてしまう可能性もある．これに対して暗黙的な投票を利用するシステムでは，ユーザに特別な操作を求めることなく評価を取得することができるが，逆に正確なユーザの嗜好を把握することが難しいという欠点がある．

本研究の場合，検索効率の向上が目的であるため，推薦機能のためにユーザに参照した部品の適合性を投票させるといった余分な操作をさせることは好ましくない．このため，検索履歴をソフトウェア部品に対する暗黙的な投票とみなし，自動的に推薦を行う手法を採用する．

3. 提 案 手 法

本節では，蓄積されたユーザの検索履歴に対して協調フィルタリングを適用することで，ソフトウェア部品の推薦を行う手法を提案する．提案する推薦手法は，GroupLens で提案されたユーザ間の相関を基にした手法をもとに，ソフトウェア部品の推薦に適用するための拡張を加えたものである．提案手法の大まかな流れは以下の通りである．

a） ユーザの評価の取得

ユーザの部品参照履歴を取得し，ユーザの評価としてデータベースに記録する．

b） 相関係数の算出

データベースに記録された評価を基にして，ユーザ間の相関係数を求める．

c） 推薦値の算出

上記のユーザーの評価および相関係数を用いて，部品に対する推薦値を求める．

d） 推薦の実行

求めた推薦値を基に部品それぞれを推薦するかどうかを決定し，ユーザに対して部品の推薦を行う．

以降，これらの各段階について詳しく説明する．

3.1 a) ユーザの評価の取得

協調フィルタリングでは，ユーザの評価を利用してアイテムの推薦を行う．そのためにユーザから各アイテムに対する評価を取得する必要があるが，ユーザに評価・投票の労力を求めることは検索効率の向上を目的とするには好ましく無いと考えられる．そのため，提案手法ではユーザが部品を参照した時に暗黙的に投票したとみなし，その部品に対して評価値 1 で投票したとする．

ユーザ間の相関を基に協調フィルタリングによる推薦を行う場合，ユーザの評価傾向は変化しないことが前提となる．この前提は，GroupLens などのように評価傾向がユーザの好みを反映する場合は自然であり問題は無いと考えられるが，検索システムにおいてはこの前提は成立しない．なぜなら，ある部品について，その部品がある目的には適合しているが，他の目的には適合していない場合が存在しうることは容易に想像でき，検索目的によってユーザの評価傾向は容易に変化しうるからである．

検索システムにおいては，評価傾向が変化しないのは，ユーザ単位での評価傾向ではなく，検索目的を単位とした評価傾向である．そのため，一般的な協調フィルタリングにおける「ユーザ」を「検索目的」とみなして，検索目的間の類似度から，部品の推薦に利用するのが妥当であると考えられる．これより，検索目的が変化した段階で別のユーザとして扱う必要があるが，システムがユーザの検索行動から暗黙的に目的の変化を取得することは一般には難しい．本研究では，検索システムの利用終了でユーザの目的が変わったとみなし，システムの使用開始から終了までの評価を 1 ユーザ（検索目的）の評価として扱う．

ここで注意すべき点として，検索対象の部品群の中には，コピーして利用された部品など，類似した部品・同一の部品が複数存在することがある．これらの類似部品中からどの部品を参照するかということはユーザにとって意味を持たないことが多い．しかし，協調フィルタリングを適用する場合，本来同じ部品に対する投票が別々の部品に対する投票として扱われてしまうため，ユーザ間の相関の計算に悪影響を及ぼし，正しい推薦を行うことができない場合が考えられる．さらに，ユーザが部品を参照した時に，同時にその類似部品群中の部品全てに対して投票を行ったと見なしてしまうと，投票した部品と似ている部品の数（同一部品群に含まれる部品の数）によって，投票の重みが変わってしまい公平でなくなるという問題点がある．そこで，本提案手法では，類似部品群を推薦アイテムの単位として扱う．

3.2 b) 相関係数の算出

協調フィルタリング手法においては，推薦を受けるユーザと似たような行動を取っているユーザを推定するために，ユーザ間の類似度を相関係数として計算する必要がある．

GroupLens においては，2 ユーザ間の相関係数を求める際には，2 ユーザが共通して投票したアイテムに対する評価を用いている．一般にこの方法はうまく働くが，本研究のように評価が 0，1 の 2 値である場合には意味のある値とならない．また，本研究はユーザを区切って扱うため，1 ユーザあたりの投票数が少なくなる傾向にあり，推薦の精度が落ちるという問題もある [2]．そこで，ユーザ間の相関係数を求める手法として，Breese [2] の提案する拡張を行う．これは，2 ユーザのいずれかが評価したアイテム，およびどちらも評価していないアイテムを基に相関係数を求める手法である．このとき未評価の部品に対する値を設定する必要があるが，本手法では未評価の部品の評価値を 0 として計算を行う．

ユーザ a とユーザ i 間の相関係数 $c(a,i)$ を求める数式は以下のようになる．式中の I_i はユーザ i が評価した部品，I はユーザ a，i のいずれかが評価した部品および 2 ユーザのいずれも評価していない定数個の部品の集合を表す．

$$v_{i,j} = \begin{cases} 1 & \text{if } j \in I_i \\ 0 & \text{if } j \notin I_i \end{cases}, \ \bar{v}_i = \frac{1}{|I|}\sum_{j \in I} v_{i,j}$$

$$c(a,i) = \frac{\sum_{j \in I}(v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_{j \in I}(v_{a,j} - \bar{v}_a)^2 \sum_{j \in I}(v_{i,j} - \bar{v}_i)^2}}$$

3.3 c) 推薦値の算出

一般的な協調フィルタリングにおいては，推薦を受けるユーザと似たような行動を取っているユーザの評価をもとに，各アイテムに対して，そのアイテムの推薦値を計算する．実際には，ある未評価の部品に対する推薦値は，対象ユーザがその部品に対して投票するであろう評価値の推定値として求めることができる．各部品の推定値は，それぞれのユーザとの相関係数を重みとした，他のユーザのその部品への評価値の加重平均により表される．

提案手法においては，各部品の推薦値を求める際の計算量を考慮し，あらかじめ関連のあるユーザの集合を抽出する．この集合に含まれるユーザは，対象ユーザと参照した部品が 1 つ以上重複し，かつ，相関係数が負でなく，かつ，投票数が 2 以上のユーザとしている．また，相関係数をそのまま重みとして利用せず，相関の高いユーザにはより高い重みを，相関の低いユーザにはより低い重みを与えるよう，累乗して重みとして利用することで，より有効な推薦が行えることが知られている [2]．本手法でもそれを採用し，相関係数の 2 乗を重みとして利用する．

$U = \{i | I_a \cap I_i \neq \phi \wedge c(a,i) > 0 \wedge |I_i| > 1\}$ とした時，ユーザ a の部品 k に対する推薦値 $p_{a,k}$ は，以下の式により求められる．

$$p_{a,k} = \frac{\sum_{i \in U} c(a,i)^2 v_{i,k}}{\sum_{i \in U} |c(a,i)^2|}$$

3.4 d) 推薦の実行

前節で説明した式により，データベース中の部品に対して推薦値を求め，ユーザに対して推薦値の高い順に部品を推薦する．ただし，ユーザが既に参照した部品に関しては，ユーザはその部品については既知であると見なし，推薦対象から除外する．

4. 提案手法の実現 -SPARS-J への実装

我々は，前節で提案した協調フィルタリングを利用した部品推薦手法を，Java 部品検索システム SPARS-J に対して推薦機能を実装する形で，実現した．以下では，SPARS-J のアーキテクチャを説明した後に，追加した推薦部について説明する．

4.1 SPARS-J のシステム構成

SPARS-J のシステム構成を図 1 に示す．図中の各項目の意味は以下の通りである．

ライブラリ　検索対象となる Java ソースファイルのライブラリの集合である．検索結果から部品の詳細を表示する際には，このライブラリ中の該当部品が存在するファイルを参照して表示する．

データベース　ライブラリ中の Java ソースファイルを解析した結果としてのファイル情報や部品情報，利用関係および索引情報がデータベース中に格納される．解析時や検索時に部品の関連情報を提供するために利用される．

部品登録部　ライブラリ中のファイルを基にデータベースを構築する．まずライブラリのファイルを解析し，解析結果をデータベースに格納する．続いて検索結果のランキングに必要な評価値を計算し，その評価値をもとにソートを行う．

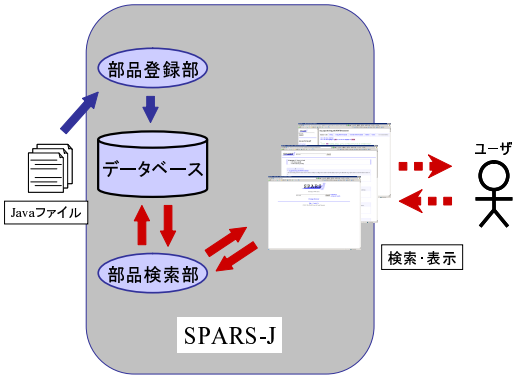

図 1　SPARS-J の構成

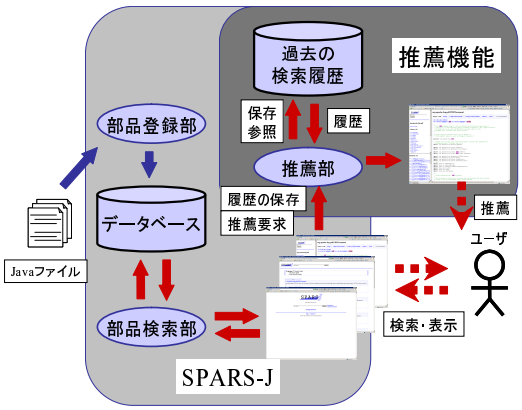

図 2　推薦部の構成

部品検索部　ユーザの入力した検索キーに該当する部品を検索してユーザに提示する．このとき，検索結果と併せてソースコードや利用関係，メソッド一覧などの解析情報を部品の詳細情報として提供する．

4.2　推薦機能の実装

我々は，提案手法に基づいて SPARS-J に対して履歴取得・推薦を行う機構を組み込むことで，SPARS-J 上での推薦機能を実現した．図 2 は推薦部を組み込んだ場合の SPARS-J の構成である．以下，推薦機能の詳細について，個別に説明する．

4.2.1　履歴の記録機能

SPARS-J では，ユーザは Web ブラウザを利用して部品の検索，表示を行う．3.1 節において考察したとおり，推薦機能では類似部品群を推薦アイテムの単位として扱う．すなわち，投票は類似部品群に対してなされたと見なし，類似部品群を推薦する．SPARS-J では個々の部品にはコンポーネント ID として一意な ID が振られているが，これとは別に類似部品群にも同様にグループ ID が振られている．推薦部の実装においてはこのグループ ID を利用して類似部品群の識別に利用する．

また，システムの使用開始から終了までを 1 ユーザとして扱うと説明したが，手法の実現においてはセッション Cookie を用いて識別する．すなわち，セッションをユーザに対応させ，同一セッションからの部品参照要求を，同一ユーザの評価とみなす．セッションには一意となる ID（セッション ID）を与えて以降の処理に利用する．

以上より，ユーザがブラウザにて部品のソースコードを表示した際に，システムは，ユーザがその部品群に対して評価値 1 で投票したと見なし，データベースに記録する．記録する内容は，セッション ID，コンポーネント ID，グループ ID，削除フラグである．なお削除フラグについては，次節で説明する削除機能の実装で利用する．

4.2.2　履歴の閲覧機能

ユーザがセッション内で参照した部品の部品名（クラス名）を一覧表示する．それぞれの部品名には，その部品詳細表示画面へ遷移するリンクがつけられている．また，それぞれの部品名の隣に「削除」リンクを付加し，ユーザが自分が利用しているセッション内での履歴を削除することができる．削除された部品は削除フラグを true として履歴内で保存され，そのセッション内では推薦されない．この機能により，そのセッションにおける推薦の精度を向上させることができるとともに，他のユーザがこのセッションを利用して推薦を受けるときにも推薦の精度の向上が期待できる．

4.2.3　推薦部品の表示機能

提案手法により計算された推薦値をもとに，システムが作成した推薦部品の一覧を表示する．推薦部品一覧の作成時においては，ある閾値以下の部品についてはその部品を利用しても満足の行く結果が得られないと判断し，一覧からあらかじめ除く．

実際の利用を考えた場合，このシステムで推薦を受けるためには，そのセッション内でいくつか部品を閲覧することが必要となる．このことと，部品の閲覧後の推薦画面への移行のしやすさを考慮して，推薦部品が表示される画面へは，検索結果における部品の詳細表示画面から遷移できるようにしている．

推薦画面では，図 3，図 4 のように推薦される部品の一覧が表示され，それぞれの部品名には部品詳細画面へ遷移するリンクが付加される．また，各部品に対する推薦値も併記し，ユーザが推薦部品の中から実際に参照する部品を選択する時の参考にできるようにもしている．

また，推薦システムにおいては，ユーザが目的に応じて使い分けることができるように，以下の 2 通りの推薦方法を用意している．これらの推薦部品を履歴の閲覧画面と同一の画面に表示することで，ユーザは自然に推薦機能を利用することができる．また，同一類似部品群の部品については初期状態では全て表示せずに 1 部品のみを表示し，必要に応じて全ての部品を表示する．

（1）　全推薦部品を表示する (図 3).
推薦部品の一覧を，推薦値の順でソートして表示する．この推薦方法は，必要な部品を効率的に閲覧することを目的としており，同時に利用するであろうライブラリなどの把握に適している．

（2）　利用関係にある推薦部品を表示する (図 4).
推薦前に閲覧していた部品と利用関係にある部品だけを対象として，推薦値順にソートした部品一覧を表示する．この場合，ユーザは大量の部品の中からでも，有用な利用関係に基づいて効率的に閲覧することができ，対象システムの全体像の理解に有効であると考えられる．

表 1　実 験 結 果

		GP1					GP2				
		A1	A2	A3	A4	平均	A5	A6	A7	A8	平均
検索時間 (分)	P1	28	47	14	50	34.8	**4**	**3**	**28**	**15**	**12.5**
	P2	25	19	2	28	18.5	**2**	**4**	**5**	**2**	**3.2**
	P3	**14**	**60**	**3**	**9**	**21.5**	9	18	44	26	24.2
	P4	**3**	**38**	**7**	**12**	**15**	23	45	13	23	26
適合率	P1	0.42	0.53	0.23	0.26	0.36	**1**	**1**	**0.78**	**0.75**	**0.89**
	P2	0.06	0.38	1	0.15	0.18	**1**	**1**	**0.4**	**1**	**0.73**
	P3	**1**	**0.5**	**1**	**1**	**0.88**	1	1	0.55	0.37	0.73
	P4	**0.67**	**0.64**	**0.83**	**1**	**0.79**	0.55	0.63	0.73	0.73	0.66

図 3　推薦機能の画面 (全推薦部品)

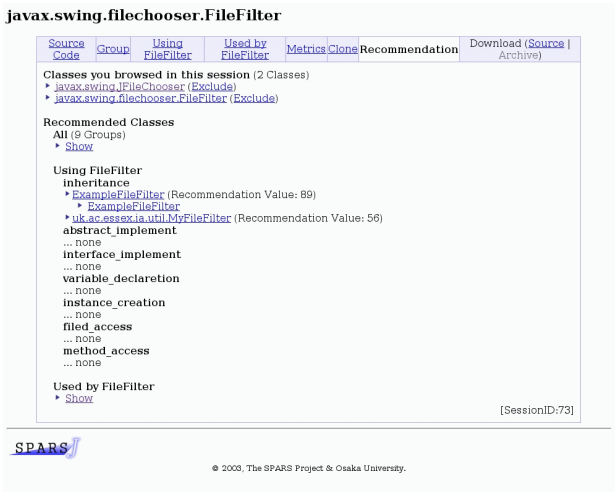

図 4　推薦機能の画面 (利用関係にある推薦部品)

5. 適 用 実 験

本節では，4. 節で実現したシステムに対して適用実験を行い，提案手法の有効性を評価する．実験においては，ソフトウェア部品検索システムの使用時に協調フィルタリングによる推薦機能を利用することで，検索効率が向上することを確認する．

5.1 実 験 概 要

実験においては，Java のスケルトンコードの未実装部分の実装を課題として，被験者が

（1） 用意した資料および SPARS-J のみ（推薦機能なし）を利用した場合

（2） 用意した資料および推薦機能を組み込んだ SPARS-J を利用した場合

の 2 通りの場合を比較し，検索効率が向上することを確認する．

以下，詳細について簡単に説明する．

5.1.1 課題プログラム

課題は，Java のスケルトンコードの形で，練習課題 1 課題を含む 5 課題を用意した．それぞれ課題となる処理以外のコードは記述済みであり，該当部分を記述・コンパイルするだけで実行が可能で，動作確認を容易に行うことができる．また，各課題の内容はそれぞれ独立しており，実装内容が重複したり，ある課題プログラムが他の課題の答えやヒントを含むようなことは無い．

5.1.2 被　験　者

実験は，大阪大学大学院情報科学研究科の学生および Ph.D の 8 名に対して行った．それぞれの被験者は Java の知識を持っている．また各被験者は，実験の前にはあらかじめ SPARS-J の利用方法に関する 1 時間の講習を受けており，SPARS-J を用いての検索を十分に利用できる状態になっている．

5.1.3 環　　境

被験者が課題を行う時の SPARS-J のデータベースは，JDK1.4 のソースコードとデモコード，および WWW 上から収集した Java ソースコードから構築したもので，総クラス数は約 35,000 クラスである．このデータベースには課題の作成に必要な情報は全て含まれており，SPARS-J を用いてコードを検索することで，全ての課題プログラムを作成できる．また，部品検索履歴のデータベースについては，事前情報は与えず空の状態から実験を行う．

5.1.4 評 価 項 目

評価に利用する項目として，以下を計測した．

検索時間　被験者が，課題開始から終了までに消費した作業時間の内，コードを記述していた時間を除いた時間とする．全体の時間ではなく検索時間で比較することで，個人のプログラミング能力の影響を減らすことができると考えられ，この時間が短ければ検索効率が良いといえる．

適合率　ここでは，適合率を（スケルトンコードの実装に利用したと見なせる部品／検索中に参照した部品の個数）とする．この値が 1 に近いほど検索効率が良いといえる．

5.1.5 手　　順

（1） まず全被験者が推薦機能を利用せずに練習課題 P0 を

行う．これは SPARS-J の利用および課題に十分に慣れ，後の Exp1，Exp2 間で慣れによる差違が発生しない様にするためである． また，この時にも各評価項目を測定し，能力が均一になるように被験者を 2 グループ GP1，GP2 に分けた．

（2） 推薦機能を利用せずに，GP1 の被験者が課題 P1，P2 を，GP2 の被験者が課題 P 3，P4 を行う．また，終了後にそれぞれの被験者のプログラムに対して，客観的に適合していない部品についての参照履歴を削除する．これは，履歴の削除機能を利用したことに相当する．

（3） 推薦機能を利用して，GP1 の被験者が課題 P3，P4 を，GP2 の被験者が課題 P1，P2 を行う． なお，この段階の進行によってもユーザの履歴は蓄積されるが，推薦内容に影響を及ぼさないように，フェイズ（3）以降で取得された検索履歴は推薦に利用しない．

5.2 結　　果

実験結果は表 1 の様になった．表中の A1,A2,...,A8 はそれぞれ各被験者を表し，推薦機能を利用した時の結果は太字で示されている．

5.3 考　　察

実験結果より，各課題に関して，推薦機能を利用したグループの方が平均で検索時間・適合率ともに優れていることがわかる．

課題 P3 については推薦機能の有無による差異が小さい．これは，課題 P3 を推薦機能を利用せずに行ったグループ G2 において，被験者 A5，A6 のみが該当分野に関する知識を持っていたことが原因であると考えられる．知識があることで，部品を推薦機能がなくても，検索結果や利用関係の一覧から有用な部品を容易に判別し，利用することができたと推測できる．その他の課題，特に P1，では差が大きく出ていることを考えると，推薦機能は，検索目的がユーザにとって未知の分野である場合に，検索効率の向上に大きく役立つといえる．

また，各被験者毎に見た時には，多くの被験者で推薦機能を利用した場合，検索時間・適合率の向上を確認できた．しかし，A2，A7 では逆に悪くなっている．この 2 人の被験者の参照履歴を解析したところ,「検索開始からの序盤で多くの部品を参照し，後半は序盤で見た部品を再び参照することが多かった」という特徴を確認することができた．つまり，最初にできるだけ多くの部品に目を通して，後で見直しながら絞り込んでいくという方法をとっていた．この場合，推薦される筈の部品が序盤でほとんど履歴に入ってしまい，利用できる部品がほとんど推薦されなかったと考えられる．

このことを考慮すると，参照済みの部品は推薦しないという方針を見直し，参照済みの部品も推薦対象とするかどうか，利用者に選択させる方式にする等の修正が必要であり，これは今後の課題である．

6. ま と め

本研究では，協調フィルタリングを利用したソフトウェア部品推薦手法を提案し，Java ソフトウェア検索システム SPARS-J に対して実装を行った．また，実現したシステム上での適用実験を行い，協調フィルタリングによる推薦を利用することで実際に検索効率が向上することを確認した．

今後の課題は，履歴に対する重み付けなどによる精度の向上や，検索からよりシームレスに推薦を得る為のユーザーインターフェースの改善，および，より大規模な実験があげられる．

謝辞 本研究は，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) の支援を受けている．

文　　献

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry: “Using Collaborative Filtering to Weave an Information Tapestry.”, *Communications of the ACM*, Vol.35, No.12, pp.61-70, 1992.

[2] J. S. Breese, D. Heckerman and C. Kadie: “Empirical Analysis of Predictive Algorithms for Collaborative Filtering”, In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp.43-52, 1998.

[3] 北，津田，獅子掘: “情報検索アルゴリズム”, 共立出版, 2002.

[4] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter: “PHOAKS: A System for Sharing Recommendations”, *Communications of the ACM*, Vol.40, No.3, pp.59-62, 1997.

[5] 西, 梅森, 横森, 山本, 松下, 楠本, 井上: ”Java ソフトウェア部品 解析・検索システム SPARS-J の構築”, 電子情報通信学会技術研究報告, SS2003-23, Vol.103, No.481, pp.43-48, 2003

[6] 大杉, 門田, 松本, 森崎: “ソフトウェア機能の推薦の為の協調フィルタリング”, ソフトウェアシンポジウム 2002 論文集, pp.83-89,2002.

[7] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl: “GroupLens: An Open Architecture for Collaborative Filtering of Netnews”, In *Proceedings of the 1994 CSCW*, pp. 175-186, 1994.