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.
SS 2000-27 (2000-11)

ソフトウェアコンポーネント利用情報の収集と共有

古賀 健太郎† 飯田 元‡ 松本 健一† 井上 克郎†¶

† 奈良先端科学技術大学院大学 情報科学研究科

〒630-0101 奈良県生駒市高山町 8916-5

TEL：(+81)743-72-5312 FAX：(+81)743-72-5319

‡ 奈良先端科学技術大学院大学 情報科学センター

¶ 大阪大学 大学院基礎工学研究科

E-mail:{kenta-ko, iida, matumoto}@is.aist-nara.ac.jp, inoue@ics.es.osaka-u.ac.jp

あらまし 本研究では，コンポーネントを用いたソフトウェア開発において，その開発事例である「利用情報」を収集し検索対象とすることによって，従来の仕様書のみの検索からでは困難であった開発者の具体的なニーズに沿ったコンポーネントの検索を可能とするシステムを提案する．利用情報とは，開発ツールや完成したソフトウェアから収集し蓄積されたコンポーネントの接続関係・動作環境等のことである．また，コンポーネント利用者が利用情報を参照することによって，有用な開発情報の獲得も可能にする．本稿では，提案システムの各機能を説明し，コンポーネントを用いたソフトウェア開発支援に対する有用性を考察する．

キーワード コンポーネント検索，ソフトウェア再利用，開発知識の共有，コンポーネントウェアハウス

Software Component Retrieval Assistance based on Utilization Profile

Kentaro Koga† Hajimu Iida‡ Ken-ichi Matsumoto† Katsuro Inoue†¶

† Graduate School of Information Science, Nara Institute of Science and Technology

8916-5 Takayama, Ikoma, Nara 630-0101, Japan

TEL：(+81)743-72-5312 FAX：(+81)743-72-5319

‡ Information Technology Center, Nara Institute of Science and Technology

¶ Graduate School of Engineering Science, Osaka University

E-mail:{kenta-ko, iida, matumoto}@is.aist-nara.ac.jp, inoue@ics.es.osaka-u.ac.jp

Abstract This paper proposes the system which enables the retrieval of the component suitable for developer's concrete requirement by collecting other developers' development cases as *Utilization Profile*. *Utilization Profile* is collected from software developent environment or derived from software itself. It contains information about connections between component and executable environment of components. Providing *Utilization Profile* enables the developer to aquire useful development infomation . This paper describes features of this system, and discusses the capability for component-based software development support .

key words component retrieval, software reuse, sharing of development knowledge, component warehouse

1 はじめに

1.1 背景

近年，巨大で複雑化してきたソフトウェアシステムへの要求に対応するために，ソフトウェアの高生産性，高信頼性，及び開発期間の短縮が要求されるようになってきた．それらの要求を解決する目的として，ソフトウェアの再利用を用いた開発形態が以前よりも増して注目されている．ソフトウェアを再利用する一つの方法として，既存の再利用可能なソフトウェアコンポーネント (以下コンポーネント) を用いて新たなシステムを開発する方法 (以下コンポーネント指向ソフトウェア開発) が存在する [2].

ここで言うコンポーネント技術とは，JavaBeans，ActiveX/DCOM等に代表される，ソフトウェアの機能の一部をバイナリ形式で保存し，後の開発でシステムに組み込んで直ちに利用可能にしたものであり，それらを実現する技術を総称してコンポーネントウェアと呼ぶ [1].

コンポーネント指向ソフトウェア開発の利点は，以下の3点が主に挙げられている [10].

- 新たな設計実装をするための時間を削減できる
- 新規のソフトウェア開発よりその信頼性が高い
- ソフトウェアの開発コストを削減できる

1.2 コンポーネントの利用形態

従来のコンポーネントの流通利用形態の典型的な例を図1に示す．まずコンポーネントは，コンポーネントウェアハウスと呼ばれるリポジトリに保存される．コンポーネントの開発者は，自ら開発したコンポーネントとその仕様書や利用環境などを記した「カタログ」をコンポーネントウェアハウスに随時登録していく．

コンポーネント指向ソフトウェア開発では，要求仕様書において定義されたシステムの機能 (またはその一部) が既存のコンポーネントを再利用して実現される．そのため，

- 実現すべき機能あるいはその一部を持つコンポーネントが存在するかどうか
- コンポーネントの機能を利用するための具体的なインターフェース

を知る必要がある．従ってコンポーネントウェアハウスを検索するシステムが重要な役割を持つ．コンポーネントの検索には，必要機能を抽象化したキーワードやそれらを組合せた論理式を入力とする検索システムを用いるのが一般的である．検索システムは，検索文に基づいてコンポーネントウェアハウス内を検索し，その結果を利用者に提示する．利用者は，その中から必要なコンポーネントを取得し，現在開発中のシステムに組み込んで利用する．

2 コンポーネント検索支援の必要性

2.1 コンポーネントの検索

Yeらの分類によると，利用者がコンポーネントを選択して利用する際には次の3通りの方法を用いている [12].

- 記憶による利用
- 思い出しによる利用
- 予測による利用

記憶による利用では，利用者は必要とするコンポーネントの存在を記憶しており，かつどれがそのコンポーネントであるのかを理解している．この場合，利用者は問題無く，直接そのコンポーネントを利用することができる．

思い出しによる利用では，利用者は必要としているコンポーネントの存在を記憶しているが，どのコンポーネントであったかを理解していない．したがって，利用者は検索することによって実際のコンポーネントの存在を確認する．

予測による利用では，利用者は必要としているコンポーネントの存在を記憶していないが，「もしかしたら存在しているかもしれない」という予測に基づいて検索する．しかし，その予測が常に的中するとは限らない．

図2はこれらの3つの利用に対応した利用者のコンポーネントウェアハウスに関する知識レベルを表現している．長方形 R はコンポーネントウェアハウスに存在している実際のコンポーネントの存在空間を表している．L1に存在するコンポーネントは記憶による利用に対応し，L2に存在するコンポーネントは思い出しによる利用と対応している．L3に存在するコンポーネントは予測による利用と対応しており，利用者にとって未知のコンポーネントはL4に存在する．L3とL4の領域は新たなコンポーネントの登録に伴って増加してゆき，積極的にコンポーネントのリリース情報を収集するか，検索ツールを用いることでL2やL1へと移行する．

コンポーネント指向開発の効率を上げるには，存在するコンポーネントを記憶することによってL1の領域を限りなく長方形 R に近づけることが望まれるが，全てのコンポーネントを記憶するには大きなコストを伴う．現実には，$(R-L1)$ の領域に含まれるコンポーネントの発見を検索ツールで支援することが必要である．

コンポーネントの検索については，過去に様々な研究がなされている．例えば，仕様書の記述内容を対象として検索を行う方法 [5][9] や，カタログ内に検索に便利な情報，キーワードなど記述しておいて検索に利用する方法 [7] が提案されている．また，ソースコードを対象とする検索方法 [6][12] も研究されているが，本研究の扱うコンポーネントはバイナリで流通し，ソースコードが得られない場合が多いため，ここでは考慮しない．

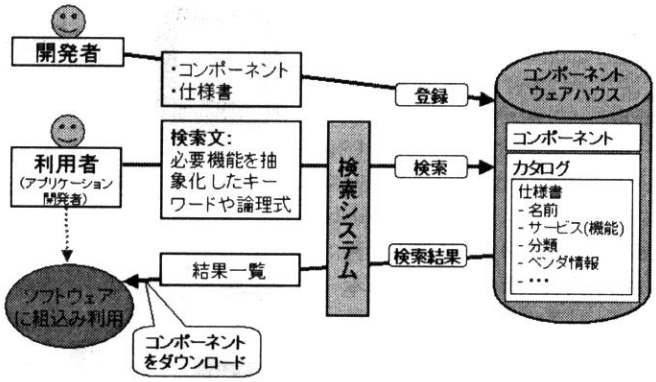

図 1: コンポーネントの利用形態

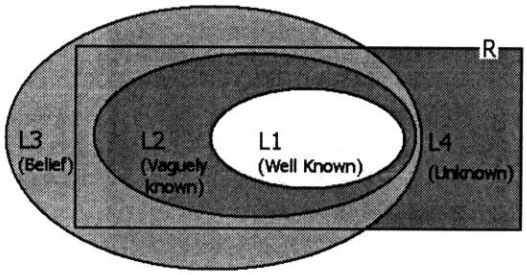

図 2: 既存コンポーネントに関する利用者の知識 [12]

2.2 コンポーネント検索における課題

従来のコンポーネントの検索・利用方法では，以下のような問題がある．

(1) コンポーネントが増加するに従い，必要コンポーネントの選択が困難になる

既存システムから汎用的な機能を切り出してコンポーネント化すると，各コンポーネントの規模が小さくなり，その数が増大する傾向がある [11]．また，コンポーネントウェアハウスに蓄積されたコンポーネントの数は，コンポーネントの新規開発や追加や購入により，時間と共に増加する．

コンポーネントウェアハウス内の部品が増大すると，検索システムによって類似コンポーネントが複数発見されるようになり，利用者はどのコンポーネントを自分の要求に合うものとして選択するべきか，という判断が困難になる．カタログの情報を利用して，より詳細な検索を行うことも考えられるが，カタログ情報を予め記述して登録する手間が必要な上，適切なコンポーネントを発見するために多数のカタログ項目に対して条件を指定するには，カタログ項目に関する知識の習得が必要となる．

(2) コンポーネントの組み合わせ方に関する有用な情報が不足している

コンポーネントは，自ら持つ機能を呼出すための「メソッド」を複数備えている．コンポーネント指向ソフトウェア開発では，複数のコンポーネントをシステムに組み込み，それらのもつメソッドを組み合わせることで，システムに要求される機能を実現する．

コンポーネントに含まれる機能の数は，コンポーネントの粒度の大きさによって変化する．GUI ボタンのように粒度の小さいコンポーネントは単一の機能を持ち，ワープロ機能を提供するような粒度の大きなコンポーネントでは多数のメソッドが提供される．

このように，様々な粒度のコンポーネントが混在するコンポーネントウェアハウスから適切なコンポーネントを検索して実際のシステムに組み込んで利用する場合，取得したコンポーネントに含まれる各メソッドを，開発中のシステムのどのメソッドから呼出して利用するべきかを決定しなければならないが，そのために参考となる情報としては，カタログに記述されたごく少数のサンプルコードしか存在しない．

このように，多様な粒度を持つコンポーネントの具体的な利用方法を多数用意して適切に参照できることが望まれる．

2.3 利用情報による検索支援

そこで本研究では，コンポーネントの検索を効率化し，再利用を促進するための手段として，他のユーザによる過去のコンポーネントの利用履歴や他のコンポーネントとの接続事例などを「利用情報」として収集し，開発者間で共有することを考える．利用情報とは，コンポーネントの仕様からだけでは解りにくい具体的な利用事例を

収集したものである．この利用情報を用いて，より具体的なニーズに沿ったコンポーネントの検索を支援するシステムを提案する．なお，本研究では社内のシステム開発部署のように比較的小規模な組織内での支援を想定し，扱うコンポーネントは Java のコンポーネント技術である Java Beans [4] に限定する．

3 利用情報

3.1 概要

ここでの利用情報とは過去にそのコンポーネントを利用して作成されたソフトウェア開発事例から取得される情報の総称である．ここでは，以下の 4 種類の情報に限定して議論する [8].

- 接続関係
- 動作環境
- 利用者
- 同時取得コンポーネント

これらは，過去のコンポーネント利用履歴から獲得可能で，かつ将来の開発に有用であると思われる情報である．利用情報の収集は，各々の開発環境に情報収集機能を付加して，コンポーネントウェアハウスへ自動的に送信することによって行うことができる．各情報の定義とその用途を以下に述べる．

3.2 接続関係

コンポーネント指向ソフトウェア開発では，コンポーネントに含まれるメソッド同士を接続して特定の機能を実現する．

図 3 は，アプリケーション内に 2 つのコンポーネントを組込まれている状態を表している．この例では，コンポーネント 1 に含まれるメソッド A が，コンポーネント 2 に含まれるメソッド E を呼び出して，その機能を利用している．

メソッド同士の接続は，それらを視覚的に繋ぎ合わせるビルダと呼ばれる開発ツールを用いて行う．アプリケーション内で作成された接続関係は，コンポーネント本体を書き換えずに，アダプタと呼ぶオブジェクト (ビルダツールが自動生成する) によって実現される．したがって，ビルダツールの改造や，生成されたコードの解析によって，接続関係をソフトウェア開発中，あるいは開発後に容易に取得することが可能である．

図 4 は，Java Beans[4] 開発キットである BDK(Beans Development Kit)[3] に含まれるビルダツール，Bean Box を用いてアニメーションの表示を行うコンポーネントと，それを制御する 2 つのボタンコンポーネントから成る単純なアプリケーションを構成する手順を示している．このアプリケーションでは，start ボタンを押下すると Juggler と呼ばれるコンポーネントがアニメーションを開始し，stop ボタンを押下するとアニメーションを停止

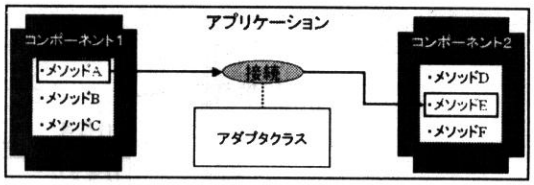

図 3: コンポーネントの接続関係

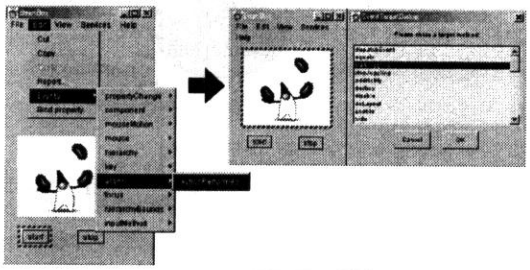

図 4: コンポーネント接続事例

する．このような機能を実現するためには，各ボタンコンポーネントとアニメーションコンポーネントのメソッド同士を接続する必要がある．

コンポーネントに含まれるメソッドの一覧は，メニュー形式で表示される．ここでは，start ボタンを押下された際に呼び出される actionPerformed メソッドを選択し (図 4 左)，アニメーションコンポーネントに含まれる startJuggling と呼ばれるアニメーションを開始するメソッドに接続することによって，start ボタンを押すとアニメーションを開始する機能をを実現している (図 4 右)．stop 機能も同様に作成する．その際に取得される接続関係を以下に示す．

Source Component Unique-Code:	365781
Source Component:	OurButton
Source Method:	actionPerformed
Target Component Unique-Code:	5922857
Target Component:	Juggler
Target Method:	startJuggling

Source Component Unique-Code 及び Target Component Unique-Code は，それぞれのコンポーネントのインスタンスの識別に用いられる固有の値である．Source Component は接続の始点であるコンポーネントを示し，Target Component は接続の終点であるコンポーネントを示す．Source Method は Source Component に含まれ，接続の始点であるメソッドを示す．Target Method も同様に Target Component に含まれ，接続の終点であるメソッドを示す．

接続関係は，実際の開発において同時によく利用された具体的な結合例をコンポーネント利用者に提示するために用いられる．

3.3 動作環境

動作環境とは，そのコンポーネントを実際に利用した際の利用者側のプラットフォームで，具体的には，オペレーティングシステム及び実行環境のバージョン，地域情報などを指す．コンポーネントウェアハウスには様々な種類のコンポーネントが登録されており，コンポーネントによっては，オペレーティングシステムや実行環境及び言語等の違いによって，現在開発しているシステムに容易に組み込めない場合がある．したがって，そのコンポーネントが過去にどのようなシステムにおいて利用されたかの実績を記述した情報が有益である．動作環境は，ビルダーツールにコンポーネントが配置された際にJava Virtual Machineの抽出機能を用いて取得される．抽出された動作環境記述(抜粋)の例を以下に示す．

component-name:	sunw.demo.juggler.Juggler
java-runtime-name:	Java 2 Runtime Environment
java-runtime-version:	1.3.0-C
java-vm-vender:	Sun Microsystems Inc.
os-name:	Windows 2000
user-language:	ja
user-region:	JP
...	...

3.4 利用者

ここで言う利用者とは，過去にそのコンポーネントを利用した個人を指す．具体的には，コンポーネントウェアハウスのユーザ情報として収集され，ユーザ名，E-mailアドレス，所属部署等を含む．

各コンポーネントの利用者を記録することで，コンポーネントに不具合及び欠陥が発見された場合に利用者間で相互に連絡を行ったり，コンポーネントの利用方法について利用者間でノウハウを共有したりすることも可能になる．

3.5 同時取得コンポーネント

同時取得コンポーネントとは，利用者がコンポーネントウェアハウスの検索を行ってコンポーネントを取得した際に，同一のセションにおいて同時に取得したコンポーネントの情報である（同一セションとは，支援システムの利用開始から，利用終了までの間を指す）．同一セション中に取得されたコンポーネントは，互いに関連性の高いコンポーネントであると推測される．例えば先の例を挙げると，アニメーションをするJugglerコンポーネントを取得する利用者は，同一セション中にそれを制御するButtonコンポーネントも取得していく可能性が高い．各利用者の同時取得コンポーネントを記録・蓄積していくことによって，あるコンポーネントを取得する際に，それと関連性の高いコンポーネントとして同時に推薦することが可能となる．

4 利用情報を用いた開発支援システム

4.1 概要

本研究で提案する開発支援システムは，コンポーネント利用者の具体的なニーズに沿ったコンポーネント検索を支援し，開発に有用な情報を利用者に提示するシステムである．本システムの機能として，

- 利用情報収集機能
- コンポーネント検索・取得機能
- 利用情報提示機能

が含まれる．

図5は本研究の提案する開発支援システムに基くコンポーネント利用プロセスを表示している．従来（図1）と異なる点は，コンポーネントウェアハウス内のカタログに3節で述べた利用情報が含まれている点である．従来から用いられているカタログ項目による検索の併用も可能であるが，ここでは省略する．

コンポーネント開発者によるコンポーネントの登録は従来通りに行われる．アプリケーション開発者は，従来の項目(仕様書)に加えて利用情報を検索の対象として指定することができる．検索システムは，検索文とマッチしたコンポーネントを利用者に列挙し，利用者は，その中から最適なコンポーネントを選択し開発に利用する．選択の際に，そのコンポーネントの利用情報の詳細を閲覧することによって必要なコンポーネントを絞り込むことが可能である．

検索システムは，HTTPサーバに組み込まれサーバサイドで動作する．ユーザは，WWWブラウザを用いてコンポーネントの検索とダウンロードができる．各ユーザの利用情報は，利用情報収集ツールからHTTPプロトコルによって検索システムに送信され，コンポーネントウェアハウスに随時登録される．以降，各機能について説明する．

4.2 利用情報収集機能

3節で提案した4つの利用情報のうち，接続関係，動作環境は利用者の開発環境（Bean Box）に利用情報収集機能を組み込み，コンポーネントの利用情報をネットワークを通じてコンポーネントウェアハウスに送信し登録する．

利用情報収集機能には，

- 完成したソフトウェアから接続関係を取得する機能
- 開発中にそれぞれのコンポーネントの動作環境を取得する機能
- 収集した利用情報をファイルにまとめる機能
- それらをネットワークを通じて検索システムに送信する機能

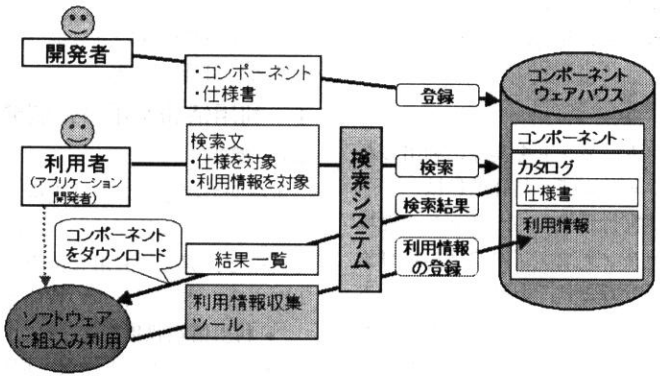

図 5: 提案する開発支援システム

が含まれる．

利用者，同時取得コンポーネントはコンポーネント検索システムで直接取得する．

4.3 コンポーネント検索機能

コンポーネント検索機能とは，利用者が送信した検索文をもとにコンポーネントウェアハウス内を検索し，その結果を利用者に提示する機能である．ここで，利用者が検索システムをどのような流れで利用するのか例を挙げて詳説する．

図6は，検索文入力画面である．この画面で利用者は，必要とするコンポーネントの名前，機能，接続事例及びその実行環境に関する検索文を入力することができる．この画面の例では，名前は未知,「アニメーション」の機能を持ち，過去に「OurButton」コンポーネントから呼出される形で利用された事例があり,「JDK1.2，Windows」で動作するもの，という意味で入力してある．入力後，searchボタンを押下すると検索を開始し，その結果 図7のような画面が表示される．

図7の画面では，与えられる検索文にマッチしたコンポーネントが表示されている．この例では，3件のコンポーネントが表示されている．利用者はここでコンポーネントを選択し直ちに取得（ダウンロード）することができるが，どのコンポーネントが最も要求に近いコンポーネントなのか判断できない場合には，次に説明する利用情報提示機能を用いてその仕様及び他のユーザの利用情報を閲覧してから，取得するコンポーネントを選択することができる．

4.4 利用情報提示機能

利用情報提示機能は，他のユーザの利用情報をコンポーネントの選択と実際の利用に役立つ情報として提供する機能である．利用者が開発中のシステムに組込み可能かどうかの判断材料として，各コンポーネントの仕様書，

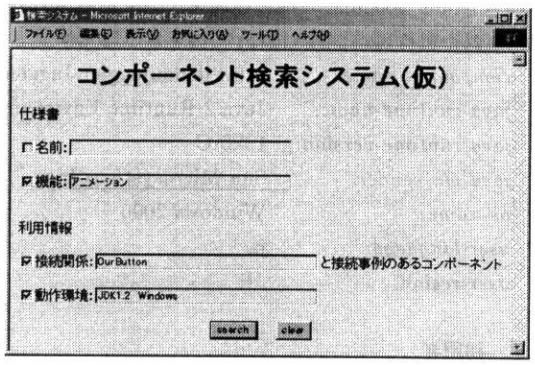

図 6: 検索文入力画面

及び利用情報を閲覧できる．

利用情報は，図7の画面に表示される各リンク（接続関係，動作環境，利用者・同時取得コンポーネント）を辿ることによって閲覧できる．例えば，接続関係へのリンクを選択すると，図8の画面が表示される．

図8はこのコンポーネントの接続関係の事例表示画面である．この画面では、このコンポーネントが接続元（つまり別のコンポーネントの機能を利用する側）として利用された事例と，接続先（つまり別のコンポーネントに対して機能を提供する側）の2種類に分けてサマリが表示される．この例では、接続元としての利用事例は登録されておらず，接続先としての利用事例として，OurButtonコンポーネントから呼び出される例（143件）とExplicitButtonコンポーネントから呼び出される例（5件）の2種類の接続事例が示されている．また，詳細としてOurButtonから接続された事例では，OurButtonのactionPerformedメソッドからJugglerのstartJugglingへの接続であることが示されている．このように，利用を

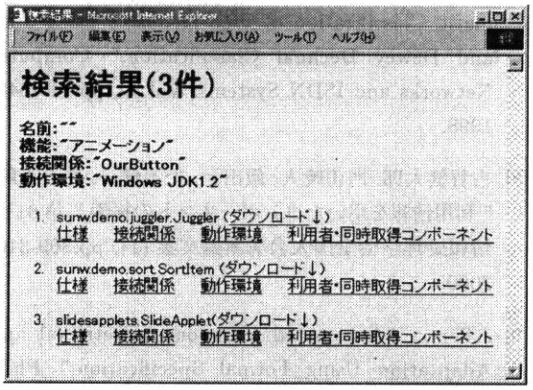

図 7: 検索結果表示画面

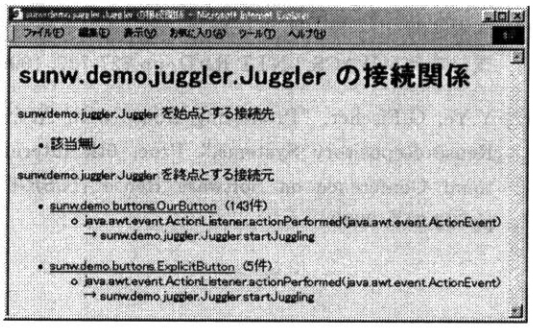

図 8: 接続関係表示画面

検討するコンポーネントと，他のコンポーネントとの接続実績やそのときの具体的なメソッドの対応関係を提示することが，コンポーネントの選択や利用の支援となる．

図9は，Juggler コンポーネントの利用者・同時取得コンポーネントの表示画面である．画面上部が利用者を示しており，この例では過去に本システムから Juggler コンポーネントを取得したユーザ名が表示されている．このリンクを辿ることによって，そのユーザの個人情報を閲覧することができる．もし利用者が Juggler コンポーネントを利用している時に何らかの問題が発生した場合に，これらのユーザと情報を交換することで過去のコンポーネント利用のノウハウを共有することができる．

画面下部は，Juggler コンポーネントの同時取得コンポーネントである．この表示例では，過去の利用者が OurButton コンポーネントを同時に取得した回数が過去に 5 件，ExplicitButton コンポーネントを同時に取得した回数が 1 件あることが示されている．これらのコンポーネントは Juggler コンポーネントの利用と何らかの関連があると予想され，利用者のコンポーネント選択の

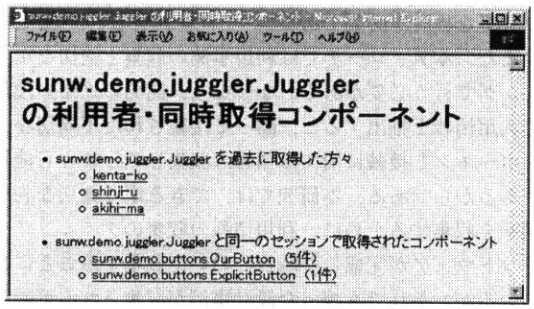

図 9: 利用者・同時取得コンポーネント表示画面

参考になる．

以上のような情報を参照しつつ，利用者は検索結果の中から必要なコンポーネントを絞り込んでいくことができる．また，開発に有用な事例情報を収集することができる．

5 考察

本研究では，利用者のコンポーネントの検索及びその選択と利用を支援するために，他のユーザの利用情報を収集し共有することによって，利用者の具体的なニーズに沿った検索及び，具体的な開発事例の提供を可能とする方法について検討した．

図 2 の L3，L4 内に存在するコンポーネントの増加によって適切なコンポーネントの検索が困難になる問題については，事例収集ツールと検索システムを用いて利用者の具体的なニーズに沿った検索を可能にすることで対処している．本システム単体では，L1（既知）の範囲を大きくしたり，L4 の中から誰も使用したことのないコンポーネントを発見したりすることは直接はできないが，$(R - L1)$ に含まれるコンポーネントから，適切なコンポーネントを取得するための支援としては有用であると考える．

一方今回のアプローチでは以下のような仮定を置いている．

(1) JavaBeans の利用を仮定

本実装では，使用するコンポーネント技術は JavaBean に限定している．ActiveX/DCOM，CORBA などの他のコンポーネント技術には，個別の対応が必要である．

(2) 開発ツールとして BDK を仮定

本実装では，利用情報収集のために開発ツールを改造した．そのため，BDK のように予めソースコードが公開されている開発ツールのみ利用可能である．

(3) 比較的小規模なコミュニティ内での利用を仮定

現段階では，インターネット上の公の環境に本システムを適用する際には，開発上の機密や，個人のプライバ

シー保護が問題となる．利用情報の開示方法についてより詳細な検討が必要である．

また，本アプローチでは利用事例の収集を適切なタイミングで行う必要がある．なぜなら，開発の途中の段階で利用情報を抽出すると，誤って接続された不適切なコンポーネント接続に関する情報も収集してしまう可能性があるからである．本研究では，できる限り適切な利用情報を収集するために，利用情報の収集のタイミングをソフトウェアが完成した時点において収集することにしているが，それでも誤った設計事例が収集される可能性は存在する．収集された事例の信頼度を高めるための工夫が必要である．

6 おわりに

本稿では，コンポーネント指向ソフトウェア開発を支援する目的で，利用情報を用いた検索支援システムを提案した．この支援システムを用いて他のユーザの利用情報を開発者間で共有することによって，有用な開発情報を得ることが可能であると考えられる．本手法は，従来からのコンポーネント検索手法を置き換えるものではないが，従来手法と併用することで，コンポーネントの検索と利用の効率向上に寄与すると考えられる．

謝辞

本研究に対して貴重な助言をいただいた，日立製作所公共システム事業部の福地 豊氏に深く感謝いたします．

参考文献

[1] 青山幹雄, 中所武司, 向山博, " コンポーネントウェア," 共立出版, 1998.

[2] V.Basili, L.Briand, W.Melo, "How Reuse Influences Productivity in Object-Oriented Systems," Commun. of the ACM 39, 10, pp.104-116, 1996.

[3] Beans Development Kit http://java.sun.com/products/javabeans/software/bdk_download.html

[4] Java Beans http://java.sun.com/products/javabeans/

[5] B.Fischer, M.Kievernagel, W.Struckman, "VCR:A VDM-based software component retrieval tool," Proc. ICSE-17 Workshop on Formal Methods Application in Software Engineering Practice, 1995.

[6] S.Henninger, "Supporting the Construction and Evolution of Component Repositories," Proc. ICSE-18, pp.279-288, 1996.

[7] C.Jenkins, M.Jackson, P.Burden, J.Wallis, "Automatic Classification of Web resources using Java and Dewey Decimal Classification," Computer Networks and ISDN Systems, Vol.30, pp.646-648, 1998.

[8] 古賀健太郎, 門田暁人, 飯田元, 松本健一, 井上克郎, " 利用情報を用いたコンポーネントの検索," 第61回情報処理学会全国大会講演論文集 (1), pp.309-310, 2000.

[9] J.Penix, "Automated Component Retrieval and Adaptation Using Formal Specification," Ph.D Dissertation, University of Cincinnati, April, 1998.

[10] I.Sommerville, et al., "Software Engineering forth edition," Addison-Wesley, 1992.

[11] 高木浩則, 山本修一郎, " 広域コンポーネント利用支援システム WACS," NTT R&D, pp.727-737, 1996.

[12] Y.Ye, G.Fischer, "Promoting Reuse with Active Reuse Repository Systems," Proc. 6th Internatonal Conference on Software Reuse (ICSR-6), pp302-317, 2000.