

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-242811

(P2008-242811A)

(43) 公開日 平成20年10月9日(2008.10.9)

(51) Int.Cl.

G06F 9/44 (2006.01)

F I

G06F 9/06 620K

テーマコード (参考)

5B176

審査請求 未請求 請求項の数 3 O L (全 13 頁)

(21) 出願番号 特願2007-82379 (P2007-82379)
 (22) 出願日 平成19年3月27日 (2007. 3. 27)

(71) 出願人 000233055
 日立ソフトウェアエンジニアリング株式会社
 東京都品川区東品川四丁目12番7号
 (74) 代理人 100095267
 弁理士 小島 高城郎
 (74) 代理人 100124176
 弁理士 河合 典子
 (74) 代理人 100111604
 弁理士 佐藤 卓也
 (72) 発明者 辻野 誠
 東京都品川区東品川4丁目12番7号 日
 立ソフトウェアエンジニアリング株式会社
 内

最終頁に続く

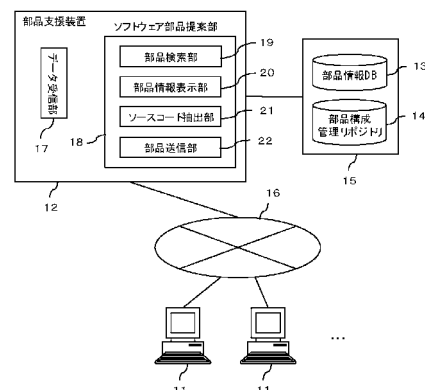
(54) 【発明の名称】 ソフトウェア部品ソースコード生成システム

(57) 【要約】

【課題】ソフトウェア部品を入手する際に、開発者の要求を満たしたソフトウェア部品のソースコードのみを入手することを可能とするソフトウェア部品ソースコード生成システムを提供する。

【解決手段】部品支援装置12では、開発者端末11から送信されたカテゴリ情報とテストデータと結果一致率やカバレッジ率の閾値の情報に基づいて、開発者からの要求を満たすソフトウェア部品を提案する。そして、部品支援装置12におけるソースコード抽出部21では、提案されたソフトウェア部品のソースコードのうち必要なソースコードを抽出する。抽出されたソースコードは、ソフトウェア部品送信部22によって部品支援装置12へと送信される。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

開発者が画面上から要求するソフトウェア部品のカテゴリ情報を入力する機能及び該カテゴリ情報を送信する機能を有する開発者端末と、

ソフトウェア部品を提案する機能及びソースコードの抽出を行う機能を有する部品支援装置と、

ソフトウェア部品の利用状況及び格納情報を管理する部品情報データベース、及びソフトウェア部品のソースコードや仕様書を管理する部品構成管理リポジトリが登録されている記憶装置と

を備えたソフトウェア部品ソースコード生成システムであって、

10

前記部品支援装置は、

前記開発者端末から送信された情報を受信するデータ受信部と、

前記開発者端末から送信された情報に基づいて開発者が要求するソフトウェア部品を検索して提案するソフトウェア部品提案部と

を備えており、

前記ソフトウェア部品提案部は、

前記開発者端末から送信されたカテゴリ情報と要求されるソフトウェア部品のテストデータと結果一致率やカバレッジ率の閾値の情報とに基づいて、開発者の要求を満足するソフトウェア部品を検索するソフトウェア部品検索部と、

検索されたソフトウェア部品のソースコード、仕様情報、関連ファイルを表示するソフトウェア部品情報表示部と、

20

前記ソフトウェア部品情報表示部で表示されている情報のソフトウェア部品を前記開発者端末に送信するソフトウェア部品送信部と、

表示されたソフトウェア部品を開発者が入手する際に開発者が要求するソフトウェア部品のソースコードのうち必要なソースコードを抽出するソースコード抽出部とを備えることを特徴とするソフトウェア部品ソースコード生成システム。

【請求項 2】

前記部品支援装置は、前記開発者端末から送信されるソフトウェア部品のテストデータに基づいて前記部品構成管理リポジトリから開発者が要求するソフトウェア部品と類似しているソフトウェア部品を検索して提供する際に、登録されたソフトウェア部品のソースコードから開発者が必要とするソースコードを抽出して実行可能な状態でソースコードを提供する機能を有することを特徴とする請求項 1 に記載のソフトウェア部品ソースコード生成システム。

30

【請求項 3】

前期ソースコード抽出部は、開発者の必要とするソフトウェア部品のソースコードを抽出する際に、テストデータと前記部品構成管理リポジトリに格納されているソフトウェア部品のソースコードとを用いてテストを実行し、そのとき得られるカバレッジの結果からテスト時に処理が行われたソースコードを開発者が必要なソースコードと決定し、開発者に提供するソースコードとして抽出することを特徴とする請求項 1 又は請求項 2 に記載のソフトウェア部品ソースコード生成システム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、開発者が要求するソフトウェア部品を、開発者によって作成されたテストデータを基にソフトウェア部品が蓄積されている構成管理リポジトリから探し出して提案する機能を有するとともに、開発者がソフトウェア部品のソースコードを入手する際に開発者によって利用されるソースコードのみを抽出する機能を有するソフトウェア部品ソースコード生成システムに関するものである。

【背景技術】

【0002】

50

ソフトウェア部品の検索サイトにおいて、検索して見つけたソフトウェア部品に開発者の要求を満たす機能があったとしても、ソフトウェア部品のソースコードの中で開発者が使用しない不要なソースコードが含まれる場合がある。また、開発者が必要とするソースコードを提供するために、ソースコード間の依存関係を把握する必要がある。特許文献1に記載された技術では、ソフトウェア部品の合成可能範囲の情報から、複数のソフトウェア部品を組み合わせ新たなソフトウェア部品を開発し、ここで組み合わせたソフトウェア部品に対して開発者が必要なソースコードを選択して、不要なソースコードを省くことができるようにすることで、開発者が要求するソフトウェア部品を生成する方法が開示されている。

【特許文献1】特開2006-031424号公報

10

【発明の開示】

【発明が解決しようとする課題】

【0003】

しかしながら、特許文献1に記載された技術の場合、組み合わせたソフトウェア部品のソースコードから開発者が不要なソースコードを選択して取り除くようにしているため、開発者がソースコード中の情報を把握しておく必要がある。取り除くことで要求を満たすソフトウェア部品が開発されるとしても、組み合わせたソフトウェア部品のソースコードを開発者が必要であるか判断して除去することは容易ではない。そのため、開発者はソフトウェア部品のソースコードの中で必要なソースコードがどれであるか判断することに相当の時間を要することになる。

20

【0004】

本発明の目的は、こうした従来技術の問題点を解決するためになされた発明であり、該当するソフトウェア部品を提供する際に不要なソースコードを自動的に取り除くことにより開発者がソフトウェア部品のソースコードの内容を理解する必要なく要求を満たしたソフトウェア部品の必要十分なソースコードを入手することを可能とするソフトウェア部品ソースコード生成システムを提供することである。

【課題を解決するための手段】

【0005】

上記目的を達成するために、本発明のソフトウェア部品ソースコード生成システムは、開発者が画面上から要求するソフトウェア部品のカテゴリ情報を入力する機能及び該カテゴリ情報を送信する機能を有する開発者端末と、ソフトウェア部品を提案する機能及びソースコードの抽出を行う機能を有する部品支援装置と、ソフトウェア部品の利用状況及び格納情報を管理する部品情報データベース、及びソフトウェア部品のソースコードや仕様書を管理する部品構成管理リポジトリが登録されている記憶装置とを備えたソフトウェア部品ソースコード生成システムであって、前記部品支援装置は、前記開発者端末から送信された情報を受信するデータ受信部と、前記開発者端末から送信された情報に基づいて開発者が要求するソフトウェア部品を検索して提案するソフトウェア部品提案部とを備えており、前記ソフトウェア部品提案部は、前記開発者端末から送信されたカテゴリ情報と要求されるソフトウェア部品のテストデータと結果一致率やカバレッジ率の閾値の情報とに基づいて、開発者の要求を満足するソフトウェア部品を検索するソフトウェア部品検索部と、検索されたソフトウェア部品のソースコード、仕様情報、関連ファイルを表示するソフトウェア部品情報表示部と、前記ソフトウェア部品情報表示部で表示されている情報のソフトウェア部品を前記開発者端末に送信するソフトウェア部品送信部と、表示されたソフトウェア部品を開発者が入手する際に開発者が要求するソフトウェア部品のソースコードのうち必要なソースコードを抽出するソースコード抽出部とを備えることを特徴とする。

30

40

【発明の効果】

【0006】

本発明のソフトウェア部品ソースコード生成システムによれば、次のような効果がある。

50

・開発者によって要求されるソフトウェア部品のソースコードから、必要とされるソースコードのみが命令単位で抽出されるとともに、抽出されたソースコードが実行可能な状態で開発者へ提供される。

・開発者によって与えられたテストデータから、開発者の必要とするソースコードが自動で提供されることでソフトウェア部品の入手が容易になり、ソフトウェア開発作業の効率化が可能となる。

【発明を実施するための最良の形態】

【0007】

以下、本発明を適用したソフトウェア部品ソースコード生成システムの一実施形態について説明する。

図1に示すように、本実施形態に係るソフトウェア部品ソースコード生成システムは、複数の開発者端末11と、部品支援装置12と、部品情報データベース13及び部品構成管理リポジトリ14が登録されている記憶装置15とを備えている。

開発者端末11は、ソフトウェア部品を検索するとともに、ソフトウェア部品のダウンロードを行うために開発者によって使用される装置である。

【0008】

部品支援装置12は、開発者からの要求に対して要求機能のソフトウェア部品検索や、ソフトウェア部品の提案、ソフトウェア部品の必要部分のソースコード抽出を行う機能を有している装置である。部品情報データベース13には、ソフトウェア部品に関する情報が管理されている。部品構成管理リポジトリ14には、ソフトウェア部品やソフトウェア部品の仕様書に関する情報、及びテストドライバが格納されている。

そして、開発者端末11はネットワークとしてのインターネット16を介して部品支援装置12と通信可能に接続されている。

【0009】

部品支援装置12は、開発者から送信された要求に関する情報を受信するデータ受信部17と、開発者の要求を満足するソフトウェア部品を検索して開発者に提案するソフトウェア部品提案部18とを備えている。

ソフトウェア部品提案部18には、ソフトウェア部品検索部19、ソフトウェア部品情報表示部20、ソースコード抽出部21、ソフトウェア部品送信部22が備えられている。

【0010】

ソフトウェア部品検索部19は、開発者が要求するソフトウェア部品を部品構成管理リポジトリ14から検索する機能を有する。ソフトウェア部品情報表示部20は、ソフトウェア部品検索部19で検索されたソフトウェア部品を提案ソフトウェア部品であるソースコードや仕様情報として開発者端末11の画面に表示する機能を有する。ソースコード抽出部21は、開発者が提案ソフトウェア部品をダウンロードする際に開発者が必要とするソースコードのみを抽出する機能を有する。そして、ソフトウェア部品送信部22は、開発者が提案したソフトウェア部品を取得したい場合に、ソフトウェア部品のソースコード、仕様書に関する情報を開発者に送信する機能を有する。

【0011】

次に、開発者端末11の画面上に表示される開発者からのソフトウェア部品提案に関する情報を入力するソフトウェア部品検索依頼画面31について説明する。

図2に示すように、ソフトウェア部品検索依頼画面31には、テキストボックス32、チェックボックス33、検索ボタン34、キャンセルボタン35、結果一致率36、カバレッジ率37が表示される。

【0012】

テキストボックス32の欄には、開発者が要求するソフトウェア部品のテストデータの格納場所が記述される。チェックボックス33では、開発者が要求するソフトウェア部品の分類が指定される。検索ボタン34が押下されると、検索の実行依頼が要求される。キャンセルボタン35が押下されると、ソフトウェア部品検索がキャンセルされる。結果一

10

20

30

40

50

致率 36 の欄には、開発者が用意するテストデータに対するテスト結果の閾値が表示される。

【0013】

カバレッジ率 37 の欄には、開発者によって用意されたテストデータに対してテスト実行時に得られるカバレッジの閾値が表示される。

結果一致率 36 とカバレッジ率 37 との閾値を超えるものがソフトウェア部品として提案される。また、検索カテゴリに表示される分野は部品情報データベースに格納されている分類の項目が表示される。

【0014】

次に、開発者に提案されるソフトウェア部品のソースコードや仕様情報を表示する提案部品表示画面 38 について説明する。

図 3 に示すように、提案部品表示画面 38 には、ファイルリスト 39、提案部品仕様情報 40、ソースビュー 41、検索結果 42、カバレッジ率 43、結果一致率 44、次へボタン 45、前へボタン 46、ダウンロードボタン 47、キャンセルボタン 48 が表示される。

【0015】

ファイルリスト 39 の欄には、提案されるソフトウェア部品と依存関係のあるファイルが表示される。提案部品仕様情報 40 の欄には、部品構成管理リポジトリ 14 に格納されている提案するソフトウェア部品の仕様書の内容が表示される。ソースビュー 41 の欄には、提案するソフトウェア部品のソースコードが表示される。ソースビュー 41 の欄に表示されるソースコードは提案するソフトウェア部品の関数(またはメソッド)が表示される。検索結果 42 の欄には、提案ソフトウェア部品数が表示される。カバレッジ率 43 の欄には、提案ソフトウェア部品のソースコードとテストデータを用いてテストしたときのカバレッジの結果とが表示される。

【0016】

結果一致率 44 の欄には、提案ソフトウェア部品のソースコードとテストデータを用いてテストしたときのテストデータ記載の期待値とどのくらいの割合で一致していたのかが表示され。次へボタン 45 が押下されると、ソフトウェア部品が検索された結果、複数件該当した場合、一つ後の候補のソフトウェア部品情報へと遷移する。前へボタン 46 が押下されると、一つ前の候補のソフトウェア部品情報へと遷移する。ダウンロードボタン 47 は、開発者が提案したソフトウェア部品を取得したい場合であって、ソフトウェア部品送信の要求を依頼する際に押下されるボタンである。キャンセルボタン 48 は、提案したソフトウェア部品の情報画面を閉じる場合であって、終了依頼をする際に押下されるボタンである。

【0017】

次に、部品支援装置 12 における処理の概要について説明する。

図 4 に示すように、まず、開発者端末 11 から画面に入力されたカテゴリ情報やテストデータが送信されると、当該カテゴリ情報やテストデータは部品支援装置 12 のデータ受信部 17 によって受信される(ステップ S401)。次に、部品支援装置 12 によって、受信されたデータを基に部品構成管理リポジトリ 14 に格納されているソフトウェア部品群から要求を満足するソフトウェア部品が検索される(ステップ S402)。

【0018】

次に、要求を満足したソフトウェア部品のソースコード及び仕様情報が提案部品表示画面 38 に表示されることで、開発者にソフトウェア部品が提案される(ステップ S403)。次に、提案部品表示画面 38 で提案されたソフトウェア部品を開発者がダウンロードする際に、該当するソフトウェア部品のソースコードのうち開発者が必要とするソースコードが抽出される(ステップ S404)。そして、抽出されたソフトウェア部品のソースコードが開発者端末 11 に送信される(ステップ S405)。

【0019】

次に、ソフトウェア部品検索部 19 の処理の流れについて説明する。

図 5 に示すように、まず、カテゴリ情報、テストデータ、及び結果一致率やカバレッジ率の閾値情報がデータ受信部 17 によって受信される(ステップ S 5 0 1)。次に、カテゴリをキーとして、部品情報データベース 13 に登録されているソフトウェア部品のソフトウェア部品 ID、ソースコード格納情報が取得される(ステップ S 5 0 2)。次に、ソースコード格納情報から部品構成管理リポジトリ 14 に格納されているソフトウェア部品のソースコードが取得される(ステップ S 5 0 3)。

【0020】

取得されたソースコード及びテストデータからテストドライバが作成される(ステップ S 5 0 5)。次に、作成されたテストドライバが実行されて、テストの実行結果が集計される(ステップ S 5 0 6)。ステップ S 5 0 6 では、カバレッジと実行結果とテストデータに記載の期待値通りのものが得られた割合(以下、結果一致率とする)が集計される。集計された結果、カバレッジや結果一致率がカバレッジに対する閾値と結果一致率に対する閾値を超えているか否かが判断される(ステップ S 5 0 7)。閾値を超えていると判断された場合には(ステップ S 5 0 7)、開発者に提案するソフトウェア部品としてリストに追加される(ステップ S 5 0 8)。また、ステップ S 5 0 7 で取得されたカバレッジの結果が部品情報データベース 13 に格納される(ステップ S 5 0 9)。

【0021】

一方、ステップ S 5 0 7 において、閾値を超えていないと判断された場合にはステップ S 5 0 8、S 5 0 9 の処理は実行されない。ステップ S 5 0 5 からステップ S 5 0 9 までの処理は、ソフトウェア部品構成格納リポジトリから取得したソフトウェア部品の関数(メソッド等)の数だけ繰り返し実行される(ステップ S 5 0 4)。そして、ステップ S 5 0 4 で作成されたソフトウェア部品提案リストはソフトウェア部品情報表示部 20 に送信される(ステップ S 5 6 1 0)。作成されたソフトウェア部品提案リストには、ソフトウェア部品 ID、ファイル名、関数名、関数開始行、関数終了行、カバレッジの結果、結果一致率が記述される。関数開始行や関数終了行はテストドライバ作成時に、ソースコードから抽出される。

【0022】

次に、ソフトウェア部品情報表示部 20 の処理の流れについて説明する。

図 6 に示すように、ソフトウェア部品検索部 19 から提案ソフトウェア部品のリストが取得される(ステップ S 6 0 1)。次に、取得されたリストに記載されているソフトウェア部品 ID をキーとして、部品情報データベース 13 から提案ソフトウェア部品が依存するファイルとソフトウェア部品の仕様格納情報とが取得される(ステップ S 6 0 2)。次に、仕様格納情報を基に、部品構成管理リポジトリ 14 からソフトウェア部品の仕様書が取得される(ステップ S 6 0 3)。

【0023】

次に、提案ソフトウェア部品が依存するファイル名、仕様書の内容とソフトウェア部品のソースコードとが提案部品表示画面 38 のファイルリスト 39、提案部品仕様情報 40、ソースビュー 41 に出力される(ステップ S 6 0 4)。提案部品表示画面 38 に出力される際に、リストに記述された関数開始行から関数終了行のソースコードがソースビュー 41 に、また関数の仕様情報が提案部品仕様情報 40 に表示されるようにする。

【0024】

次に、ソフトウェア部品送信部 22 の処理の流れについて説明する。

図 7 に示すように、開発者によってダウンロードボタン 48 が押下されたときに、画面上に表示されているソフトウェア部品のソフトウェア部品 ID が取得される(ステップ S 7 0 1)。次に、取得されたソフトウェア部品 ID をキーとして、部品情報データベース 13 から該当するソフトウェア部品のソフトウェア部品と依存関係のあるファイルが取得される(ステップ S 7 0 2)。

【0025】

次に、取得された格納情報を基に、部品構成管理リポジトリ 14 からソフトウェア部品のソースコード及び仕様書が取得される(ステップ S 7 0 3)。次に、部品構成管理リポジ

10

20

30

40

50

トリ 14 から取得されたファイルに対して、開発者が必要とするソフトウェア部品のソースコードのみが抽出される（ステップ S 804）。そして、開発者が必要とするソフトウェア部品のファイル一式は開発者端末 11 へと送信される（ステップ S 705）。

【0026】

次に、ソフトウェア部品検索時に取得したカバレッジの情報から開発者が必要とするソースコードを抽出する流れについて説明する。

図 8 に示すように、ソフトウェア部品 ID に基づいて、対象ソフトウェア部品のカバレッジ情報が部品情報データベース 13 から取得される（ステップ S 801）。また、カバレッジ情報に基づいて依存関係のあるクラスが把握される。次に、部品構成管理リポジトリ 14 から対象ソフトウェア部品のソースコードが取得される（ステップ S 802）。次に、ステップ S 802 で取得されたソースコードからインポート文、パッケージ文、クラス名の宣言が抽出される（ステップ S 804）。

【0027】

次に、ステップ S 804 で宣言されたクラス内でカバレッジが取得されたメソッドについて抽出される（ステップ S 806）。次に、ステップ S 806 で追加されたメソッド内でカバレッジを取得することができなかったソースコードの対象行番号、開始列番号から抽出処理を行う位置情報が取得される（ステップ S 808）。次に、ステップ S 808 で取得された位置情報から、開始列番号から終了列番号の文字数分の命令処理が半角スペースに置き換えられる（ステップ S 809）。

【0028】

抽出されたソースコードから、不要な処理が取り除かれる（ステップ S 810）。ステップ S 804 からステップ S 809 までの処理は、カバレッジが取得されたクラス数の数だけ繰り返し実行される（ステップ S 803）。また、ステップ S 806 からステップ S 809 までの処理は、カバレッジが取得されたメソッド数の数だけ繰り返し実行される（ステップ S 805）。そして、ステップ S 808 及びステップ S 809 の処理は、追加されたメソッド内の除外対象の命令数の数だけ繰り返し実行される（ステップ S 807）。

【0029】

次に、ソフトウェア部品のソースコードを抽出した中で不要な処理を取り除く作業の中で if 文の命令除去処理の流れについて説明する。

図 9 に示すように、抽出されたソースコードの中で if 文、else-if 文、else 節が宣言されている行番号がネストの深い順に抽出される。抽出された if 文もしくは else-if 文、else 節の開始行番号、開始列番号、開始括弧、閉じ括弧の行番号、列番号が取得される。そして、if 文もしくは else-if 文、else 節の開始行番号、開始列番号が現在の処理行番号、列番号とされる（ステップ S 903）。次に、if 文の括弧内で命令処理が記述されているか否かチェックされる（ステップ S 904）。

【0030】

ステップ S 904 において、命令処理が記述されていた場合、if 文の開始位置から閉じ括弧までの文字が半角スペースに置き換えられる（ステップ S 905）。次に、ステップ S 905 において処理された内容は、if 文の置換処理であったか否かがチェックされる（ステップ S 906）。if 文の置換処理であった場合、この if 文に else-if 文が含まれているか否かがチェックされる（ステップ S 907）。else-if 文が含まれていた場合には、else-if 文に書かれている文字列「else」が半角スペースに置き換えられ（ステップ S 908）。

【0031】

ステップ S 908 では、例えば文字列「else if {log("Error")}」となっていた場合、「if {log("Error")}」というように文字列「else」が半角スペースに置き換えられる。ステップ S 907 において、else-if 文が含まれていなかった場合には、else 節が含まれているか否かがチェックされる（ステップ S 909）。else 節が含まれていた場合には、else 節が半角スペースに置き換えられる（ステップ S 910）。ステップ S 910 では、例えば文字列「else {k++;}」となっていた場合、「k++;」というように else

10

20

30

40

50

節が半角スペースに置き換えられる。ステップ S 9 0 3 からステップ S 9 1 0 までの処理は、i f 文の分岐の数だけ繰り返し実行される（ステップ S 9 0 2）。

【0032】

i f 文のソースコードを整理するアルゴリズムに沿って抽出されたソースコードを整理をしたときの、ソースコード整理前とソースコード整理後との結果例を図 1 0 に示す。ソースコード整理前で else 節の中身が空であったときに、else 節の開始括弧から終了括弧まで内容を除去する処理が実行される。図 9、図 1 0 では条件分岐を例にしたが、他にも while 文や for 文、switch 文などの処理に対してもソースコードの整理が行われる。

【0033】

次に、ソフトウェア部品情報が格納されるソフトウェア部品情報テーブル 5 1 の構造について説明する。

図 1 1 に示すように、ソフトウェア部品情報テーブル 5 1 は、部品 I D 列 5 2、分類列 5 3、概要列 5 4、部品格納場所 5 5、仕様書格納情報 5 6 で構成される。

部品 I D 列 5 2 には、ソフトウェア部品毎に一意の I D が記録される。当該部品 I D 列 5 2 は主キーであり、ソフトウェア部品を登録する際に、一意な値が自動的に発番される。分類列 5 3 は、ソフトウェア部品がどの分野のソフトウェア部品であるのかについての情報が記録される。

【0034】

概要列 5 4 には、ソフトウェア部品の概要が記録される。部品格納場所 5 5 には、部品構成管理リポジトリ 1 4 内におけるソフトウェア部品のソースコードの格納先が記録される。仕様書格納情報 5 6 には、部品構成管理リポジトリ 1 4 におけるソフトウェア部品の仕様書の格納先が記録される。

【0035】

次に、ソフトウェア部品検索時に取得されたソフトウェア部品のカバレッジ情報を格納するカバレッジ情報格納テーブル 5 7 のテーブル構造について説明する。

図 1 2 に示すように、カバレッジ情報格納テーブル 5 7 は、I D 列 5 8、メソッド名 5 9、依存部品 I D 6 0、依存クラス名 6 1、依存メソッド名 6 2、パッケージ名 6 3、開始行番号 6 4、そして終了行番号 6 5 で構成されている。

【0036】

I D 列 5 8 には、検索対象となるソフトウェア部品の I D 列が記録される。メソッド名 5 9 には、開発者の要求するソフトウェア部品のメソッド名が記録される。依存部品 I D 6 0 には、メソッド名 5 9 と依存関係のあるソフトウェア部品の I D が記録される。依存クラス名 6 1 には、メソッド名 5 9 と依存関係のある名称が記録される。依存メソッド名 6 2 には、依存関係があるクラス内のどのメソッドと依存関係があるか表す名称が記録される。

【0037】

パッケージ名 6 3 には、依存クラス名 6 1 のパッケージ構成を表す名称が記録される。開始行番号 6 4 には、依存メソッド名 6 2 のソースコード中の開始行が記録される。そして、終了行番号 6 5 には、依存メソッド名 6 2 のソースコード中の終了行が記録される。なお、ソフトウェア部品 I D 5 8、メソッド名 5 9、依存クラス名 6 1、依存メソッド名 6 2、パッケージ名 6 3 で主キーを表す。

【0038】

次に、ソフトウェア部品検索時にカバレッジを取得しなかったソースコードの情報を格納するカバレッジ未取得情報テーブル 6 6 のテーブル構造について説明する。

図 1 4 に示すように、カバレッジ未取得情報テーブル 6 6 は、部品 I D 列 6 7、メソッド名 6 8、対象クラス名 6 9、対象メソッド名 7 0、パッケージ名 7 1、対象行番号 7 2、開始列番号 7 3、終了列番号 7 4 で構成されている。

【0039】

部品 I D 列 6 7 には、検索対象となるソフトウェア部品の I D が記録される。メソッド名 6 8 には、開発者の要求するソフトウェア部品のメソッド名が記録される。対象クラス

10

20

30

40

50

名 6 9 には、カバレッジ情報格納テーブル 5 7 で登録された対象メソッドの中でカバレッジが未取得のソースコードが存在するクラス名が記録される。対象メソッド名 7 0 には、カバレッジ情報格納テーブル 5 7 で登録された対象メソッドの中でカバレッジが未取得のソースコードが存在するメソッドが記録される。

【0040】

パッケージ名 7 1 には、対象クラス名 6 9 のパッケージ構成が記録される。対象行番号 7 2 には、対象メソッド名 7 0 の中でカバレッジが未取得であるソースコードの行が記録される。開始列番号 7 3 には、対象メソッド名 7 0 の中でカバレッジが未取得であるソースコードの開始列番号が記録される。そして、終了列番号 7 4 には、対象メソッド名 7 0 の中でカバレッジが未取得のソースコードの終了列番号が記録される。なお、部品 I D 6 7、メソッド名 6 8、対象クラス名 6 9、対象メソッド名 7 0、パッケージ名 7 1、対象行番号 7 2、開始列番号 7 3 で主キーを表す。

10

【0041】

以上のように、本実施形態におけるソフトウェア部品ソースコード生成システムによれば、次のような効果がある。

- ・開発者が要求するソフトウェア部品のソースコードから必要とするソースコードのみを命令単位で抽出し、抽出したソースコードを実行可能な状態で開発者へ提供することができる。

- ・開発者がシステムにテストデータを与え、自動で開発者が必要とするソースコードを提供することでソフトウェア部品の入手が容易になる。

20

【図面の簡単な説明】

【0042】

【図 1】ソフトウェア部品ソースコード生成システムの概要図。

【図 2】ソフトウェア部品検索依頼画面の説明図。

【図 3】提案部品表示画面の説明図。

【図 4】部品支援装置における処理の概要を示すフローチャート。

【図 5】ソフトウェア部品検索部の処理の流れを示すフローチャート。

【図 6】ソフトウェア部品情報表示部の処理の流れを示すフローチャート。

【図 7】ソフトウェア部品送信部の処理の流れを示すフローチャート。

【図 8】開発者が必要とするソースコードを抽出する流れを示すフローチャート。

30

【図 9】 i f 文の命令除去処理の流れを示すフローチャート。

【図 10】ソースコード整理前とソースコード整理後との結果例。

【図 11】ソフトウェア部品情報テーブルの構造。

【図 12】カバレッジ取得情報テーブルのテーブル構造。

【図 13】カバレッジ未取得情報テーブルのテーブル構造。

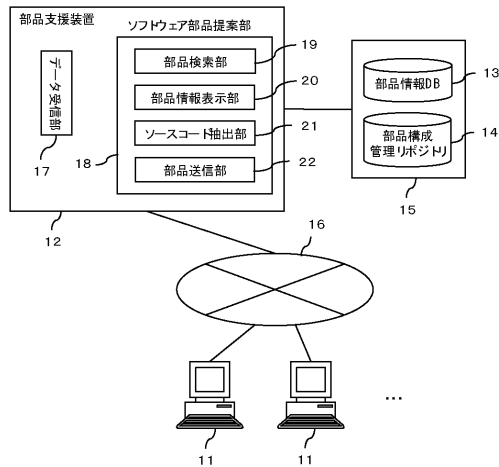
【符号の説明】

【0043】

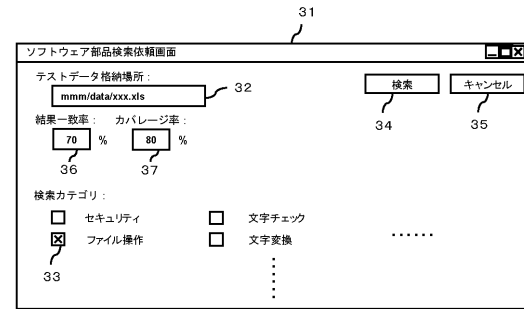
1 1 …開発者端末、1 2 …部品支援装置、1 3 …部品情報データベース、1 4 …部品構成管理リポジトリ、1 5 …記憶装置、1 7 …データ受信部と 1 8 …ソフトウェア部品提案部、1 9 …ソフトウェア部品検索部、2 0 …ソフトウェア部品情報表示部、2 1 …ソースコード抽出部、2 2 …ソフトウェア部品送信部。

40

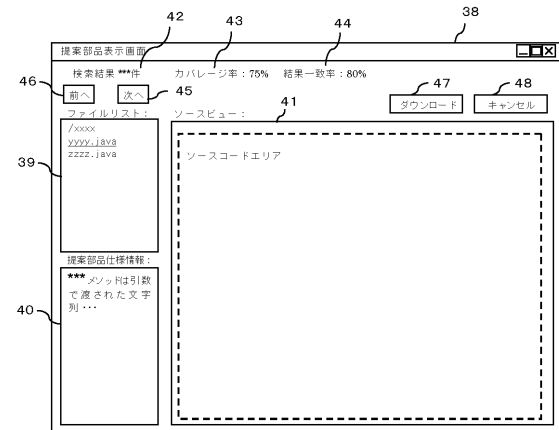
【図 1】



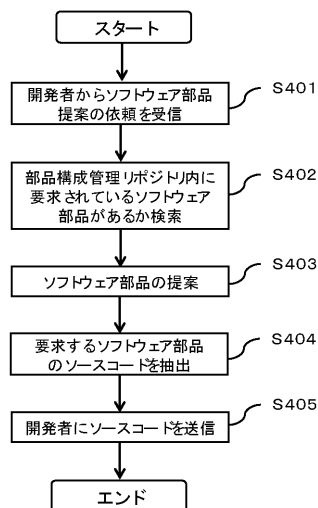
【図 2】



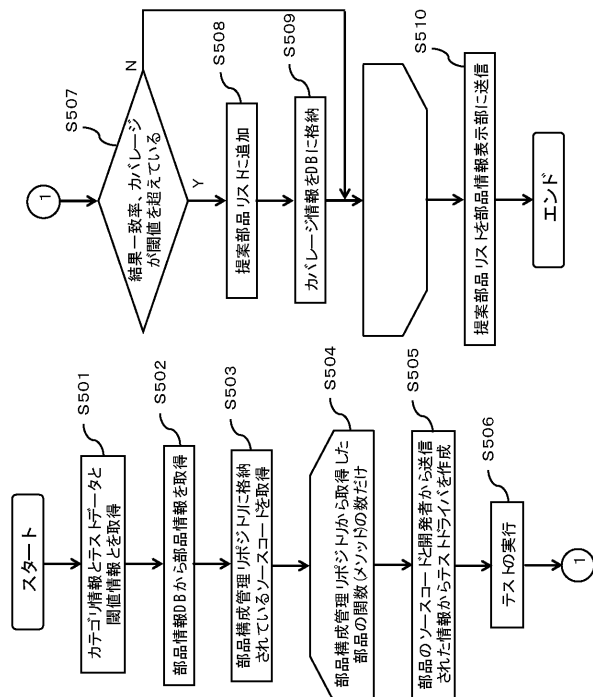
【図 3】



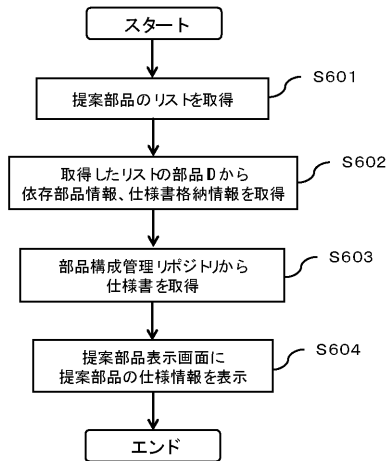
【図 4】



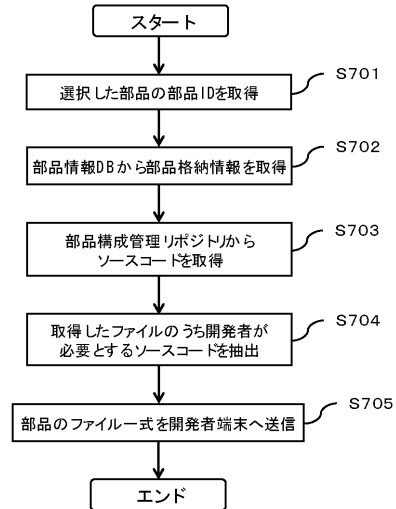
【図 5】



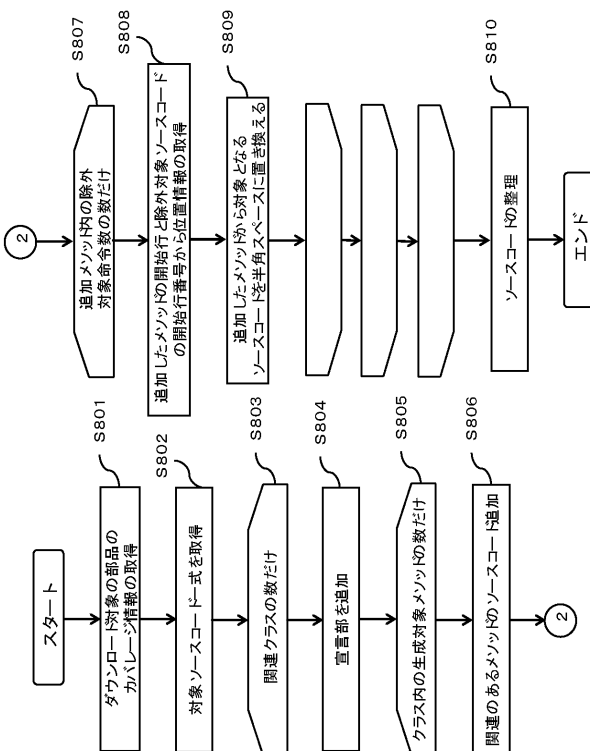
【図 6】



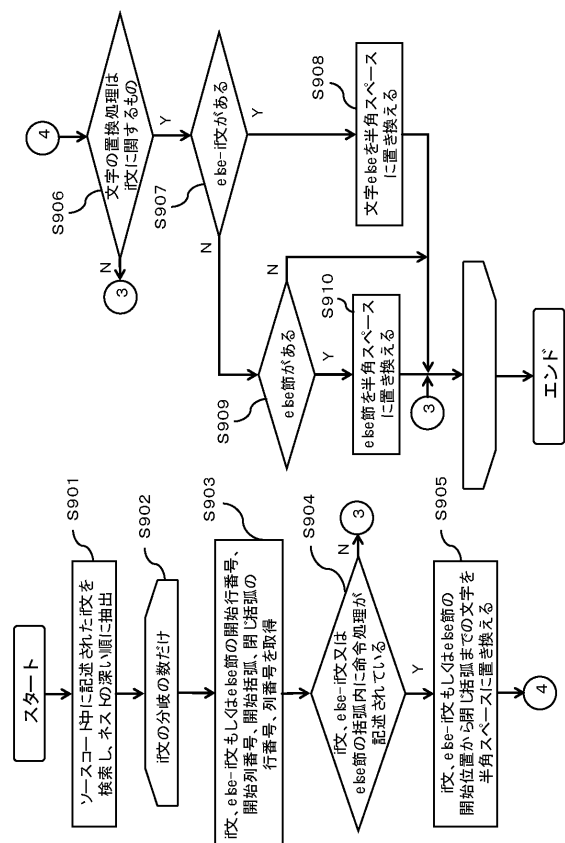
【図 7】



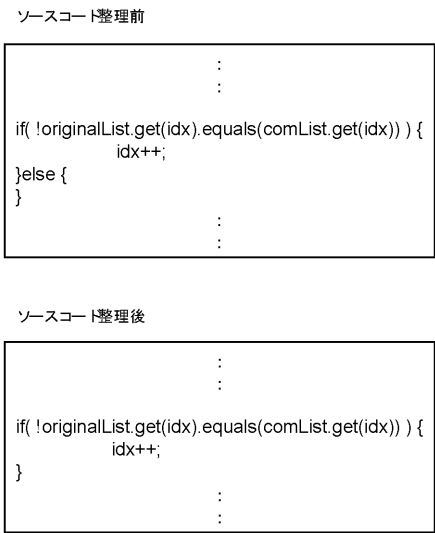
【図 8】



【図 9】



【図 1 0】



【図 1 1】

部品ID	分類	概要	部品格納場所	仕様書格納場所
P000001	ファイル操作	CSV ファイルを ...	P0000001/src/***.xxx	P0000001/spec/***.xls
P000002	ファイル操作	パス情報から ...	P0000002/src/***.xxx	P0000002/spec/***.xls
P000003	セキュリティ	パスワードを ...	P0000003/src/***.xxx	P0000003/spec/***.xls
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

【図 1 2】

部品ID	メンツ名	依存クラス名	依存メンツ名	パッケージ名	開始行番号	終了行番号
P000001	getCsvdata(String name...	StringUtil	stringTrim(String s, ...)	jp.co.xxx.string	64	91
P000001	getCsvdata(String name...	StringUtil	stringCheck(String str, ...)	jp.co.xxx.string	223	237
P000002	copyFile(String path...	FileUtil	genPath(String dir, ...)	jp.co.xxx.file	119	183
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

【図 1 3】

部品ID	メンツ名	対象クラス名	対象メンツ名	パッケージ名	対象行番号	開始列番号	終了列番号
P000001	getCsvdata(String name...	StringUtil	stringTrim(String s, ...)	jp.co.xxx.string	81	13	22
P000001	getCsvdata(String name...	StringUtil	stringCheck(String str, ...)	jp.co.xxx.string	227	9	25
P000002	copyFile(String path...	FileUtil	genPassword(int length, ...)	jp.co.xxx.file	119	13	41
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...

フロントページの続き

(72)発明者 澤田 康秀

東京都品川区東品川4丁目12番7号 日立ソフトウェアエンジニアリング株式会社内

Fターム(参考) 5B176 DD06