

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3625084号
(P3625084)

(45) 発行日 平成17年3月2日(2005.3.2)

(24) 登録日 平成16年12月10日(2004.12.10)

(51) Int. Cl.⁷

G06F 9/06

F I

G06F 9/06 530W

請求項の数 4 (全 18 頁)

(21) 出願番号	特願平7-89217	(73) 特許権者	000006666
(22) 出願日	平成7年4月14日(1995.4.14)		株式会社山武
(65) 公開番号	特開平8-286901		東京都渋谷区渋谷2丁目12番19号
(43) 公開日	平成8年11月1日(1996.11.1)	(74) 代理人	100064621
審査請求日	平成10年5月29日(1998.5.29)		弁理士 山川 政樹
審査番号	不服2003-14346(P2003-14346/J1)	(74) 代理人	100067138
審査請求日	平成15年7月24日(2003.7.24)		弁理士 黒川 弘朗
		(74) 代理人	100076392
			弁理士 紺野 正幸
		(74) 代理人	100081743
			弁理士 西山 修
		(74) 代理人	100098394
			弁理士 山川 茂樹

最終頁に続く

(54) 【発明の名称】 ソフトウェア部品の再利用システム

(57) 【特許請求の範囲】

【請求項1】

制御対象のプロセス装置とそれを制御するための制御機器との関係を各ノードとこれらノード間をリンクにより接続した状態を表示可能に示すソフトウェアの設計情報を設計する設計部、前記設計部により設計された設計情報を格納する設計情報格納部を有する設計情報構築部と、

前記設計情報を構築するための部品が複数の構成要素から階層構造となり、下位の階層構造をした構成要素からなる部品を包含して格納された部品格納部、前記部品格納部に格納されている部品を構成する各層を検索抽出する部品検索部、前記部品格納部に階層構造とした部品を登録する部品登録部を有する設計情報管理部と

から構成され、

前記部品格納部に格納されて指定された部品の階層構造の状態を表示出力し、この表示出力され階層構造における指定された部品の階層構造を再利用部品として選択し、この選択した再利用部品を前記設計部における設計情報の設計に再利用することを特徴とするソフトウェア部品の再利用システム。

【請求項2】

請求項1記載のソフトウェア部品の再利用システムにおいて、

前記部品格納部に格納される部品にその内容の説明情報と他との関係情報とを付帯させたことを特徴とするソフトウェア部品の再利用システム。

【請求項3】

10

20

請求項 1 または 2 記載のソフトウェア部品の再利用システムにおいて、前記部品格納部に格納されている部品のシミュレーションを実行する部品動作シミュレーション実行部を有することを特徴とするソフトウェア部品の再利用システム。

【請求項 4】

請求項 1 から 3 のいずれか 1 項に記載のソフトウェア部品の再利用システムにおいて、前記部品格納部に格納されている部品には、この部品の属性を示す属性キーが付与され、前記部品検索部は、前記属性キーに基づいて前記部品格納部に格納されている部品を構成する各層を検索抽出することを特徴とするソフトウェア部品の再利用システム。

【発明の詳細な説明】

【0001】

10

【産業上の利用分野】

この発明は、ソフトウェアの設計およびその設計情報を再利用するソフトウェア部品の再利用システムに関する。

【0002】

【従来の技術】

ソフトウェア（アプリケーションソフト）を作成するとき、まず初めにソフトウェアの設計情報を構築し、これに則ってそのソフトウェアを構成していく。

従来より、この設計情報を構成するプログラム（部品）の再利用ということで様々な工夫がなされてきたが、そのアプローチには大きく 2 つの形態がある。

まず、第 1 として、標準的な再利用部品を予め作成し、個々のソフトウェア作成時にこれ

20

を利用していくという形態がある。次に、第 2 として、既にあるソフトウェア作成時に用いた設計情報の中から、再利用者が再利用できそうな部分を選択して、必要に応じてカスタマイズを施した上で利用する形態である。

【0003】

【発明が解決しようとする課題】

従来は以上のようななされていたので、以下に示すような問題点があった。

まず、上述した第 1 の形態では、ライブラリといわれる標準部品を予め作成しておくわけであるが、この仕様や使い方が、再利用するときの利用目的に合致しないという問題があった。

30

つまり、ライブラリ作成時の仕様と、再利用時の利用目的との間に、多少なりとも常に差異が生じており、この使い方や仕様などを理解するのに手間がかかるという問題である。

【0004】

予め用意しておくライブラリが、数値計算用の比較的汎用的で規模が小さい場合は、上述したような問題はあまりないが、例えば、プロセスコントローラ用のアプリケーションプログラムのためのソフトウェア設計情報などの、特化した大きなものになると、その内容が複雑で理解が難しいものとなる。

そして、カスタマイズして利用するにも手間がかかり、カスタマイズにより予想でき得ないバグが発生するなどの問題もあった。

【0005】

40

一方、第 2 の形態では、再利用するとき、既存の設計情報を構成する部品の中から利用目的に合致する部分を選び出して利用する程度であり、元のソフトウェア設計情報全体に対する各部分の果たす役割が分かっていないと、どの部分が目的に合致するのかが分からないという問題があった。

また、元のソフトウェア設計情報全体の中で、どこからどこまでの部分が所望とする機能を有しているのかが分からないという問題があった。

【0006】

他方、第 1 の形態において、予め用意して有るライブラリを利用するときは、ライブラリの中の各設計情報に検索子として属性キーを割り当てたり、その設計情報を構成する各部品に対して検索子を割り当てて、それぞれ 1 次元的に単純な検索をすることはできたが、

50

設計情報と部品という2次元に対して検索することができなかった。

【0007】

この発明は、以上のような問題点を解消するためになされたものであり、ソフトウェア設計情報の作成者の意図と、それを構成する各部品の中より利用できるものは再利用しようとする利用者の利用目的とが異なっているにもかかわらず、再利用できるようにすることを目的とする。

また、所望とする設計情報およびその各部品が効率よく検索できるようにすることを目的とする。

そして、所望とする各部品の仕様や構造およびその動作に関して、再利用者が容易に理解できるようにすることを目的とする。

10

【0008】

【課題を解決するための手段】

この発明のソフトウェア部品の再利用システムは、制御対象のプロセス装置とそれを制御するための制御機器との関係を各ノードとこれらノード間をリンクにより接続した状態で表示可能に示すソフトウェアの設計情報を設計する設計部、設計部により設計された設計情報を格納する設計情報格納部を有する設計情報構築部と、設計情報を構築するための部品が複数の構成要素から階層構造となり、下位の階層構造をした構成要素からなる部品を包含して格納された部品格納部、部品格納部に格納されている部品を構成する各層を検索抽出する部品検索部、部品格納部に階層構造とした部品を登録する部品登録部を有する設計情報管理部とから構成され、部品格納部に格納されて指定された部品の階層構造の状態
を表示出力し、この表示出力され階層構造における指定された部品の階層構造を再利用部品として選択し、この選択した再利用部品を設計部における設計情報の設計に再利用しようとしたものである。

20

また、部品格納部に格納される部品にその内容の説明情報と他との関係情報とを付帯させたことを特徴とする。

そして、部品格納部に格納されている部品のシミュレーションを実行する部品動作シミュレーション実行部を有することを特徴とする。

また、部品格納部に格納されている部品には、この部品の属性を示す属性キーが付与され、部品検索部は、属性キーに基づいて部品格納部に格納されている部品を構成する各層を検索抽出する。

30

【0009】

【作用】

階層状態を確認しながら部品を検索抽出し、これをソフトウェアの設計に再利用する。

また、部品に関する情報が得られる。

そして、部品の動作状態が確認できる。

【0010】

【実施例】

以下この発明の1実施例を図を参照して説明する。

図1は、この発明の1実施例であるソフトウェア部品の再利用システムを実現するための、再利用管理装置の構成を示す構成図である。

40

同図において、1aはCRT、1bはキーボード、2はマウス、3は装置本体であり、この装置本体3は、大きく分けて、設計情報管理部4と設計情報構築部5から構成される。設計情報管理部4は、部品検索部6と部品動作シミュレーション実行部7と部品登録部8と部品格納部9とから構成されている。

また、設計情報構築部5は、プログラムおよび仕様書や説明書などの付帯情報からなる設計情報を開発するための開発環境を提供する設計部10と、設計情報格納部11とから構成されている。

【0011】

ソフトウェア開発者は、CRT1a、キーボード1b、もしくはマウス2を媒介として装置本体3にアクセスする。

50

この装置本体 3 には設計情報構築部 5 が配置され、これを用いてソフトウェアの開発設計を行う。

設計部 10 により、新規のソフトウェア設計情報の作成や、既存のソフトウェア設計情報の変更修正を行うことができる。

設計部 10 で作成 / 変更された、又はその作業の途中のソフトウェア設計情報は、設計情報格納部 11 に格納することができる。

【 0 0 1 2 】

また設計情報格納部 11 に格納されたソフトウェア設計情報を設計部 10 に呼び出して、さらに編集を加えることも可能である。

ソフトウェア開発者は、前述のように設計情報構築部 5 のみを用いて、全てを新規にソフトウェア設計情報を作成することもできるが、生産効率および品質を考慮すると、既存の実績のあるソフトウェア設計情報の再利用を交えて開発することが望ましい。

【 0 0 1 3 】

これを支援するのが設計情報管理部 4 である。

設計情報管理部 4 の構成要素である部品格納部 9 には、既存のソフトウェア設計情報や、そこから汎用性や再利用性を考慮して部分的に抽出した標準部品などの種々のソフトウェア設計情報の部品を記憶している。

これら、部品格納部 9 に登録されたソフトウェア設計情報を、以降ソフトウェア部品と称し、開発中のソフトウェア設計情報と区別する。

ソフトウェア開発者は、部品検査部 6 を介してそれらのソフトウェア部品から利用可能なものを選択し、設計部 10 で開発中のソフトウェア設計情報の中に取り込んで利用する。

【 0 0 1 4 】

また、ソフトウェア部品を開発中のソフトウェア設計情報の中に取り込む前に、そのソフトウェア部品の使い方を把握できるように、その動特性をシミュレートするための、部品動作シミュレーション実行部 7 が設けられている。

さらに設計情報構築部 5 で開発したソフトウェア設計情報を、ソフトウェア部品として登録するため、部品登録部 8 が用意されている。

【 0 0 1 5 】

この部品登録部 8 では、設計情報構築部 5 で開発完了したソフトウェア設計情報全体をそのまま登録することもできる。

また、その中から汎用性や再利用性に注目して一部分を抽出し、必要ならば修正を施した上で登録することもできる。

登録されたソフトウェア部品は、部品格納部 9 に格納され、以降のソフトウェア設計情報の開発時に利用されることとなる。

【 0 0 1 6 】

なお、本実施例では、設計情報構築部 5 と設計情報管理部 4 が同一装置上に配置されたスタンドアロン構成をとっているが、これに限るものではない。

設計情報構築部 5 がクライアントの装置上に配置され、設計情報管理部 4 がサーバの装置上に存在するクライアント・サーバ方式による分散環境下にも適用できることはいうまでもない。

【 0 0 1 7 】

以下、まず、ソフトウェア部品に関して詳細に説明する。

本発明におけるソフトウェア部品は、その内部構造として階層的な木構造を有している。

図 2 は、この実施例におけるソフトウェア部品の構成要素の種類およびその構成を示す構成図である。

同図において、201 はソフトウェア部品内部の階層構造を形成するための基本部品である階層パーツ、202 はソフトウェア部品内部のプログラムモジュールを形成するための基本部品であるプログラムパーツ、203 はプログラムパーツの制御対象（計装設計においては、分散制御システムの制御部や、プラントの装置にあたる）をシミュレートする部品である機能パーツである。

10

20

30

40

50

【0018】

また、204はアイコン、205はそのパーツの詳細構造をネットワーク表現で記述するための枠組みであるネットワーク記述シート、206はそのパーツの詳細動作をフロー表現（フローチャートやSFCなど）で記述定義するための枠組みであるフロー記述シート、207はテキスト情報や絵情報を記述する枠組みであり、その階層パーツの機能や目的などの機能仕様や設計情報を、図表現や自然言語を用いて記述するために用いられるドキュメント記述シートである。

アイコン204は、そのパーツが他のパーツのネットワーク記述シート205、フロー記述シート206、ドキュメント記述シート207上で記述された場合のCRT1a（図1）上での表示形態、もしくはその表示情報である。アイコンの形状は、階層パーツの使い方や好みによって自由に変更成形することが可能である。

10

【0019】

そして、208は音声情報を格納する枠組みであり、その階層パーツに関する機能仕様、設計情報やコメントなどの情報を音声情報として格納する音声情報格納部、209はソフトウェア部品の再利用時に、再利用する部分を検索表示するときの検索キーを格納する検索用属性格納部、210は外部変数を格納している外部変数I/F、211はそのプログラムパーツの機能動作を定義するためのプログラムをテキストチャルなプログラム言語で記述するための枠組みであるプログラム記述シート、212はシミュレーション実行時（詳細は後述）に制御対象のを振る舞いをシミュレートするためのプログラムを格納するシミュレーションプログラム格納部である。

20

外部変数I/F210は、当該パーツのプログラム記述シート211上に記述されたプログラムが参照している、制御対象となるプラント機器や分散制御システムの制御部などの外部変数を格納している。

これは、シミュレーション実現のために利用されるものである。

【0020】

なお、CRT1a（図1）上に表示されるアイコン204をマウス2（図1）などで指定することで、対応する階層パーツのシート（ネットワーク記述シート205、フロー記述シート206または、ドキュメント記述シート207）を開き表示することができる。同様にアイコン204と音声情報格納部208との関係は、親パーツのシートに対応しているアイコン204をマウス2などで指定することで、音声情報格納部208に音声を記憶させたり、既に記憶されている音声を再生することができる。

30

【0021】

以下、階層パーツによる階層構造の形成に関して説明する。

図3は、階層構造を示す説明図であり、301は1個の階層パーツであり、その内部には、アイコン302とネットワーク記述シート303とフロー記述シート304およびドキュメント記述シート305が存在する。

また、子側の階層の階層パーツ301が親側の階層パーツのシート306に置かれている様子が表されている。

このシート306には、ネットワーク記述シート303、フロー記述シート304、ドキュメント記述シート305のいずれかが置かれているものである。

40

【0022】

また、シート306には他の子となる階層パーツのアイコン307も置かれており、このアイコン307をもつ階層パーツは、階層パーツ301から見て同階層にあるという。一方の階層パーツの保持するシート306に、他方の階層パーツ301のアイコン302を貼り付けることで、前者のパーツが後者の階層パーツ301を所有したことを表す。この所有した方を親または親階層パーツといい、所有された方を子または子階層パーツという。

【0023】

次に、前述の階層パーツ、プログラムパーツ、機能パーツを用いて、上述したソフトウェア部品の内部構造を説明する。

50

図4は、この発明におけるソフトウェア部品の構成を示す構成図であり、計装ソフトウェア部品の一例を示したものである。

同図(a)はソフトウェア部品の内部構造のイメージを示した階層構成を示し、(b)は階層構成のみを簡略化して示したツリー構成を示す構成図である。

【0024】

まず初めに、図4(a)の内部構造のイメージに関して説明する。

各ソフトウェア部品は、ルート階層パーツ401を頂上とし、階層パーツの親子関係により階層構造を形成している。

図中のソフトウェア部品は、プロセス制御を目的とする計装ソフトウェア部品であり、次のような階層構造を形成している。

10

【0025】

ルート階層パーツ401は1枚のネットワーク記述シート402をもつ。

そのネットワーク記述シート402には、制御対象のプロセス装置とそれを制御するための制御機器との関係を記した「コントロールフロー図」が記述されている。

すなわち、プロセス装置や制御機器をノードとし、それら各ノード間をリンクにより接続した状態を示すものである。

ここにはいくつかの機能パーツ406と、階層パーツを示す3つのアイコン403~405がノードとして配置されており、それぞれ「シーケンス」、「インタロック」、「比率」と名付けられている。

【0026】

20

そして、ネットワーク記述シート402における「シーケンス」を示すアイコン403には、その「シーケンス」のプログラムのフローチャートが記述されたフロー記述シート407が下層に付随している。

このフロー記述シート407には、仕込み工程408、攪拌工程409という名の2つの階層パーツと、払い出し工程410という名の付いたプログラムパーツのアイコンが置かれている。

さらに、このフロー記述シート407の払い出し工程410の下層には、この詳細なプログラムフローを記述したプログラム記述シート411が付随している。

【0027】

次に図4(b)に記述されているツリー構成に関して説明する。

30

これは、図4(a)のソフトウェア部品の内部構造に対応させて表現したものであり、ルート階層パーツ401にノード412が対応する。

また、ルート階層パーツ401を構成するネットワーク記述シート402中のアイコン403, 404, 405で示されている下位の階層パーツに「比率」、「インタロック」、「シーケンス」などを示すノード413, 414, 415が対応する。

そして、そのまた下位のフロー記述シート407に、「仕込み工程」、「攪拌工程」、「払い出し工程」などを示すノード416, 417, 418が対応する

【0028】

上述したような階層構造のデータとなっているソフトウェア部品を再利用しようとするときは、例えば、ノード415を選択することで、その子であるノード416, 417, 418・・・が芋蔓式に取り出される。

40

ノード416, 417, 418はノード415の子であり、これらはノード415をルートとする「部分木」の一部である。

ただし、階層構造の一部の「部分木」の中で一部の階層パーツを除外した状態で再利用する単位としたり、複数の「部分木」をまとめて1つの再利用する単位とすることもできる。

【0029】

前述のように本発明においては、再利用を目的に登録したソフトウェア部品の単位でなく、そのソフトウェア部品の構成要素である(階層パーツによる)部分木の単位で再利用が可能となっている。

50

したがって、ソフトウェア開発において、再利用しうるソフトウェア部品を探すためには、まず利用可能なソフトウェア部品を探し、次にそのソフトウェア部品の中から利用できそうな階層パーツもしくはプログラムパーツ（部分木のルート）を探す、と言う2段階の検索が一般的には必要となる。

このような2段階の検索を、風潰しに行うのではなく効果的にこなすというのが「水平/垂直ブラウジング機能」である。

【0030】

まずその原理を図5を用いて説明する。

図5は、3個のソフトウェア部品A501, B502, C503から利用可能な部分木のルート階層パーツを検索する場合の例が示されている。

同図において、504~512はソフトウェア部品A501, B502, C503の部分木を構成する階層パーツ、513はソフトウェア部品管理テーブルである。

このソフトウェア部品管理テーブル513は、各ソフトウェア部品A501~C503の部品名とルートとなる階層パーツ504~506へのポインタを記憶するテーブルである。

【0031】

図5では、ソフトウェア部品A501の階層パーツ510と、ソフトウェア部品B502の階層パーツ511と、ソフトウェア部品C503の階層パーツ512は同一または類似機能を持つ階層パーツである。

以下ではこのような3つのソフトウェア部品A501, B502, C503から目的の階層パーツ512を探すまでのプロセスを例にとり、この「水平/垂直ブラウジング機能」を説明する。

【0032】

このブラウジングは、次の3つの機能からなる。

まず、第1として、ソフトウェア部品レベルのブラウジングとして、どのようなソフトウェア部品が登録されているのか確認すると共に、どのソフトウェア部品から探し初めるのかを決めるための機能がある。

これは、ソフトウェア部品管理テーブル513を参照し、どのソフトウェア部品から、その内部構造を検索表示するかを決定するものである。

【0033】

実際の内部処理としては、ソフトウェア部品管理テーブル513内の部品名をどれか選択することで、そのソフトウェア部品のルートの階層パーツが求まる。

この時、ルートの階層パーツが保持する、図2に示すような、ネットワーク記述シート205、フロー記述シート206、ドキュメント記述シート207などの各種シート情報を表示したり、音声情報格納部208が格納している音声メッセージなどを出力したりすることで、これらを参考情報とすることができる。

【0034】

次に、第2の機能として、ソフトウェア部品の内部階層構造に基づくブラウジング（垂直ブラウジング）がある。

これは、ある特定のソフトウェア部品の内部階層構造に基づき、階層パーツ（またはプログラムパーツ）が保持している、図2に示すネットワーク記述シート205、フロー記述シート206、ドキュメント記述シート207、および音声情報格納部208に格納されている音声情報などの情報を参照しながら、階層構造内を上下することで再利用すべき部分木を絞り込んでいく機能である。

正確には、その部分木のルート階層パーツを検索していく機能である。

【0035】

図5をもとに、その検索方法を簡単に説明する。

階層パーツが保持している情報を基に判断して、ソフトウェア部品A501が再利用したい機能を持っていると判断した場合、そのルートの階層パーツ504以下の各階層パーツ507および階層パーツ508の保持情報を順次チェックする。

10

20

30

40

50

階層パーツ507と階層パーツ508の両方も再利用したい機能に関係している場合は、階層パーツ504をルートノードとする部分木(というより全体)を利用すればよいことが分かる。

【0036】

しかし、階層パーツ508が求める機能と何ら関係しない場合、階層パーツ507をルートとする部分木(または、さらにその部分木)を利用すればよいことが分かる。

このプロセスを順次繰り返すことにより、再利用したい機能を持つ必要最小の部分木を絞り込んでいくことができる。

この例において必要最小の部分木は下位層パーツ510のみだと仮定すると、必要最小の部分木のルート階層パーツの検索は、階層パーツ504、階層パーツ507、階層パーツ509を経て階層パーツ510に辿り着く。

10

【0037】

そして、第3として水平ブラウジング(同一/類似機能を持つ階層パーツの検索)がある。

これは、あるソフトウェア部品の中で利用可能な階層パーツ(部分木)を発見した後、他のソフトウェア部品の中に、それと同様の機能を有するものが有るかを確認し、あればそれを芋蔓式に取り出そうというのが水平ブラウジング機能である。

【0038】

この水平ブラウジングを実現するために、各階層パーツおよびプログラムパーツに属性キーが付与されている。これは、例えば、第2図における検索用属性格納部209に格納するようになればよい。

20

属性キーは、ソフトウェア部品を作成した段階で設定、変更できるものであり、また、各パーツに付与する属性キーの個数は、必ずしも1個とは限らなく、複数の属性キーを所有することも可能である。

その属性キーをもとに、他のソフトウェア部品(または該当ソフトウェア部品)の中に同一の属性キーを持つ階層パーツまたはプログラムパーツが、あるかどうかを検索表示するのが本機能である。

【0039】

図5を用いてこれを説明すると、前述のように、階層パーツ510、511、512は同一の属性キーを保持しているとしている。

30

従って、前述の垂直ブラウジングにより階層パーツ510が捜し当てられていれば、芋づる式に同一の属性キーを持つ階層パーツ511、512を引き出すことができる。

さらに、各階層パーツの保持情報を参照することで、階層パーツ510、511、512の中で最も目的の機能に近い階層パーツを選ぶことができる。

図の例では、太い線の矢印を辿ることにより、階層パーツ512が選択されたことが示されている。

【0040】

図5を用いた上記の説明では、第1から第3までの機能を順に使って検索を行なったが、必ずしもこの順序で検索する必要はない。

例えば、再利用したい階層パーツ(部分木のルート階層パーツ)が保持している属性キーに予め心当りがあれば、上述の第3の水平ブラウジングから検索を開始して手間を省くことが可能である。

40

【0041】

図6は、上述した水平/垂直ブラウジングを行うための、マン・マシン・インタフェースを示す画面図である。すなわち、図1のCRT1a上の部品検索ウインドウ内に表示される画面状態を示すものである。

図6において、601は部品選択表示部、602は部品内部階層表示部、603はパーツ保持情報表示部、604は部品表示部、605はカテゴリ表示部、606、607、608はノード識別子リスト表示部、609は上位階層シフトボタン、610は下位階層シフトボタン、611はシート情報表示部、612はドキュメント情報表示部、613はアイ

50

コン表示部、614は音声ガイダンスコントロールボックスである。

【0042】

まず、ブラウジングを開始するソフトウェア部品の選択を、部品選択表示部601で行ない、次に前述の水平/垂直ブラウジングは部品内部階層表示部602で行なう。

そこで、図2に示したような、ソフトウェア部品の特定の階層パーツ201やプログラムパーツ202を指定すると、パーツ保持情報表示部603において、以下のことがなされる。

すなわち、それらのパーツが保持するネットワーク記述シート情報205、フロー記述シート206、プログラム記述シート211、ドキュメント記述シート207の内容や音声情報格納部208に格納された音声メッセージを再生することができる。

10

【0043】

このように指定した階層パーツ(またはプログラムパーツ)の保持情報を表示することにより、水平/垂直ブラウジングを行なう際に強力な支援が可能となる。

また、この部品内部階層表示部602で再利用しようとする部分木のルート階層パーツが選択できた場合、その階層パーツ表示を画面上のポインタを移動操作することによるドラッグ&ドロップ等により、その階層パーツをルートとする部分木を、他の設計などの作業が行われているウインドウに取り込むこともできる。

この設計作業は、設計部10(図1)でなされるものである。

【0044】

一方、部品内部階層表示部602は、カテゴリ表示部605と、n個のノード識別子リスト表示部606~608、上位階層シフトボタン609、下位階層シフトボタン610で構成されている。

20

各ノード識別子リスト表示部606~608には、ソフトウェア部品内部の階層構造における同一階層に属す階層パーツおよびプログラムパーツを拾い集め、名前やID番号など、そのパーツを識別しうる属性である識別子がリスト形式で表示される。

また、各ノード識別子リスト表示部606~608間には親子関係が定義されている。

【0045】

親のノード識別子リスト表示部で表示されている階層パーツのうち1つを選択すると、それに連動して、子供のノード識別子リスト表示部に該当パーツの子供の階層パーツの識別子をリスト形式で表示する。

30

ちなみに、「NODE(1)」で示されるノード識別子リスト表示部606に最上位階層が表示され、その次の子供の階層は「NODE(2)」で示されるノード識別子リスト表示部607に表示される。

同様にして孫、ひ孫、と続き、n番目の階層は「NODE(n)」で示されるノード識別子リスト表示部608に表示される。

【0046】

以上のようにすることで、ノード識別子リスト表示部606で選択した階層パーツをルートとする部分木を最大、深さn階層まで表示することができる。

また、このノード識別子リスト表示部606~608では、最大n階層までの部分木しか同時に表示できない。

40

これより上位または下位のノードを表示するために、上位階層シフトボタン609、下位階層シフトボタン610が用意されている。

これらのボタンを押すことで、それまで表示されていた情報を1つ右の、または1つの左のノード識別子リスト表示部に順にずらし、1階層上または1階層下のノードを表示することができる。

【0047】

ノード識別子リスト表示部の個数を3個と限定した図7を用い、以上の利用方法の詳細説明をする。

同図において、701は部品選択表示部、702は部品内部階層表示部、703は部品表示部、704はカテゴリ表示部、705~707はノード識別子リスト表示部、708は

50

上位階層シフトボタン、709は下位階層シフトボタンである。

また、710は階層構造となったソフトウェア部品のデータベース(データ構造)の一部を示すソフトウェア部品A、711は同様にソフトウェア部品Bである。

【0048】

ここで、ソフトウェア部品A710、B711は、識別子と属性キーとを付された階層パーツの階層構造となっている。

例えば、ソフトウェア部品A710のルート階層パーツには、識別子として「部品1」が付され、属性キーとして「CAT_A」が付されている。

部品表示部703には、「ソフトウェア部品A」と「ソフトウェア部品B」が表示され、カテゴリ表示部704には属性キーが表示される。

そして、識別子のうち、ルートの階層パーツを親として孫までの階層パーツが、部品内部階層表示部702に表示される。ここでは、ノード識別子リスト表示部705~707の3つにノード識別子リストが表示されていく。

【0049】

そして、部品表示部703の中で、「ソフトウェア部品B」を選択し、カテゴリ表示部704の中で「CAT_B」を選択することで、ソフトウェア部品B711の中のルートの階層パーツが選択され、表示状態が変更されて反転表示状態となる。

このことにより、ノード識別子リスト表示部705にその識別子「部品2」が表示され、以下のノード識別子リスト表示部706に「部品22」、ノード識別子リスト表示部706に「部品23」が表示される。

【0050】

また、「部品2」を親とする「部品21」以外の子の「部品22」がノード識別子リスト表示部706の2段目に表示され、「部品23」以外の孫の「部品24」がノード識別子リスト表示部707の2段目に表示される。

ここで、所望の階層パーツとして、「部品2」、「部品21」、「部品23」を選択することで、これらの表示状態も変更され、例えば反転表示される。

【0051】

ところで、部品内部階層表示部702では、親、子、孫までのつながりのノード識別子を一度に表示するようにしている。

このため、例えば、図7に示す状態では、ノード識別子リスト表示部707の最上段に表示されている「部品23」で示される階層パーツより下の階層パーツに関しては分からない。

ここで、下位階層シフトボタン709を選択することで、部品内部階層表示部702に表示される識別子の領域を変更すれば、例えば、「部品21」がノード識別子リスト表示部705に表示され、「部品23」がノード識別子リスト表示部806に表示され、「部品25」がノード識別子リスト表示部707に表示された状態とすることもできる(図示せず)。

【0052】

以上のような操作により、ソフトウェア部品の内部階層構造を形成する階層パーツを隈無く検索することができる。

図7では、部品内部階層表示部702を、3個のノード識別子リスト表示部705~707で構成していたが、これに限るものではない。

ノード識別子リスト表示部を4個以上の複数としても良いことは、いうまでもない。

【0053】

なお、上述では、部品選択表示部701の部品表示部703の中よりソフトウェア部品を選んでいくようにしたが、これに限るものではない。

カテゴリ表示部704を用い、カテゴリによりソフトウェア部品を選ぶようにすることもできる。

例えば、カテゴリ表示部704の中で、「CAT_D」を選択すると、この属性キーをもつものは、図7(b)に示したように、ソフトウェア部品B711の中では、識別子が「

10

20

30

40

50

部品 2 1」と「部品 2 6」であるので、ノード識別子リスト表示部 7 0 5 の最上段には「部品 2 1」が表示される。

【 0 0 5 4 】

そして、ノード識別子リスト表示部 7 0 6 の最上段には「部品 2 3」が表示され、ノード識別子リスト表示部 7 0 7 の最上段には「部品 2 5」が表示され、ノード識別子リスト表示部 7 0 6 の 2 段目には、「部品 2 1」のもう一つの子である「部品 2 4」が表示される。

一方、ノード識別子リスト表示部 7 0 5 の 2 段目には、属性キーが「CAT_D」である「部品 2 6」が表示される。

【 0 0 5 5 】

ところで、前述したように、階層パーツの検索は、各階層パーツが保持する属性キーによって行なう。

各階層パーツは同時に複数の属性キーを持つことができるため、検索時に複数の検索キーを組み合わせて検索対象を絞り込んでいくことができる。

さらに、属性キーを階層定義することで検索範囲の拡大、縮小が効率良くできるようになる。

【 0 0 5 6 】

属性キーの階層化とは、属性キー間に親子関係が定義され、(親の属性キーによる検索対象範囲) = (親の属性キーを持つ階層パーツ) + (子供の属性キーによる検索対象範囲) という関係が成り立つことをいう。

例えば、図 8 を例にとって説明すると、図の左側に属性キーの木構造が定義されている(実際このようなデータ構造は一般的な 2 分木などのアルゴリズムを用いれば簡単に実現できる)。

【 0 0 5 7 】

このような属性キーの階層構造が形成されているときに、属性キー「CAT_A」を用いて検索すると、それは「CAT_A」を持つ階層パーツ以外に子供の属性キー「CAT_D」と「CAT_X」を持つ階層パーツも合わせて検索されることになり、「部品 1」、「部品 1 3」、「部品 1 6」、「部品 2 1」、「部品 2 6」がその検索結果として得られる。

【 0 0 5 8 】

また、上記では属性キーが静的なものであることを仮定して説明したが、動的に属性キーを作成したり、実際にはどの階層パーツも保持していない抽象的な属性キーも定義することも可能であり、検索対象の範囲の絞り込みの自由度が高くなる。

例えば図 8 において、「CAT_A」と「CAT_B」をそれぞれルートとする属性キーの木があるが、「CAT_B」と「CAT_A」を子とする属性キー「CAT_Z」を定義すると 2 つの属性キーの木をマージした属性キーの木が構成できる。

【 0 0 5 9 】

「CAT_Z」は実際の階層(or プログラムパーツ)に登録されていない属性キーであり、抽象的な属性キーと呼ぶ。

「CAT_Z」の検索範囲は、「CAT_A」および「CAT_B」の検索範囲の和となり、検索の範囲を拡大することができる。

また、ここでは、ルートノードにあたる階層パーツの合成を行なったが、それ以外に「CAT_C」と「CAT_X」を子とする抽象的属性キーも定義しうる。

【 0 0 6 0 】

このように属性キーを階層化することで、まず、検索実行時に動的に抽象属性キーを設定することができ、検索範囲の絞り込みをユーザーが適宜設定でき柔軟性が高くなる。

また、各階層パーツの属性キーに保持するデータが少なくすむ。すなわちより詳細化された属性キーを持てば、その属性キーの上位に当たるよりグローバルな属性キーを同時に持っているのと同じ効果を生じるためである。

10

20

30

40

50

例えば、「CAT_B」、「CAT_C」、「CAT_Y」と「CAT_E」を属性キーとして持たなくても、「CAT_Y」と「CAT_E」さえ持っていれば同様の検索が可能となる。

【0061】

ここで、前述したように部品検索部6(図1)を用いてソフトウェア部品内から階層パーツまたはプログラムパーツを選定しようとしたとき、そのパーツを実際動作させてみることにより、そのパーツの動特性が把握でき、利用者の目的に合っているかどうかを的確に判断できる。

図1に示した、この発明における再利用管理装置では、部品格納部9から部品検索部6を用いて選択したソフトウェア部品に対し、部品動作シミュレーション実行部7を用いることにより、そのソフトウェア部品のシミュレーションが実行できる。

10

【0062】

シミュレーションの実行単位はソフトウェア部品単位ではなく、その内部構成要素である、ある階層パーツをルートとする部分木、またはプログラムパーツの単位でシミュレーションが可能である。

また、階層パーツ201やプログラムパーツ202といった、制御プログラムのソフトウェア部品だけではなく、それらのプログラムの制御対象となる、分散制御システムの制御部やプラント装置を表わす機能パーツ203の動作もシミュレートすることが可能である。

【0063】

図9は図1に示した部品動作シミュレーション実行部7の内部構造を示す構成図であり、901はスケジュール管理手段、902はインタプリタ実現手段、903はシミュレーションプログラム実行手段、904はシミュレーション実行対象管理テーブルである。シミュレーションの対象となるプログラムノードや分散制御システムの制御部などの制御対象が、シミュレーション実行対象管理テーブル904に登録される。

20

【0064】

CRT1a(図1)上に表示される図6や図7に示した画面状態で、シミュレーションする所望のプログラムパーツをマウス2などを用いて指定することにより、そのプログラムパーツがシミュレーション実行対象管理テーブル904に登録される。

また、階層パーツを指定した場合には、その階層パーツをルートとする部分木に含まれる全てのプログラムパーツが登録される。

30

また、シミュレーション対象として登録されたプログラムパーツ202(図2)の外部変数I/F210より、そのプログラムパーツがアクセスしている分散制御システムの制御部などプログラムの制御対象をシミュレートする機能パーツ203(図2)も、シミュレーション実行対象管理テーブル904に登録される。

【0065】

スケジュール管理手段901は、シミュレーション実行対象パーツ管理テーブル904を参照し、シミュレーション実行対象の各プログラムパーツおよび機能パーツを順繰りにシミュレーション実行するためのスケジュール管理を行なう。

そのスケジューリングアルゴリズムでは、シミュレーション実行対象管理テーブル904に登録されているプログラムパーツを順に取り出し、その実行をインタプリタ実現手段902に命令する。

40

そして、インタプリタ実現手段902からの終了通知を受け取ると、次のプログラムパーツを取り出し、前と同様にその実行をインタプリタ実現手段902に命令する。

【0066】

また、最後のプログラムパーツまで実行し終わると、次は機能パーツの実行に移る。

これも、シミュレーション実行対象管理テーブル904から順繰りに、実行する機能パーツを取り出しては、その実行をシミュレーションプログラム実行手段903に命令する。シミュレーションプログラム実行手段903から終了通知を受信すると、次の機能パーツを取り出し、再びその実行をインタプリタ実現手段902に命令する。

50

以下同様にして、最後の機能パーツを実行し終わると、再びプログラムパーツの実行にもどる。

このようなサイクルによりスケジューリングを行なっている。

【0067】

インタプリタ実現手段902は、スケジュール管理手段901より特定プログラムパーツの実行を命令されると、プログラムパーツ202(図2)のプログラム記述シートで記述されたプログラムの1命令列をインタプリティブに実行し、スケジュール管理手段901に実行終了を返す。

また、同一プログラムパーツの実行を命令されたときには、前回実行したステートメントの次のステートメントを実行する。

このように、プログラムパーツの実行は、1命令列毎に実行されていくことになる。

【0068】

一方、シミュレーションプログラム実行手段903は、スケジュール管理手段901より特定機能パーツの実行を命令されると、機能パーツ203(図2)のシミュレーションプログラム格納部212に格納されたプログラムを実行する。

プログラムの最後まで実行し終わると、スケジュール管理手段901に実行終了を返す。

このように、機能パーツの実行は、プログラムパーツの実行とは異なり、全プログラムが1度実行されていくことになる。

このプログラムの実行結果は、ネットワーク表現による表示状態で、CRT1a(図1)上に、各機能パーツの表示されている近辺に表示出力される。

【0069】

以上のような構造をとることにより、プログラムパーツが保持するプログラムをインタプリティブに実行しつつ、そのプログラムの制御対象の状態をリアルタイムで表示可能なシミュレーションが実現されていく。

プログラムパーツの実行はインタプリティブであり、その実行をユーザに対して表示する場合、1命令列毎に実行を止めユーザからのコマンド待ちとするトレースモードや、止めずに最後まで実行するモードが実現できる。

【0070】

図10は、シミュレーションを行うときのCRT1a(図1)上の画面状態を示す構成図である。

同図において、101はネットワーク記述表示部、102はプログラムパーツを示すパーツアイコン、103はパーツアイコン102のプログラムが表示されるプログラム記述シート表示部、104は反応容器を示すパーツアイコン、105aはタンクAを示すパーツアイコン、105bはタンクBを示すパーツアイコン、106a, b, c, dはバルブを示すパーツアイコン、107a, b, c, d, eはセンサを示すパーツアイコンである。

【0071】

また、108a, b, c, d, e, f, gはパーツアイコン102が示すシーケンス、すなわち、プログラム記述シート表示部103に記述されている内容に沿って制御を行うコントローラを示すパーツアイコン、109aはパーツアイコン106dで示されるバルブをシーケンスとは独立して分散制御をする制御部を示すパーツアイコン、109bはパーツアイコン104で示される反応容器における攪拌動作をシーケンスとは独立して制御する制御部を示すパーツアイコン、109cはパーツアイコン107eで示されるセンサからの信号を検出してフィードバック制御を行う制御部を示すパーツアイコンである。

【0072】

この図10は、ネットワーク記述表示部101に表示されたネットワーク記述の階層パーツをルートとし、各パーツアイコンで示される部分を部分木とし、これらの単位でのシミュレーションを行う状態を示している。

その部分木で示される階層構造には、パーツアイコン102で示される「シーケンス」と名付けられたプログラムパーツと、そのプログラムパーツの制御対象となるパーツアイコン108a~gで示されるコントローラが存在する。

10

20

30

40

50

そして、プログラム記述シート表示部 103 に示されるそのプログラムの動特性をシミュレートするものである。

【0073】

そして、このシミュレートを行うためにパーツアイコン 109 a ~ c で示される制御部のもつパラメータを設定変更可能とするために、図 10 (b) に示すように、設定値変更窓 110 a, b, c が設けられている。

この設定値変更窓 110 a, b, c は、マウス 2 (図 1) などの新たな入力指示により表示させるようにすることもできるが、シミュレーション実行対象管理テーブル 904 に登録されている全機能パーツに対し、自動的に付加して表示させるようにすることもできる。

10

【0074】

これらの設定値は、プログラム記述シート表示部 103 に表示されているプログラムの実行とは関係なく、各外部変数の値 (実際には制御部は構造体を形成しており、その構造体の各メンバー変数) を設定することができる。

そこで、シミュレーションを実行させる前に、設定値変更窓 110 a ~ c を介して、パーツアイコン 109 a ~ c で示される制御部の分散制御における初期値を設定する。

【0075】

次に、プログラム記述シート表示部 103 に表示されているプログラムを実行させるが、これはインタプリタ的に 1 命令列ずつ実行される。

この実行にはモード指定が可能であり、1 命令列を実行してはストップし、ユーザーからのコマンド入力待ちとなるトレースモードや、連続実行モードが選択可能である。また、ブレークポイントの設定も可能となっている。

20

1 命令列が実行されるたびに、その結果が制御対象である各制御部 (機能パーツ) に反映し、設定値変更窓 110 a ~ c を介して参照することができる。

【0076】

以上のようにプログラムとその制御対象となる分散制御システムの制御部などの外部変数の動特性を同時にシミュレートすることで、そのプログラムの動特性が理解しやすくなる。

従って、ソフトウェア部品内部のある部分木を切り出して再利用しようとする場合に、それが利用目的に合致しているかが素早くわかり、ソフトウェア再利用の促進ができる。

30

【0077】

【発明の効果】

以上説明したように、この発明によれば、まず、複数のパーツからなる設計情報を構築するための部品 (ソフトウェア部品) の構成を階層構造とし、これを表示出力して検索抽出するようにし、かつ、これら部品単位でシミュレートするようにした。

また、各部品に内容の説明情報と他との関係情報とを付帯させるようにした。

このためソフトウェア設計情報の作成者の意図と、それを構成する部品の中より利用できるものは再利用しようとする利用者の利用目的とが異なっても、その部品が再利用できるという効果がある。

また、所望とする部品が効率よく検索できるという効果がある。

40

そして、所望とする各部品の仕様や構造およびその動作に関して、再利用者が容易に理解できるという効果がある。

【図面の簡単な説明】

【図 1】この発明の 1 実施例であるソフトウェア部品の再利用システムを実現するための、再利用管理装置の構成を示す構成図である。

【図 2】この実施例におけるソフトウェア部品の構成要素の種類およびその構成を示す構成図である。

【図 3】階層構造を示す説明図である。

【図 4】この発明におけるソフトウェア部品の構成を示す構成図である。

【図 5】部分木のルート階層パーツを検索する例を示す構成図である。

50

【図6】水平/垂直ブラウジングを行うための、マン・マシン・インタフェースを示す画面図である。

【図7】水平/垂直ブラウジングを行うための、マン・マシン・インタフェースを示す画面図である。

【図8】ソフトウェア部品の階層構造を示す構成図である。

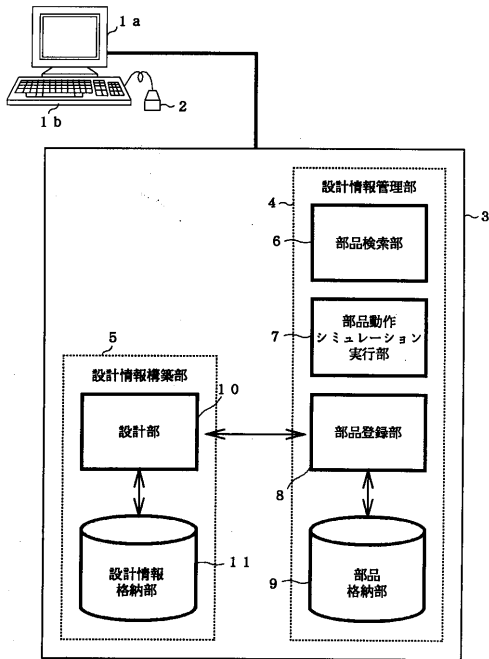
【図9】図1に示した部品動作シミュレーション実行部7の内部構造を示す構成図である。

【図10】シミュレーションを行うときのCRT1a(図1)上の画面状態を示す構成図である。

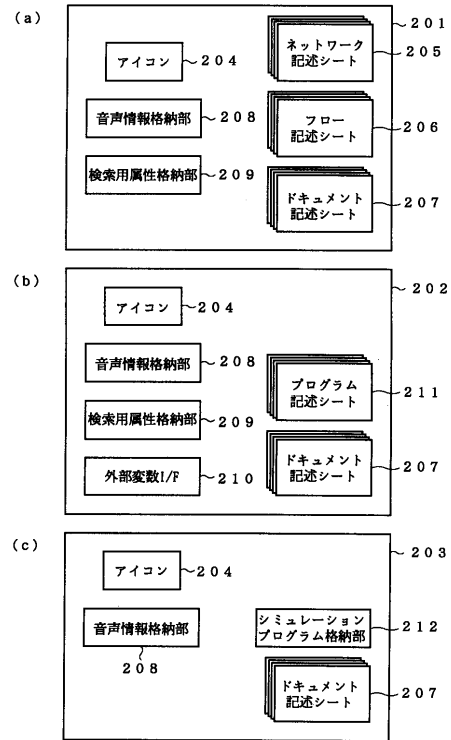
【符号の説明】

1 a ... CRT、1 b ... キーボード、2 ... マウス、3 ... 装置本体、4 ... 設計情報管理部、5 ... 設計情報構築部、6 ... 部品検索部、7 ... 部品動作シミュレーション実行部、8 ... 部品登録部、9 ... 部品格納部、10 ... 設計部、11 ... 設計情報格納部。

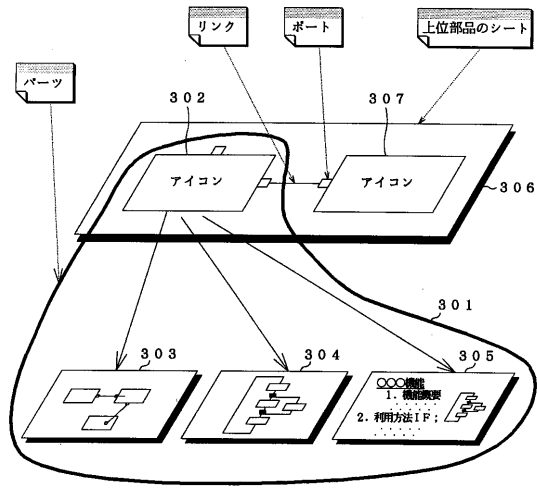
【図1】



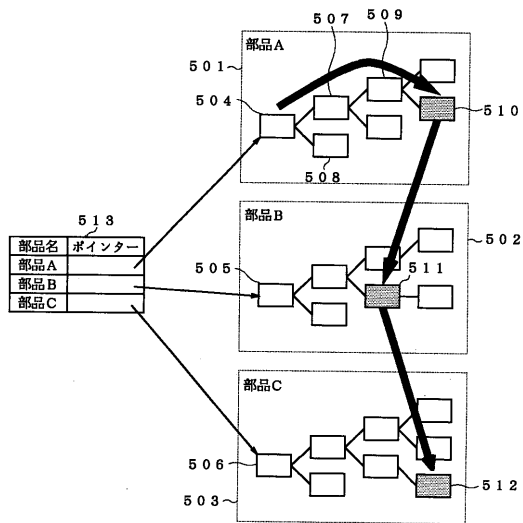
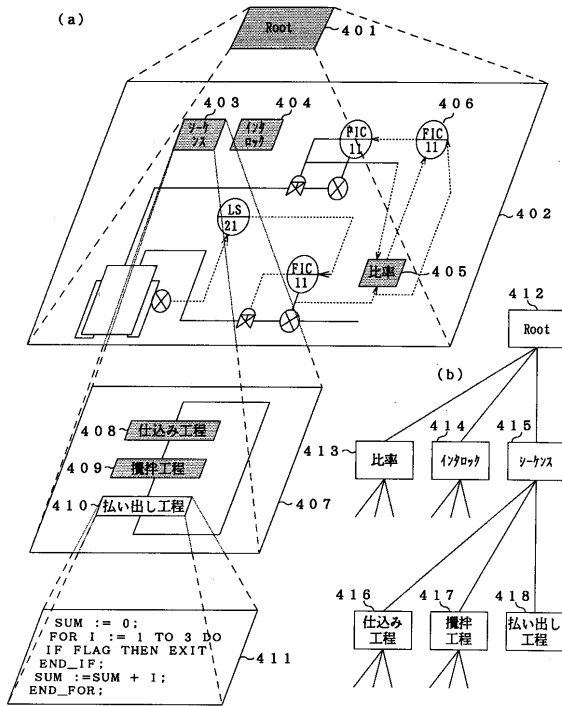
【図2】



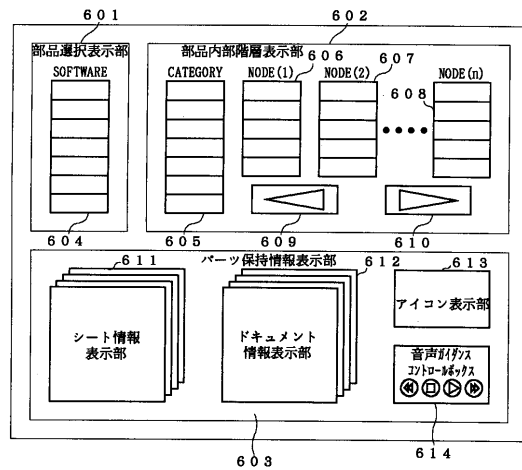
【 図 3 】



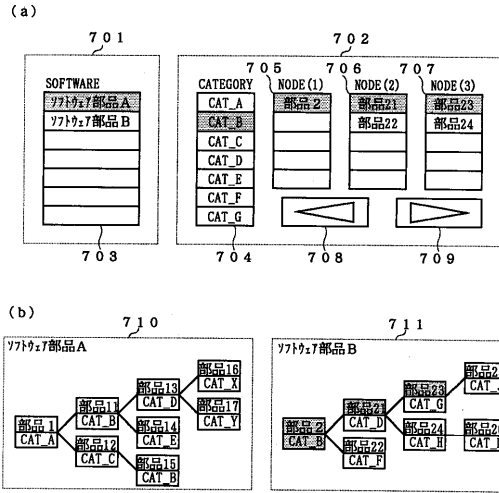
【 図 4 】



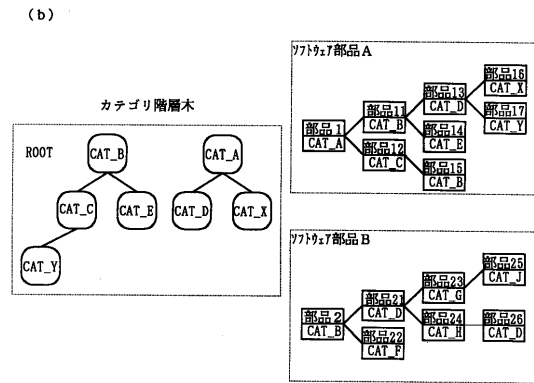
【 図 6 】



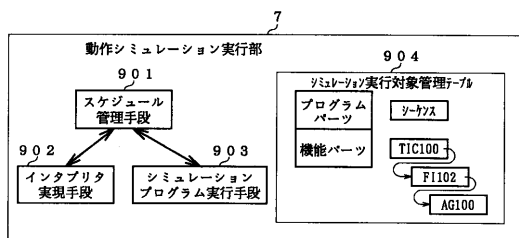
【 図 7 】



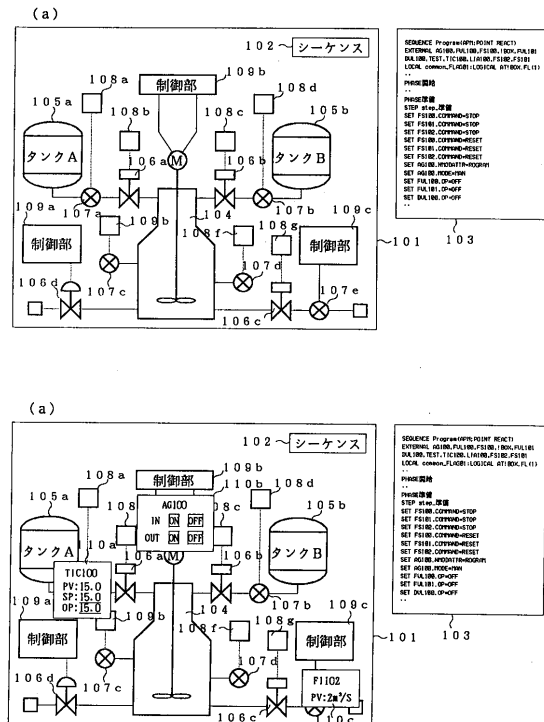
【 図 8 】



【 図 9 】



【 図 10 】



フロントページの続き

- (72)発明者 柏屋 弘
神奈川県横浜市保土ヶ谷区神戸町134 YBPウエストタワー12階 山武ハネウエル株式会社
アドバンステクノロジーセンター内
- (72)発明者 熊谷 敏
神奈川県横浜市保土ヶ谷区神戸町134 YBPウエストタワー12階 山武ハネウエル株式会社
アドバンステクノロジーセンター内
- (72)発明者 堀 雅智
神奈川県横浜市保土ヶ谷区神戸町134 YBPウエストタワー12階 山武ハネウエル株式会社
アドバンステクノロジーセンター内

合議体

審判長 川名 幹夫

審判官 橋本 正弘

審判官 松浦 功

- (56)参考文献 特開平5-307472(JP,A)
角谷和俊 外3名 "家電ソフトウェア設計支援システムVisual CASEの開発", 情報処理学会研究
報告93-S E - 91,平成5年,Vol.93, No.16, P9~15