(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2012-203777
(P2012-203777A)
(43)公開日　平成24年10月22日(2012.10.22)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 9/44	(2006.01)	G06F	9/06	620K	5B376

審査請求 未請求　請求項の数 7　OL　（全 19 頁）

(21)出願番号　特願2011-69442(P2011-69442)
(22)出願日　平成23年3月28日(2011.3.28)

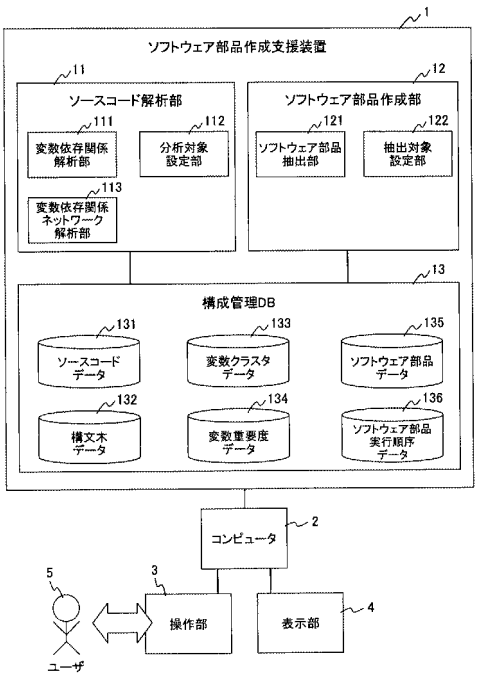
(71)出願人　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　110000350
ポレール特許業務法人
(72)発明者　福田　毅
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内
(72)発明者　吉村　健太郎
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内
(72)発明者　新　吉高
茨城県日立市大みか町七丁目１番１号　株式会社日立製作所日立研究所内

最終頁に続く

(54)【発明の名称】ソフトウェア部品作成支援装置および方法

(57)【要約】
【課題】組込みシステムとして比較的に大規模，複雑化されたソフトウェアであっても、再利用容易なソフトウェア部品を抽出可能とし，作業担当者が熟練者でなくてもソフトウェア部品の作成を容易にする。
【解決手段】コンピュータシステムが組込まれた組込みシステムのソフトウェア部品作成支援装置において、組込みシステムを制御しているソースコード中の変数依存関係を解析し、変数クラスタデータに基づいて、開発者が切り出し対象とする変数クラスタデータを指定可能な変数抽出対象選択手段と、選択された変数クラスタデータに基づき、前記ソースコードよりソフトウェア部品を少なくとも１つ作成するする。
【選択図】図１

図 1



【特許請求の範囲】
【請求項１】
電子制御に搭載されている制御対象を制御するためのソフトウェアであるソースコードから、変数同士の依存関係を対応づけた変数依存関係データを抽出する変数依存関係解析手段と、
前記変数依存関係データに基づいて、変数全体からなる集合を、少なくとも一つの変数で構成される変数クラスタに対応づけた変数クラスタデータを作成する変数依存関係ネットワーク解析手段と、
前記変数クラスタデータに基づいて、ソースコードの部分集合であるソフトウェア部品を出力するソフトウェア部品作成手段を有するソフトウェア部品作成支援装置において、
前記ソフトウェア部品作成手段は、
前記変数クラスタデータに基づいて、開発者が切り出し対象とする変数クラスタデータを指定可能な変数抽出対象選択手段と、
前記変数抽出対象選択手段により選択された変数クラスタデータに基づき、前記ソースコードよりソフトウェア部品を少なくとも１つ作成することを特徴とするソフトウェア部品作成支援装置。
【請求項２】
更新変数ごとにインターフェースを作成
請求項１において、前記ソースコードと前記ソースコードより抽出したソフトウェア部分に対し、更新変数ごとにインターフェースを作成することを特徴とするソフトウェア部品作成支援装置。
【請求項３】
モジュール性指標を表示
請求項１において、前記抽出対象選択手段は、前記変数クラスタデータに基づいて、分割されたクラスタ内の変数依存関係の数とクラスタ間の変数依存関係の数に基づいて算出するモジュール性指標を表示することを特徴とするソフトウェア部品作成支援装置。
【請求項４】
モジュール性指標を追記
請求項１において、前記ソースコードより抽出される前記ソフトウェア部品に対し、モジュール性指標を追記することを特徴とするソフトウェア部品作成支援装置。
【請求項５】
インターフェースとして関数呼び出し採用
請求項２において、前記インターフェースとして関数呼び出しを用いたことを特徴とするソフトウェア部品作成支援装置。
【請求項６】
エレベータへの適用
請求項１に記載のソフトウェア部品作成支援装置において、前記ソフトウェアはエレベータ制御ソフトウェアであり、前記変数として少なくとも、ドアゾーンスイッチ，ファイナルリミットスイッチ，メンテナンスリミットスイッチ，方向リミットスイッチ，または終端階強制減速スイッチのいずれかの状態を示す情報を含むことを特徴とするソフトウェア部品作成支援装置。
【請求項７】
方法：請求項１対応
電子制御に搭載されている制御対象を制御するためのソフトウェアであるソースコードから、変数同士の依存関係を対応づけた変数依存関係データを抽出する変数依存関係解析ステップと、
前記変数依存関係データに基づいて、変数全体からなる集合を、少なくとも一つの変数で構成される変数クラスタに対応づけた変数クラスタデータを作成する変数依存関係ネットワーク解析ステップと、
前記変数クラスタデータに基づいて、ソースコードの部分集合であるソフトウェア部品

を出力するソフトウェア部品作成ステップを有するソフトウェア部品作成支援方法において、
前記ソフトウェア部品作成ステップは、
前記変数クラスタデータに基づいて、開発者が切り出し対象とする変数クラスタデータを指定可能な変数抽出対象選択ステップと、
前記変数抽出対象選択ステップにより選択された変数クラスタデータに基づき、前記ソースコードからソフトウェア部品を少なくとも１つ作成するステップを備えたことを特徴とするソフトウェア部品作成支援方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、家庭用機器，産業用機器，医療寮機器等、電子制御を必要とする製品の特定の機能を実現するために、コンピュータシステムが組込まれている組込みシステムのソフトウェア部品作成支援装置および方法に関する。特に、携帯電話やデジタル家電、さらには自動車，鉄道，エレベータ等の輸送機器など、必要とする機能が多岐に亘るシステム，複数ハードウェア，複数ソフトウェアを組合せた規模の大きなシステムの、ソフトウェア開発，検証，保守支援に好適なソフトウェア部品作成支援装置および方法に関する。
【背景技術】
【０００２】
エレベータ，自動車，建設機械等の技術分野では、いわゆる組込みソフトウェアによって制御対象を制御する組込み制御装置が用いられている。組込みソフトウェアは、従来の機械的機構や電気回路による方式に比べて柔軟かつ高度な制御が実現でき、ソフトウェアの部分的な変更によって多くの派生製品を開発できることが利点として挙げられる。
【０００３】
このような組込み制御装置、例えばエレベータ制御装置では、一定周期または割り込みによってタスクを起動し、行き先階指定ボタンやドア安全センサ等のセンサの入力に基づいて制御変数を更新し、ドア開閉用モータやかご駆動用モータなどのアクチュエータを制御する、いわゆるデータ駆動型の計算モデルが採用されている。
【０００４】
近年、組込み制御装置に求められる制御処理が年々複雑化するとともに、制御変数間の依存関係が複雑化し、ソフトウェアの開発が困難となっている。その一方で、ソフトウェア開発サイクルは短期化を求められている。これに対して、複雑かつ大型のソフトウェアを短期間で開発するには、既存のソフトウェアを出来るだけ効率よく再利用することが重要である。
【０００５】
ソフトウェアの再利用を効率良く行うためには、既存のソフトウェアのソースコードから再利用可能なソフトウェア部品をそのデータと制御構造によって抽出し、ソフトウェア部品ごとに管理することが知られている。この際、ソフトウェア部品の抽出は熟練者による経験に依ることなく、たとえ既存ソフトウェアを熟知していない者であっても容易に抽出可能であることが望まれる。
【０００６】
既存のソフトウェアから互いに依存関係のある文の集合を正しく抽出するため、変数とアドレスの対応関係を解析することで、たとえ共用体や配列，またはポインタを含むプログラムであっても、依存関係のあるプログラムの集合を抽出する技術が知られ、例えば特許文献１に記載されている。
【０００７】
さらに、シングルコアプロセッサ用のプログラムからマルチコアプロセッサ用のプログラムを作成するため，シングルコアプロセッサ用のプログラムから複数の関数（処理）を抽出し、抽出された関数をマルチコアプロセッサ上で並行処理する技術が知られ、例えば特許文献２に記載されている。

【先行技術文献】
【特許文献】
【０００８】
【特許文献１】特開２００９－２３７７６２号公報
【特許文献２】特開２０１０－２３７９６８号公報
【発明の概要】
【発明が解決しようとする課題】
【０００９】
上記従来技術において、特許文献１に記載されたものは、注目する行，または注目する行に含まれる変数に関連したプログラム文を全て抽出する技術であり、同じプログラム文が重複して抽出されてしまうため、複数のソフトウェア部品を作成することが出来ない。
【００１０】
また、特許文献２に記載されたものは、既存のソースコードから関数単位でソフトウェア部品を作成しており、既存のソースコードが関数化されていることを前提としている。そのため、構造化(関数化)されていないソースコードに対してはソフトウェア部品が抽出できないといった課題がある。
【００１１】
本発明の目的は、上記従来技術の課題を解決し、複数のソフトウェア部品を作成する場合においても、同じプログラム文が重複して抽出されることを防ぎ、また、大規模・複雑化した組込みシステムの制御ソフトウェアにおいて、たとえ構造化されていないソフトウェアであっても、ソフトウェア部品を抽出することを可能とすることである。
【課題を解決するための手段】
【００１２】
本発明は、その一面において、ソフトウェアで記述されたコンピュータシステムが組込まれた組込みシステムのソフトウェア部品作成支援装置において、前記コンピュータに記述されたソースコードを入力とし、ソースコード中の変数依存関係を基にソフトウェア構造を解析し、再利用容易なソフトウェア部品を出力することを特徴とする。
【００１３】
本発明の望ましい実施態様においては、電子制御に搭載されている制御対象を制御するためのソフトウェアであるソースコードから、変数同士の依存関係を対応づけた変数依存関係データを抽出する変数依存関係解析手段と、前記変数依存関係データに基づいて、変数全体からなる集合を、少なくとも一つの変数で構成される変数クラスタに対応づけた変数クラスタデータを作成する変数依存関係ネットワーク解析手段と、前記変数クラスタデータに基づいて、ソースコードの部分集合であるソフトウェア部品を出力するソフトウェア部品作成手段とを有するソフトウェア部品作成支援装置において、前記ソフトウェア部品作成手段は、前記変数クラスタデータに基づいて、開発者が切り出し対象とする変数クラスタデータを指定可能な変数抽出対象選択手段と、前記変数抽出対象選択手段により選択された変数クラスタデータに基づき、前記ソースコードよりソフトウェア部品を少なくとも１つ作成することを特徴とする。
【発明の効果】
【００１４】
本発明によれば、組込みシステムとして大規模，複雑化されたソフトウェア（コンピュータプログラム）であっても、変数の依存関係に基づいたソフトウェア部品を作成可能であり、作業担当者が熟練者でなくても再利用容易なソフトウェア部品を作成することができる。
【図面の簡単な説明】
【００１５】
【図１】本発明の一実施の形態によるソフトウェア部品作成支援装置の全体構成を示すブロック図。
【図２】一実施の形態におけるソースコード解析部の構成を示すブロック図。

【図３】一実施の形態におけるソースコードデータを示す図。
【図４】一実施の形態における解析木データを示す図。
【図５】一実施の形態における変数依存関係リンクテーブルを示す図。
【図６】別の実施の形態における変数依存関係リンクテーブルを示す図。
【図７】一実施の形態における分析対象設定の構成を示す図。
【図８】一実施の形態における分析対象データを示す図。
【図９】一実施の形態における変数依存関係ネットワーク解析部を示すブロック図。
【図１０】一実施形態における変数クラスタ解析部の処理を示すフローチャート。
【図１１】一実施の形態におけるリンク切断前の変数依存関係ネットワーク図。
【図１２】一実施の形態におけるリンク切断後の変数依存関係ネットワーク図。
【図１３】一実施の形態における変数クラスタテーブルを示す図。
【図１４】一実施の形態における変数クラスタ間リンクテーブルを示す図。
【図１５】一実施の形態における変数重要度解析部の処理を示すフローチャート。
【図１６】一実施の形態における変数重要度テーブルを示す図。
【図１７】別の実施の形態における変数重要度テーブルを示す図。
【図１８】一実施の形態におけるソフトウェア部品作成部の構成を示すブロック図。
【図１９】一実施の形態によるクラスタ数が１つの場合の抽出対象設定部における表示画面例を示す図。
【図２０】一実施の形態によるクラスタ数が２つの場合の抽出対象設定部における表示を示す図。
【図２１】一実施の形態によるクラスタ数が３つの場合の抽出対象設定部における表示を示す図。
【図２２】一実施の形態によるクラスタ数が４つの場合の抽出対象設定部における表示を示す図。
【図２３】他の実施の形態による抽出対象設定部でモジュール性の変化を示す表示を示す図。
【図２４】同じく、他の実施の形態による抽出対象設定部でモジュール性最大部の強調表示を示す図。
【図２５】さらに他の実施の形態における抽出対象設定部における数値表示を示す図。
【図２６】一実施の形態における抽出対象変数クラスタデータを示す図。
【図２７】一実施の形態におけるソフトウェア部品抽出部の処理を示すフローチャート。
【図２８】一実施の形態における抽出されたソフトウェア部品を示す図。
【図２９】本発明を適用して好適なエレベータシステムの構成を示す図。
【発明を実施するための形態】
【００１６】
以下、図面を参照して本発明による一実施の形態を説明する。
【００１７】
ソフトウェア開発，検証，保守作業において、既存のソフトウェアを再利用することで効率よく派生製品を開発する方法として、既存のソフトウェアを部品化し、部品ごとに管理することが重要視されている。しかし、特にエレベータ等の組込みシステムでは、従来、機械式で行っておいた安全装置等の電子化が進められ、ソフトウェアが大規模・複雑化しており、既存のソフトウェアからソフトウェア部品を抽出するためには、膨大な作業量が必要とされている。
【００１８】
また、大規模・複雑化が進んだソフトウェアから、再利用容易なソフトウェア部品を抽出する工程は、既存のソフトウェア構造を熟知した熟練者の経験に頼る部分が大きく、初心者では極めて困難である。
【００１９】
図１は、本発明の一実施の形態による組込みシステムのソフトウェア部品作成支援装置１の全体像を示すブロック図である。

【００２０】
ソフトウェア部品作成支援部１は、ソースコード管理部１１と、ソフトウェア部品作成部１２と、構成管理ＤＢ１３とを備える。ソフトウェア解析部１１は、構成管理部１３より既存の組込みシステムの制御ソフトウェアであるソースコードデータ１３１を入力とし、構文木データ１３２と，変数クラスタデータ１３３と，変数重要度データ１３４とを出力する。構文木データ１３２は、ソースコードデータ１３１中の変数同士の参照関係や条件文関係などの依存関係を解析し、ソースコード中における演算子とオペランドを節点と葉として表現したものである。変数クラスタデータ１３３は、少なくとも１つの変数で構成される変数クラスタにソースコードデータ１３１中の変数を分類したものである。変数重要度データ１３４は、変数クラスタデータ１３３と，ソースコードデータ１３１中の変数の重要度を示すものである。

【００２１】
ソフトウェア部品作成部１２は，まず、ソースコードデータ１３１と、ソースコード解析部１１の出力結果である構文木データ１３２と、変数クラスタデータ１３３とを入力とし、構文木データ１３２と変数クラスタデータ１３３よりソースコードデータ１３１中の変数依存関係に基づいて、ソフトウェア部品を抽出する。次に、ソースコードデータ１３１の抽出部品であるソフトウェア部品データ１３５と、抽出されたソフトウェア部品の実行順序を管理するソフトウェア部品実行順序データ１３６とを出力する。

【００２２】
ここで、ユーザ５は、ソフトウェア作成支援装置１に対し、ソフトウェア部品作成支援装置の実行環境であるコンピュータ２と、ソフトウェア部品作成支援装置１を操作するための操作端末である操作部３と、ソフトウェア部品作成支援装置１の実行状況を認識するための表示部４を用いて、ソフトウェア部品を作成することができる。

【００２３】
図２は、ソースコード解析部１１の詳細な構成を示す図である。ソースコード解析部１１は、変数依存関係解析部１１１と、分析対象設定部１１２と、変数依存関係ネットワーク解析部１１５とを備えている。変数依存関係解析部１１１は、ソースコードデータ１３１に基づいてソフトウェア実行時の変数更新処理における変数間依存関係を分析する。変数依存関係解析部１１１は、変数依存関係解析部１１１の分析結果である変数依存関係データ１１４と、変数依存関係データ１１４と操作部３によるユーザの操作とに基づいて分析対象とする変数を選択する。変数依存関係ネットワーク解析部１１５は、分析対象設定部１１２の設定結果である分析対象変数データ１１５と、変数依存関係データと分析対象変数データのいずれかに基づいてソースコードデータ１３１における変数依存関係ネットワークを分析して変数クラスタおよび変数重要度を分析する。そして、ソースコード解析部１１は、変数依存関係解析部１１１より構文木データ１３２を出力し、変数依存関係ネットワーク解析部１１３より変数クラスタデータ１３３と、変数重要度データ１３４を出力する。

【００２４】
図３は、ソースコードデータ１３３の詳細を示す図である。ソースコードファイル１３１１は、関数ｘの処理手順と、関数yの処理手順と、関数ｚの処理手順とで構成されている。なお、ソースコードファイル１３１１で用いられている変数ａ，b，c，d，e，f，gはグローバル変数として定義されている。関数ｘにおいて、変数ａを入力とし、変数ｂおよびｃを更新する処理を行っている。関数ｙにおいて、変数ｅを入力とし、変数ｆおよびｇを更新する処理を行っている。関数ｚにおいて，変数ｂとｃを入力とし，変数ｄおよびｅを更新する処理を行っている。

【００２５】
図４は、ソースコードファイル１３１１に基づいて変数依存関係解析部１１１が解析した構文木データ１３２の構成の一部である構文木データ１３２１～１３２４を示す図である。構文木データ１３２１～１３２４は、ソースファイル１３１１内の演算子とオペランドを木構造で示した構成を採っている。例えば、構文木データ１３２２は、ソースコード

ファイル１３１１（図３）の２行目“ｂ＝（ｂ｜ａ）”に対応した木構造となっている。

【００２６】

図５は、ソースコードファイル１３１１に基づいて変数依存関係解析部１１１が抽出した変数依存関係データ１１４の詳細な構成である変数依存関係リンクテーブル１１４１を示す図である。変数依存関係リンクテーブル１１４１は、リンクＩＤ，変数名（始点），変数名（終点）で構成され、変数依存関係を示している。例えば、リンクＩＤ１は、変数名（始点）がａ、変数名（終点）がｂである。リンクＩＤ１は、変数ｂの更新時に変数ａを参照して演算を行っていることを示している。これは、ソースファイル３１の２行目“ｂ＝（ｂ｜ａ）；”に対応している。また、リンクＩＤ３は変数名（始点）がｂ、変数名（終点）がｄであり、リンクＩＤ４は変数名（始点）がｃ、変数名（終点）がｄである。リンクＩＤ３および４は、変数ｄの更新時に変数ｂおよび変数ｃを参照して演算を行っていることを示している。これは、ソースファイル１３１１の１２行目“ｄ＝（ｂ＆ｃ）；”に対応している。

【００２７】

図６は、ソースファイル１３１１に基づいて変数依存関係解析部１１１が抽出した変数依存関係データ１１４の詳細な構成の別形態である変数依存関係リンクマトリクス１１４２を示す図である。変数依存関係リンクマトリクス１１４２は、始点となる変数名と、終点となる変数名からなるリンクをマトリクス形式で表現したものである。例えば、始点をａ、終点をｂとする要素は“１”となっており、変数ｂの更新時に変数ａを参照して演算を行っていることを示している。また、始点をａ、終点をｄとする要素は“０”となっており、変数ｄの更新時には変数ａを参照せずに演算を行っていることを示している。

【００２８】

なお、図６では、始点と終点が逆の要素も“１”とした例を示している。例えば、始点をａ、終点をｂとする要素は“１”となっており、その逆の要素である始点をｂ、終点をａとする要素も“１”としている。

【００２９】

図７は、分析対象設定部１１２の設定画面１１２１を示す図である。本画面では、変数依存関係ネットワークにおける分析対象とする変数を、ユーザがチェックすることによって選択する。例えば、設定画面１１２１では、分析対象から外された変数はなく、変数ａ～ｇの全てが変数依存関係ネットワークの分析対象となっていることを示している。

【００３０】

図８は、分析対象設定部１１２により選択された分析対象変数データ１１５の詳細な構成である分析対象変数テーブル１１５１を示す図である。分析対象変数テーブル１１５１は、分析対象設定部１１２において、変数ａ～ｇを分析対象とした場合の分析対象変数データである。

【００３１】

この例では、図７のデータのうち、始点と終点が逆の要素は削除した例を示している。例えば、始点をａ、終点をｂとする要素がリンクＩＤ１に存在するので、その逆の要素であり、図７に存在した始点をｂ、終点をａとする要素は、図８には存在させていない。

【００３２】

図９は、変数依存関係ネットワーク解析部１１３の詳細な構成を示す図である。変数依存関係ネットワーク解析部１１３は、変数依存関係データ１１４または分析対象変数データ１１５に基づいて、結合度が高く凝集度が低い変数のモジュール構成を示す変数クラスタデータ１３３を抽出する変数クラスタ解析部１１３１と、変数依存関係データ１１４または分析対象変数データ１１５に基づいて、変数の重要度を示す変数重要度データ１３４を抽出する変数重要度解析部１１３２とを備える。

【００３３】

図１０は、変数クラスタ解析部１１３１の詳細な実行フローを示す図である。なお、図１０で示す変数クラスタ解析はリンク媒介中心性コミュニティ解析（Ｎｅｗｍａｎ法）を用いた場合の実行フローであるが、固有ベクトルコミュニティ解析を用いても良い。ステ

ップＳ１１３１０から処理が開始される。ステップＳ１１３１１では、あるリンクが変数間の最短経路上に存在する度数を示すリンク媒介中心性の計算を、全てのリンクに対して実行する。ステップＳ１１３１２では、リンク媒介中心性が最大値を取るリンクの切断を行う。ステップＳ１１３１３では、全てのリンクを切断済みか否かを判定する。全てのリンクを切断済みの場合にはステップＳ１１３１４に進み、リンクが残っている場合にはステップＳ１１３１１に戻る。ステップＳ１１３１４では、結合度が高く凝集度が低いモジュールの性質を示すモジュール性の指標を、リンク分割パターンごとに演算し、最もモジュール性の高いリンク分割パターンを、変数クラスタとして採用する。

【００３４】

モジュール性は、次のように計算される。変数依存関係ネットワークがｋ個のクラスタに分割されているとき、ｋ×ｋのクラスタの関係を示す行列ｅ＝（ｅ$_{ij}$）を作る。ｅ$_{ij}$はクラスタｉに分類される変数とクラスタｊに分類される変数の間に張られる変数依存関係の数が、変数依存関係ネットワーク全体の依存関係に占める割合である。また、ｅの行和ａを以下のようにとる。

【００３５】

【数１】

$$a_i = \sum_j e_{ij}$$

【００３６】

そして、モジュール性Ｑを次のように定式化する。

【００３７】

【数２】

$$Q = \sum_i \left(e_{ii} - a_i^2 \right)$$

【００３８】

例えば、後に分割パターンとして図１２で示すクラスタ分割結果に基づいて対称行列ｅを求めると、ｅは次のようになる。

【００３９】

【数３】

$$e = \frac{1}{14}\begin{pmatrix} 8 & 1 \\ 1 & 4 \end{pmatrix}$$

【００４０】

このとき、モジュール性は約０．３２（０．３１６・・・）と算出される。

【００４１】

図１１は、変数依存関係ネットワーク１１３Ａを図示したものである。これは有向グラフ記法に基づいているが，無向グラフを用いても良い。変数ａ，ｂ，ｃ，ｄ，ｅ，ｆ，ｇをノードとし、分析対象変数テーブル１１５１で指定されているリンク情報をノード間のリンクとして表記したものである。さらに、各リンクの上部にリンクの媒介中心性指標が記載されている。また、モジュール性指標をネットワーク図横に“モジュール性＝０．０”と表示する。

【００４２】

図１２は、リンクが切断された変数依存関係ネットワーク１１３Ｂを図示したものである。これは有向グラフ記法に基づいており、変数ａ，ｂ，ｃ，ｄ，ｅ，ｆ，ｇをノードとし、分析対象変数テーブル１１５１で指定されているリンク情報をノード間のリンクとして表記したものである。また、図１１に記載された変数依存関係ネットワーク１１３Ａにおいてもっともリンク媒介中心性が高いリンクである、“ｄ→ｅ”へのリンクが点線で図示されており、リンクが切断されたことを示している。また、モジュール性指標をネットワーク図横に“モジュール性＝０．３２”と表示する。なお、分析対象変数テーブル１１

５１に基づくクラスタ分析では、変数依存関係ネットワーク１１３Ｂに示す分割パターンがモジュール性最大値を取るため、本分割パータンがクラスタ分割結果となる。

【００４３】

図１３は、変数クラスタデータ１３３の詳細な構成である変数クラスタテーブル１３３１を示す図である。変数クラスタテーブル１３３１は、クラスタ１は変数ａ、ｂ、ｃ、ｄで構成され、クラスタ２は変数ｅ，ｆ，ｇで構成されていることを示している。

【００４４】

図１４は、変数クラスタデータ１３３の詳細な構成である変数クラスタ間リンクテーブル１３３２を示す図である。リンクＩＤ１０１は、クラスタ１を始点とし、クラスタ２を終点とするリンクであり、そのリンクに対応する変数はｄであることを示している。これはクラスタ分割によって、変数ｄはクラスタ１に、変数ｅはクラスタ２に分割されたことにより、分析対象変数テーブル１１５１におけるリンクＩＤ５におけるリンクがクラスタ間を横断するリンクとなり、かつその時の始点となる変数がｄであるためである。

【００４５】

図１５は、変数重要度解析部１１３２の詳細な実行フローを示す図である。図１５では、次数中心性に基づく重要度分析フローを示しているが、推移確率行列固有ベクトル中心性や固有ベクトル中心性を用いても良い。ステップＳ１１３２０から処理が開始される。ステップＳ１１３２１にて変数のＩＤが初期化される。ステップＳ１１３２２にてリンクのＩＤが初期化される。ステップＳ１１３２３にて分析対象リンクに分析対象変数が含まれるか否かが判定される。含まれる場合にはステップＳ１１３２４にて変数の次数中心性に１が加算され、含まれない場合にはステップＳ１１３２５に進む。ステップＳ１１３２５では分析対象のリンクＩＤが最終リンクか否かを判定し、最終リンクである場合にはステップＳ１１３２７に進み、最終リンクではない場合にはステップＳ１１３２６に進む。ステップＳ１１３２６ではリンクＩＤに１を加算し、次のリンクを分析対象としてステップＳ１１３２３に戻る。ステップＳ１１３２７では分析対象変数が最終変数か否かを判定し、最終変数である場合にはステップＳ１１３２９に進み、最終変数ではない場合にはステップＳ１１３２８に進む。ステップＳ１１３２８では変数ＩＤに１を加算し、次の変数を分析対象としてステップＳ１１３２２に戻る。ステップＳ１１３２９で処理を完了する。

図１６は、変数重要度データ１３４の詳細な構成である変数重要度テーブル１３４１を示す図である。変数重要度テーブル１３４１は、変数重要度として、図１５に示した変数重要度解析部１１３２の詳細な実行フローに基づいて算出された次数中心性を示している。例えば、変数ａは、二つの変数ｂ、ｃとリンクで接続されているため、重要度は２となる。また、変数ｄは、３つの変数ｂ，ｃ，ｅとそれぞれリンクで接続されているため、重要度は３を示している。

図１７は、変数重要度データ１３４の詳細な構成である変数重要度テーブル１３４２を示す図である。変数重要度テーブル１３４２は、変数重要度として、推移確率行列固有ベクトル中心性を示している。

【００４６】

図１８はソフトウェア部品作成部１２の構成を示すブロック図である。ソフトウェア部品作成部１２は変数クラスタデータ１３３と操作部３を用いたユーザ５の操作内容とに基づいて抽出対象とするクラスタを選択する抽出対象選択部１２２と，抽出対象選択部１２２にてユーザ５が選択した変数とクラスタの対応関係を示す抽出対象変数クラスタデータ１２３と，ソースコードデータ１３１と構文木データ１３２と抽出対象選択部１２２でユーザ５によって選択された抽出対象変数クラスタデータ１２３とに基づいて，ソフトウェア部品となるプログラムをソースコードデータ１３１より抽出し，かつ抽出されたソースコードに関数呼び出し方法を用いたインターフェースを作成するソフトウェア部品抽出部１２１とを備え，ソフトウェア部品データ１３５とソフトウェア部品実行順序データ１３６を出力する。

【００４７】

次に、本発明の大きな特徴である抽出対象設定（選択）部における，変数クラスタデータの選択画面の例を説明する。

【００４８】

図１９～図２５は、抽出対象設定部１２２（図１，図１８）において、ユーザ５が、変数クラスタデータを選択するための画面例を示している。これらの画面を見ながら、開発者は、ソフトウェア部品作成対象となる変数クラスタデータを選択する。そして、選択された変数クラスタデータに基づき、複数のソフトウェア部品を作成することができる。

【００４９】

まず、図１９～図２２は、本発明の一実施の形態によるクラスタ数が１～４の場合の抽出対象設定部１２２における第１の表示画面例を示している。図１９の抽出対象選択部の設定画面１２２１は、クラスタ数が１つの場合を示しており、ａ～ｇの全ての変数が１つのクラスタに分類されている。また、モジュール性は０．００である。ユーザ５は、変数のクラスタ分割に対し、本分割を「選択」しても良いし、「次へ」を押してクラスタ分割を進めても良い。

【００５０】

図２０は、抽出対象選択部１２２のクラスタ数が２つの場合の設定画面１２２２を示す図である。ａ～ｄを一つのクラスタに分類し、残りのｅ～ｇをもう１つのクラスタに分類した状態である。また，モジュール性は０．３２である。

【００５１】

図２１は、抽出対象選択部１２２のクラスタ数が３つの場合の設定画面１２２３を示す図である。ａ，ｃを一つのクラスタに分類し、ｂ，ｄを他のもう一つのクラスタに分類し、残りのｅ～ｇを最後のクラスタに分類した状態である。また，モジュール性は０．２４である。

【００５２】

図２２は、抽出対象選択部１２２のクラスタ数が４つの場合の設定画面１２２４を示す図である。ａ，ｃを一つのクラスタに分類し、ｂ，ｄを他のもう一つのクラスタに分類し、ｅ，ｇをさらに別のクラスタに分類し、ｆを最後のクラスタに分類した状態である。また，モジュール性は０．１３である。

【００５３】

以上の表示画面を有することで、ユーザ５は変数クラスタのクラスタ数や、クラスタ内の変数グループ内容とモジュール性を基に、抽出すべきソフトウェア部品を容易に判断することが可能である。

【００５４】

図２３～図２４は、他の実施の形態による抽出対象選択部１２２の設定画面を示す図で、横軸にクラスタ数、縦軸にモジュール性を示す座標を用いることで、図１９～図２２の画面と異なり、クラスタ数に対するモジュール性の変化を容易に読み取ることができる。

【００５５】

図２４は、図２３で示した設定画面１２２５に対し、モジュール性最大をとるクラスタ数における点を強調表示するとともに、その点のモジュール性の値を示したものである。

【００５６】

これにより、ユーザ５は、より容易にモジュール性最大のクラスタ数を知ることが可能となる。

【００５７】

図２５は、さらに他の抽出対象選択部１２２の設定画面１２２７を示す図である。クラスタ数とモジュール性の表示に、テーブル表記を用いることで、ユーザ５の認識容易性をサポートしている。

【００５８】

図２６は、抽出対象選択部１２２においてユーザ５に選択された結果である抽出対象変数クラスタデータ１２３を示す抽出対象変数クラスタテーブル１２３１を示す図である。ここでは、抽出対象選択部１２２にて、ユーザ５がクラスタ数２，モジュール性０．３２

を選択した場合である。クラスタＩＤ１０１は、変数ａ～ｄが分類されていることを表わしている。同様に、クラスタＩＤ１０２は、変数ｅ～ｇが分類されていることを示している。

【００５９】

図２７は、ソフトウェア抽出部１２１の処理を示すフローチャートであり、ステップＳ１２１００から処理が開始される。ステップＳ１２１０１にて、クラスタのＩＤが初期化される。ステップＳ１２１０２にて、変数ＩＤが初期化される。ステップＳ１２１０３にて、変数を更新しているソースコード行番号を、構文木データより取得する。ステップＳ１２１０４にて、行番号に基づいて、変数を更新しているソースコードを抽出する。ステップＳ１２１０５にて、ソースコード抽出場所に、ステップＳ１２１０４にて抽出したソースコードを呼び出す関数を作成する。ステップＳ１２１０６にて、変数が最終変数かどうか確認する。変数が最終変数であった場合（ＹＥＳ）はステップＳ１２１０８に進み、そうでない場合（ＮＯ）はステップＳ１２１０７に進む。ステップＳ１２１０７にて、変数ＩＤをインクリメントして処理を続ける。ステップＳ１２１０８にて、抽出したソースコードを結合し、ソフトウェア部品を作成する。ステップＳ１２１０９にて、クラスタが最終クラスタかどうか確認する。クラスタが最終クラスタであった場合（ＹＥＳ）は、ステップＳ１２１１１に進み、そうでない場合（ＮＯ）はステップＳ１２１１０に進む。ステップＳ１２１１０にて、クラスタＩＤをインクリメントして処理を続ける。ステップＳ１２１１１にて、ソースコード抽出元であるソースコードを、ソフトウェア部品実行順序情報として登録する。ステップＳ１２１１２にて処理を終了する。

【００６０】

図２８は、ソフトウェア部品抽出部１２１により抽出されたソフトウェア部品１３５１，１３５２，およびソフトウェア部品実行順序情報１３６１を示す図である。この図においては、これまでの実例と同じく、抽出対象選択部１２２にてユーザ５がクラスタ数２，モジュール性０．３２の値を指定した場合に得られるソフトウェア部品データ１３５とソフトウェア部品実行順序データ１３６を示している。ソフトウェア部品実行順序１３６１とソフトウェア部品１３５１，１３５２は、呼び出し関数にて接続関係にある。このような構造を持つことによって、ソフトウェア部品抽出前のソースコード１３１１と、変数の更新順序を同じくすることが可能である。また、ソフトウェア部品１３５１，１３５２と、ソフトウェア部品実行順序１３６１に対してモジュール性を表記することで、ユーザ５が変数クラスタ設定部１２２で設定した変数クラスタ情報を反映していることを容易に確認することができる。

【００６１】

図２９は、本発明を適用して好適なエレベータシステム１００００の構成を示す図である。エレベータシステム１００００は、エレベータ制御装置６と、エレベータかご７と、複数のエレベータの協調動作を制御する群管理制御装置８１と、他のエレベータかごを制御するエレベータ制御装置８２を備えている。また、行き先階ボタンが押されたことを検知する行き先階ボタンセンサ９１、ドアへの挟み込みを検知するドア安全センサ９２、ドアの開閉をおこなうドア駆動モータ９３、および、かごを昇降駆動するモータ９４を備えている。さらに、エレベータ制御装置６、群管理制御装置８１、およびエレベータ制御装置８２の間の通信を行う通信回線９５とで構成される。

【００６２】

ソフトウェア部品作成支援装置１にて作成されたソフトウェア部品データ１３５およびソフトウェア部品実行順序データ１３６は、メモリ６１へと記憶される。マイクロプロセッサ６２は、メモリ６１に記憶されたソフトウェアに基づいて、入出力デバイスである行き先階ボタンが押されたことを検知する行き先階ボタンセンサ９１と、ドアへの挟み込みを検知するドア安全センサ９２と、ドアの開閉をおこなうドア駆動モータ９３と、かごの上下動をおこなうかご駆動モータ９４とを操作し、エレベータかご７の昇降制御を行う。

【００６３】

以上述べたように、本実施の形態を用いることにより、大規模・複雑化したソースコー

ドであっても、変数の代入・参照関係を分析し、変数の依存関係に基づいたクラスタ毎にソフトウェア部品を抽出することが可能である。また、変数の依存関係を用いているため、熟練者の経験に依らず容易にソフトウェア部品を作成することができる。

【００６４】

また、組込みシステムをエレベータとした場合、ある特定の機能単位に依らないソフトウェア部分集合であるクラスタは、変数として少なくとも、ドアゾーンスイッチ，ファイナルリミットスイッチ，メンテナンスリミットスイッチ，方向リミットスイッチ，または終端階強制還俗スイッチのいずれかの状態を示す情報を含むことが望ましい。つまり、昇降路内に必要とされる定位置として決定することが好適であり、国際安全規格への対応，エレベータの構成部品の機械部品から電子部品への置き換えに依る高性能化，低コスト化を行う電子安全化への対応，安全性の説明責任，並びに遠隔アップデート等の点で望ましい。

【００６５】

なお、メンテナンスリミットスイッチとは、保守員が安全に作業を実施するための頂部隙間を確保する位置を明示するスイッチであり、方向リミットスイッチとは、乗りかごが終端階に到達したことを検出するスイッチであり、終端階強制減速スイッチとは、乗りかごが何らかの原因で過速状態になったときに定格速度以下まで強制減速させてバッファに衝突させるための位置を検出するスイッチである。

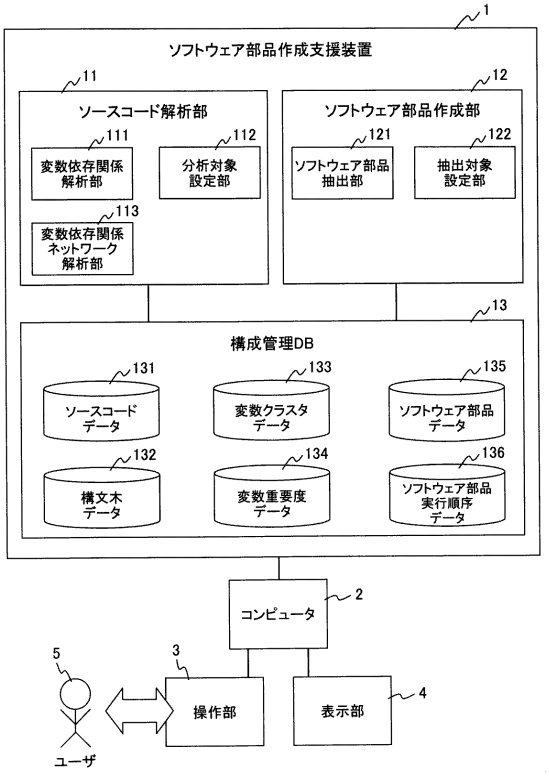
【符号の説明】

【００６６】

１：ソフトウェア部品作成支援装置、２：コンピュータ、３：操作部、４：表示部、５：ユーザ、１１：ソースコード解析部、１１１：変数依存関係解析部、１１２：分析対象設定部、１１３：変数依存関係ネットワーク解析部、１２：ソフトウェア部品作成部、１２１：ソフトウェア部品抽出部、１２２：抽出対象設定部、１３：構成管理ＤＢ、１３１：ソースコードデータ、１３２：構文木データ、１３３：変数クラスタデータ、１３４：変数重要度データ、１３５：ソフトウェア部品データ、１３６：ソフトウェア部品実行順序データ。

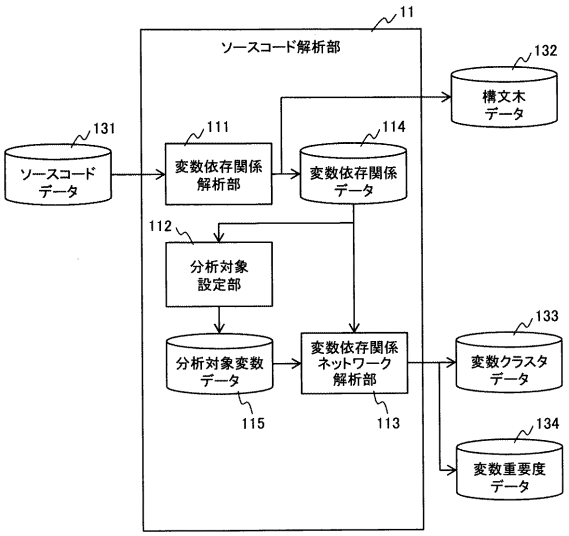
【図１】

図 1



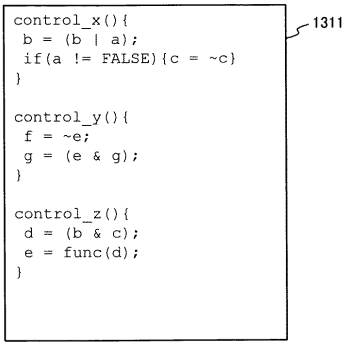
【図２】

図 2



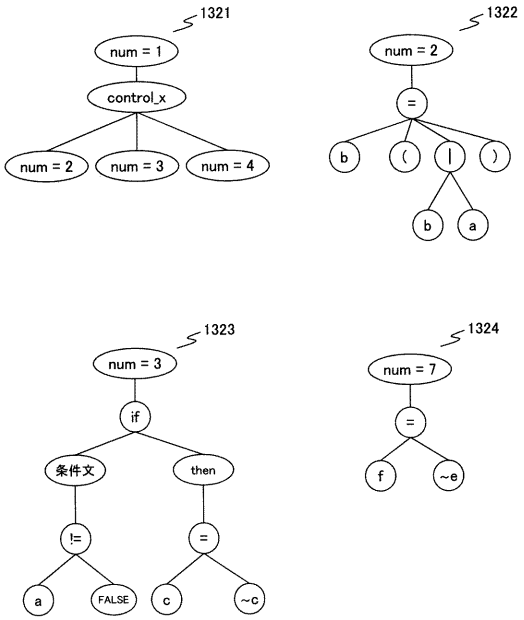
【図３】

図 3



【図４】

図 4



【図５】

図 5

リンクID	変数名（始点）	変数名（終点）
1	a	b
2	a	c
3	b	d
4	c	d
5	d	e
6	e	f
7	e	g

1141

【図６】

図 6

始点＼終点	a	b	c	d	e	f	g
a	0	1	1	0	0	0	0
b	1	0	0	1	0	0	0
c	1	0	0	1	0	0	0
d	0	1	1	0	1	0	0
e	0	0	0	1	0	1	1
f	0	0	0	0	1	0	0
g	0	0	0	0	1	0	0

1142

【図７】

図 7

選択	始点＼終点	a	b	c	d	e	f	g
✓	a	0	1	1	0	0	0	0
✓	b	1	0	0	1	0	0	0
✓	c	1	0	0	1	0	0	0
✓	d	0	1	1	0	1	0	0
✓	e	0	0	0	1	0	1	1
✓	f	0	0	0	0	1	0	0
✓	g	0	0	0	0	1	0	0

1121

【図８】

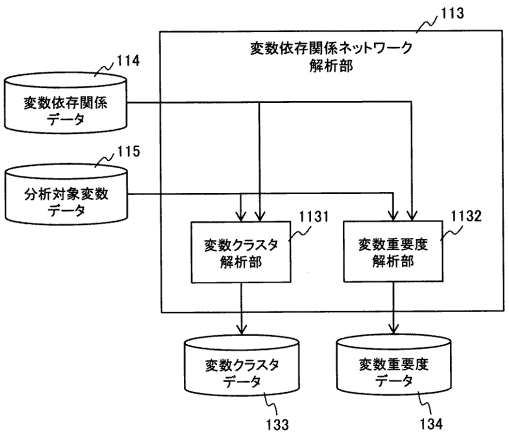
図 8

リンクID	変数名（始点）	変数名（終点）
1	a	b
2	a	c
3	b	d
4	c	d
5	d	e
6	e	f
7	e	g

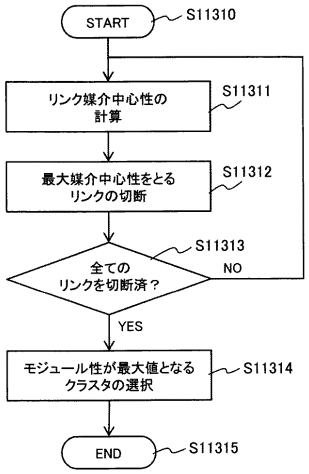
1151

【図９】

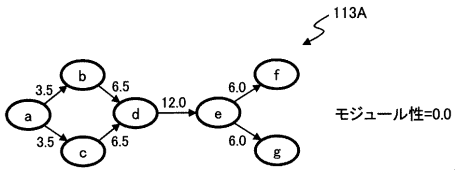
図 9



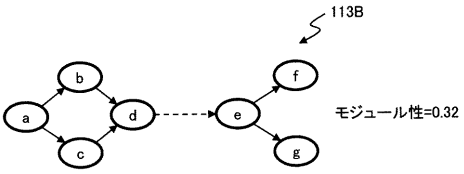
【図１０】
図 10



【図１１】
図 11



【図１２】
図 12



【図１３】
図 13

1331

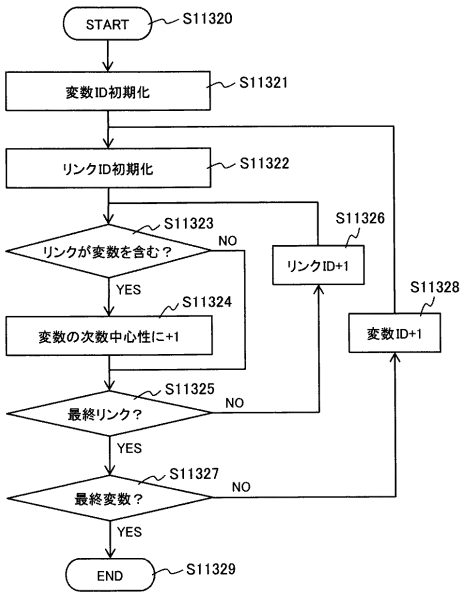
クラスタ名	変数名
クラスタ1	a, b, c, d
クラスタ2	e, f, g

【図１４】
図 14

1332

リンクID	始点	終点	リンク変数
101	クラスタ1	クラスタ2	d

【図１５】
図 15



【図１６】
図 16

1341

変数名	重要度（次数中心性）
a	2
b	2
c	2
d	3
e	3
f	1
g	1

【図１７】

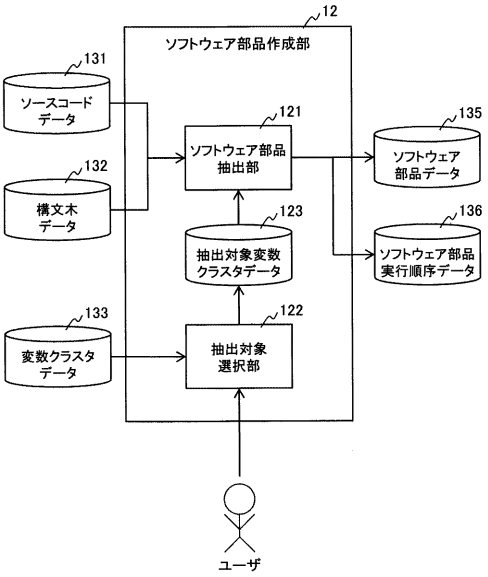
図 17

1342

変数名	重要度 (Rank)
a	0.14
b	0.14
c	0.14
d	0.20
e	0.22
f	0.08
g	0.08

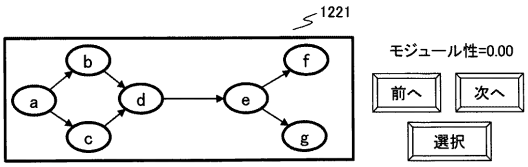
【図１８】

図 18



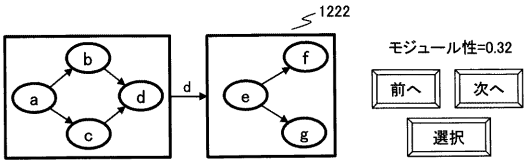
【図１９】

図 19



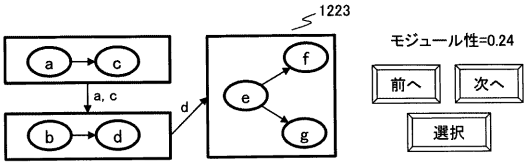
【図２０】

図 20



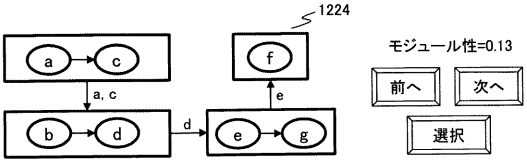
【図２１】

図 21



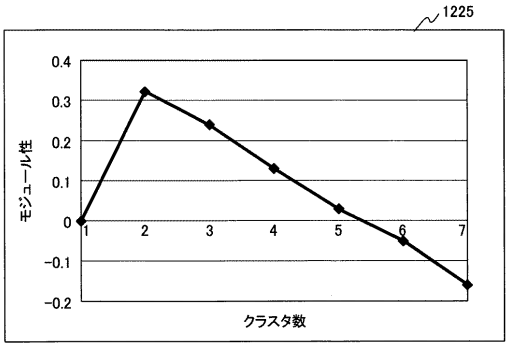
【図２２】

図 22



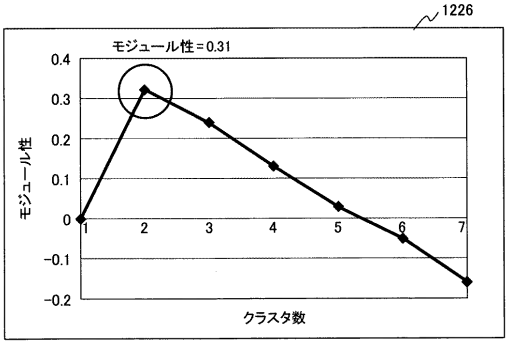
【図２３】

図 23



【図２４】

図 24



【図２５】

図 25

1227

クラスタ数	モジュール性
1	0
2	0.32
3	0.24
4	0.13
5	0.03
6	−0.05
7	−0.16

【図２６】

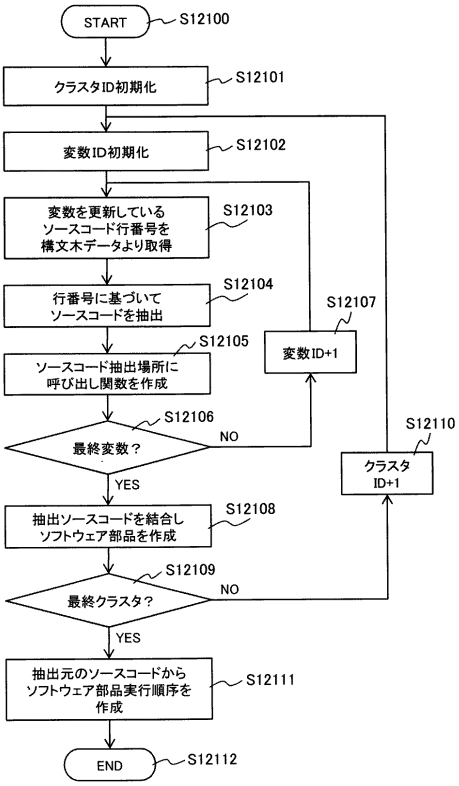
図 26

1231

クラスタID	変数
101	a, b, c, d
102	e, f, g

【図２７】

図 27



【図２８】

図 28

1361

```
// モジュール性 0.32

control_x(){
 b_update();
 c_update();
}

control_y(){
 f_update();
 g_update();
}

control_z(){
 d_update();
 e_update();
}
```

1351

```
// モジュール性 0.32

b_update(){
 b = (b | a);
}
c_update(){
 if(a != FALSE){c = ~c}
}
d_update(){
 d = (b & c);
}
```

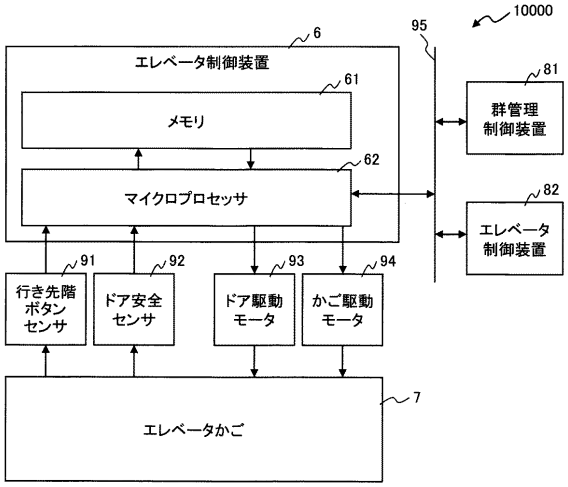
1352

```
// モジュール性 0.32

e_update(){
 e = func(d);
}
f_update(){
 f = ~e;
}
g_update(){
 g = (e & g);
}
```

【図２９】

図 29


10000
95
6
エレベータ制御装置
61
メモリ
62
マイクロプロセッサ
81
群管理
制御装置
82
エレベータ
制御装置
91
行き先階
ボタン
センサ
92
ドア安全
センサ
93
ドア駆動
モータ
94
かご駆動
モータ
7
エレベータかご

フロントページの続き

(72)発明者　杉山　洋平
茨城県ひたちなか市市毛１０７０番地　株式会社日立製作所都市開発システム社内
(72)発明者　会田　敬一
茨城県ひたちなか市市毛１０７０番地　株式会社日立製作所都市開発システム社内

Ｆターム(参考) 5B376 BC12　BC80　FA11