

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2012年9月7日(07.09.2012)



(10) 国際公開番号
WO 2012/117746 A1

- (51) 国際特許分類:
G06F 11/36 (2006.01) G06F 11/34 (2006.01)
- (21) 国際出願番号: PCT/JP2012/001478
- (22) 国際出願日: 2012年3月2日(02.03.2012)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2011-045592 2011年3月2日(02.03.2011) JP
特願 2011-045594 2011年3月2日(02.03.2011) JP
- (71) 出願人(米国を除く全ての指定国について): 兵庫県(HYOGO PREFECTURE) [JP/JP]; 〒6508567 兵庫県神戸市中央区下山手通5丁目10番1号 Hyogo (JP). 株式会社ウィッツ(WITZ CO., LTD.) [JP/JP]; 〒4600008 愛知県名古屋市中区栄二丁目13番1号白川第2ビル Aichi (JP).
- (72) 発明者; および
- (75) 発明者/出願人(米国についてのみ): 中本 幸一 (NAKAMOTO, Yukikazu) [JP/JP]; 〒6512197 兵庫県

西区学園西町8丁目2-1 兵庫県立大学内 Hyogo (JP). 大西 秀一(ONISHI, Shuichi) [JP/JP]; 〒4600008 愛知県名古屋市中区栄二丁目13番1号白川第2ビル 株式会社ウィッツ内 Aichi (JP).

(74) 代理人: 特許業務法人グローバル知財(THE PATENT CORPORATE BODY GLOBAL INTELLECTUAL PROPERTY); 〒6500024 兵庫県神戸市中央区海岸通4番地 新明海ビル3F Hyogo (JP).

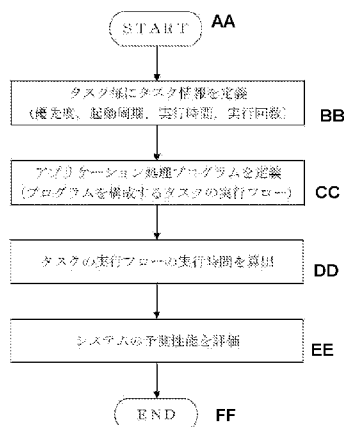
(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[続葉有]

(54) Title: COMPUTER SOFTWARE SYSTEM DESIGN SUPPORT DEVICE

(54) 発明の名称: 計算機ソフトウェアシステム設計支援装置

[図1]



- AA Start
- BB Define the task information for each task (priority, activation cycle, execution time, execution frequency)
- CC Define the application processing program (execution flow of the tasks constituting the program)
- DD Calculate the execution time for the execution flow of the tasks
- EE Evaluate the predicted performance of the system
- FF End

(57) Abstract: Provided are a computer software design support method and a design support device, wherein the design step pertaining to performance issues which is one of the serious causes for malfunction is ameliorated and development costs associated with improving the quality of embedded software and reducing the number of back track steps are reduced. At least the priority, activation cycle, execution time and execution frequency are defined for each task constituting an application processing program. Moreover, the execution flow of tasks constituting the application processing program is defined. The execution time for the execution flow of the tasks is calculated by using the priority, activation cycle, execution time and execution frequency defined for each task. The system performance is predicted from the minimum execution time, maximum execution time and execution time distribution of the application processing program.

(57) 要約: 不具合の重大な原因の一つである性能問題に関連する設計工程を改善し、組込みソフトウェアの品質向上と後戻り工数削減による開発コストの抑制を図るべく、計算機ソフトウェア設計支援方法および設計支援装置を提供する。アプリケーション処理プログラムを構成するタスク毎に少なくとも優先度、起動周期、実行時間、実行回数を定義する。また、アプリケーション処理プログラムを構成するタスクの実行フローを定義する。そして、タスク毎に定義された優先度、起動周期、実行時間、実行回数を用いて、タスクの実行フローの実行時間を算出する。そして、アプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布からシステムの性能予測

を行う。

WO 2012/117746 A1



(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称： 計算機ソフトウェアシステム設計支援装置

技術分野

[0001] 本発明は、スタンドアロンで動作する計算機ソフトウェアの性能予測を行い、計算機ソフトウェアの設計ならびに性能を事前評価するための設計支援方法と設計支援装置に関するものである。

背景技術

[0002] 情報家電や携帯電話などの計算機搭載デバイスに要求される機能が増大したことにより、デバイスに搭載されるソフトウェア量が増大し、1つのシステムに搭載されるソフトウェア部品も莫大な数になっている。特に、携帯電話の場合、通話機能、電子メール機能、ブラウザ機能など、200～300のソフトウェア部品が同居している。

そのため、このような計算機搭載デバイスの品質の問題がつきまとう。かかる品質問題は、シミュレータなどで解決することもできるが、性能問題はソフトウェア部品を結合した後でないと性能が満足するか否かわからないことが多い。

なぜなら、これらソフトウェア部品は1つのハードウェア資源を共有しながら同時に作動するので、1つのソフトウェア部品の実行が他のソフトウェア部品の実行を遅らせる原因になるからである。性能問題が発見された場合、開発工程の後戻りが発生し、膨大なコストが発生しているのが現状である。

[0003] かかる状況下、従来から、図15に示すようなサーバコンピュータの設計モデルには待ち行列モデルが多く利用されている（例えば、非特許文献1）。特に、待ち行列モデルは、ネットワークシステムで利用されてきている。しかしながら、待ち行列モデルはシステム設計には有効なもののソフトウェア設計には待ち行列モデルとソフトウェアの設計モデルに大きいギャップがあり、実用的なアプローチとは言い難いものであった。

また、ソフトウェアの実行に際して、必要なハードウェア資源量を各々のステップのまとまり単位で見積って、全体システムの性能を予測するアプローチ方法を提示している（非特許文献2）。但し、システム内の共有資源へのアクセスはしばしば性能に大きいな影響を与えるが、この性能評価においても、待ち行列モデルを利用したものであり、上記と同様の問題がある。

[0004] また、コンピュータシステム内の性能予測のアプローチ方法を開示しているものもある（特許文献1）。

また、並列計算機を対象とした性能評価方法もあるが（特許文献2）、こちらも待ち行列モデルをベースとした待ち行列網モデルを用いており上記と同じ問題がある。

[0005] また、実際にシステムでの性能を測定しそのデータに基づき性能予測を行うものがある（特許文献3）。この方法は、系列システムの開発には有効であろうが、最初にシステムを設計する場合にはあまり有効に機能するものではない。

また、計算機システム内の粒度の粗い部品レベルでの性能予測方法がある（特許文献4）。しかしながら、粒度の粗い部品レベルでの性能予測の場合、リアルタイムシステムなどの時間制約が厳しいものについては性能評価できない。

先行技術文献

特許文献

[0006] 特許文献1：特開平4-363730号公報

特許文献2：特開2000-298598号公報

特許文献3：特開2004-220453号公報

特許文献4：特開2004-272582号公報

非特許文献

[0007] 非特許文献1：滝根 哲哉、伊藤 大雄、西尾 章治郎、ネットワーク設計理論、岩波書店、2001

非特許文献2：C. Smith and L. William's, "Performance Solutions," Add

ison Wesley, 2002

発明の概要

発明が解決しようとする課題

[0008] 組込みシステムの動作不良原因は、組込みソフトウェア不具合が半数以上を占める。不具合が起きる理由として、組込みソフトウェアの大規模化と複雑化に対して、十分な対策がとられなかったためである。

例えば、携帯電話の場合、図13に示すように、製品競争力確保のために新機能の搭載や、新技術を利用した機能向上や、顧客要望に対応する機能改善を行っており、ワンセグ機能、メール機能、通話機能、ブラウザ機能、動画再生機能など数多くのプログラムが搭載されている。このため、携帯電話の開発規模が増大し、プログラムの組合せ動作検証が困難になっている。不具合原因の50%以上がプログラムに起因するものであり、それらの不具合が製品開発段階の終盤や製品出荷後に発生している。

このため、搭載される数多くのプログラムの個々の機能の性能要求を明確化し、事前動作検証を行うことにより、開発期間・工程・後戻り・不具体対策コストの抑制を図るべく、機能だけでなく要求性能を満足できるようにプログラムの設計支援を行えるツールが要望されている（図14を参照）。

[0009] 上記状況に鑑みて、本発明は、不具合の重大な原因の1つである性能問題に関連する設計工程を改善し、組込みソフトウェアの品質向上と後戻り工数削減による開発コストの抑制を図るべく、計算機ソフトウェア設計支援方法および設計支援装置を提供することを目的とする。

課題を解決するための手段

[0010] 上記目的を達成すべく、本発明の計算機ソフトウェア設計支援方法は、少なくとも1つのアプリケーション処理プログラムを搭載したシステムの性能予測を行う計算機ソフトウェア設計支援方法であって、以下の1)～4)のステップを備える。

1) アプリケーション処理プログラムを構成するタスク毎に少なくとも優先度、起動周期、実行時間、実行回数を定義するステップ

2) アプリケーション処理プログラムを構成するタスクの実行フローを定義するステップ

3) タスク毎に定義された優先度、起動周期、実行時間、実行回数を用いて、タスクの実行フローの実行時間を算出するステップ

4) アプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布からシステムの性能予測を行うステップ

[0011] かかる構成によれば、携帯電話や情報家電など多くのアプリケーション処理プログラムを搭載したシステムの性能予測について、設計段階において事前検証することができる。

上記1)のアプリケーション処理プログラムを構成するタスク毎の優先度、起動周期、実行時間、実行回数について説明する。ここで、アプリケーション処理プログラムはシステムが持つ機能をコンピュータに実現させるためのものである。また、タスクは機能をコンピュータに実現させるためのコンポーネントとなる処理のことをいう。タスクの優先度とは、実行可能なタスクが複数存在する場合にコンピュータに実行させる順番を決定するための指標である。また、タスクの起動周期とは、周期的にタスクを起動させる場合はその起動時間間隔をいう。タスクがイベントにより動作する処理である場合は起動周期を例えば0（ゼロ）とする。タスクの実行時間とは、コンピュータがタスクを実行する予測時間である。この予測時間は、タスクが行う処理ボリュームやタスクがハードウェアリソースにアクセスする場合はその処理待ち時間を考慮する。また、タスクの実行回数とは、周期的に起動されるタスクの実行回数やエラーリトライするタスクのリトライ回数をいう。

[0012] 次に、上記2)のアプリケーション処理プログラムを構成するタスクの実行フローについて説明する。ここで、アプリケーション処理プログラムとは、例えば、携帯電話の場合には、ブラウザ機能を実現するプログラムや、電子メール機能を実現するプログラムや、通話機能を実現するプログラムなどをいう。アプリケーション処理プログラムを構成するタスクの実行フローとは、アプリケーション処理プログラムを構成する1以上のタスクの実行順序

、繰り返し、条件分岐、タスク間のリソースの排他制御などをフローチャートの如く示したものをいう。

[0013] 次に、上記3)のタスク毎に定義された優先度、起動周期、実行時間、実行回数を用いて、タスクの実行フローの実行時間をどのように算出するとは、タスクの実行フローにしたがって、各タスクの実行時間を加算していくことである。

1つのアプリケーション処理プログラムを搭載するシステムの場合、そのアプリケーション処理プログラムを構成するタスクの実行フローにしたがって、順番に実行時間を加算していく。タスクが処理待ち時間を要するリソースにアクセスする場合は、その処理待ち時間も加算する。

[0014] 一方、複数のアプリケーション処理プログラムを搭載するシステムの場合、アプリケーション処理プログラム毎にタスクの実行フローが複数存在する。実行フローが複数存在する場合、実行可能なタスクが複数存在するケースも有り得る。この場合、例えば、1方の実行フローのタスクでリソースの処理待ち時間が生じた場合や特定条件が成立するまで処理を待つアルゴリズムを有する場合は、他方の実行フローで実行可能なタスクに切り替わり当該タスクが実行されるとして実行時間を加算する。また、オペレーティングシステム(OS)によって優先度の高いものが実行されることから、例えば、1方の実行フローで実行可能なタスクの優先度と他方の実行フローで実行可能なタスクの優先度とを比べて、優先度の高い側のタスクが実行されるとして実行時間を加算する。

[0015] 本発明の計算機ソフトウェア設計支援方法における上記1)のタスクの実行時間において、タスク実行時間の分布情報が付加されることが好ましい。

タスクによっては、常に一定の実行時間で処理を終わるものではないものも存在することに着眼して、タスク実行時間の分布情報を加えるものである。タスク実行時間の分布情報とは、実行時間のバラツキ分布や、実行時間の確率分布や、ガウス分布などの数式で示す分布情報などである。なお、タスク実行時間の分布情報は、タスクの実行時間のバラツキを表すものであれば

、これらに限定されずとも構わない。

[0016] 例えば、タスク実行時間の分布情報は、見積り実行時間とその重み付けで定義されることがより好ましい。重み付けは、見積り実行時間になる確率で表現してもよい。

[0017] 例えば、タスク実行時間の分布情報は、該タスクがハードウェアリソースにアクセスする場合に、ハードウェアリソースのアクセス時間を含むことがより好ましい。

ここで、ハードウェアリソースとは、共有メモリ、フラッシュメモリ (flash memory) などの不揮発性の半導体メモリ、USB (Universal Serial Bus) メモリ、ハードディスクなどをいう。また、ハードウェアリソースのアクセス時間とは、リソースにアクセスするための書き込み時間、読み込み時間などの処理待ち時間である。

[0018] また、タスク実行時間の分布情報には、該タスクが排他制御ロジックを含む場合に、排他制御の処理待ち時間を含むことがより好ましい。なお、排他制御ロジックにはタイムアウト時間を設定しても構わない。

[0019] また、タスク実行時間は、該タスクに用いられるシステムソフトウェアのアプリケーションインタフェース関数の見積り実行時間を積み上げたものを参考指標値として用いることがより好ましい。

ここで、システムソフトウェアとは、オペレーティングシステム (OS) や画面描画サービスプログラムなど、アプリケーション処理プログラムの枠外のプログラムである。

[0020] また、本発明の計算機ソフトウェア設計支援方法において、アプリケーション処理プログラム毎の性能目標となる予測時間領域を定義し、シミュレーション結果の前記実行時間分布が前記予測時間領域と比較して逸脱する場合には、該アプリケーション処理プログラムに対してアラームを出力することが好ましい。

ここで、アプリケーション処理プログラム毎の性能目標となる予測時間領

域とは、処理時間の上限を定めるもの、下限を定めるもの、上下限を定めるもの、いくつかの時間を定めるもの、いくつかの時間領域を定めるものが例として挙げられる。

[0021] また、本発明の計算機ソフトウェア設計支援方法において、アプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布のシミュレーションは、条件実行される1以上の各タスクに対して、それぞれの実行確率が定義されることにより、実行時間分布が生じることが好ましい。

ここで、条件実行される1以上の各タスクに対してそれぞれの実行確率が定義されるとは、例えば、条件によって2つの処理に分かれる場合、条件によって実行されるタスクが異なることになることから、条件によって実行時間が変わることになる。そこで、それぞれのタスクの実行確率を定義することで、条件によって処理が分かれるロジックを備えるアプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布をシミュレーションすることにしたのである。

[0022] また、本発明の計算機ソフトウェア設計支援方法において、アプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布のシミュレーションは、複数のタスクで共通アクセスされるハードウェアリソースに対して、排他制御を行うことにより、実行時間分布が生じることが好ましい。

複数のタスクで共通アクセスされるハードウェアリソースに対して排他制御を行う場合、通常、フラグやセマフォやトークンなどを用いて排他制御を行う。このフラグの値が変化するまでの時間やセマフォなど取得までの時間が処理待ち時間となる。この処理待ち時間は、タスクの実行状況によって変化するため実行時間分布が生じることになるのである。

[0023] 次に、本発明の計算機ソフトウェアシステム設計支援装置について説明する。

本発明の計算機ソフトウェアシステム設計支援装置は、アプリケーション処理プログラムを構成するタスク毎の優先度、起動周期、実行時間、実行回

数を定義してタスク情報データベースとして登録するタスク定義入力手段と、アプリケーション処理プログラム毎に用いるタスクの実行フローを、予め登録されているフロー定義部品を用いて定義して、プログラム定義データベースとして登録するプログラム定義入力手段と、タスク情報データベースとプログラム定義データベースおよびフロー定義部品が格納されているフロー定義部品データベースを用いて、アプリケーション処理プログラム毎の実行時間分布をシミュレーションする実行時間分布シミュレーション手段と、システムの性能評価結果を表示する結果表示手段と、を備える。

[0024] かかる構成によれば、携帯電話や情報家電など多くのアプリケーション処理プログラムを搭載したシステムの性能予測について、設計段階において事前検証することができる。

上記のアプリケーション処理プログラムを構成するタスク毎の優先度、起動周期、実行時間、実行回数を定義してタスク情報データベースとして登録するタスク定義入力手段について説明する。ここで、アプリケーション処理プログラムはシステムが持つ機能をコンピュータに実現させるためのものである。また、タスクは機能をコンピュータに実現させるためのコンポーネントとなる処理のことをいう。タスクの優先度とは、実行可能なタスクが複数存在する場合にコンピュータに実行させる順番を決定するための指標である。また、タスクの起動周期とは、周期的にタスクを起動させる場合はその起動時間間隔をいう。タスクがイベントにより動作する処理である場合は起動周期を例えば0（ゼロ）とする。タスクの実行時間とは、コンピュータがタスクを実行する予測時間である。この予測時間は、タスクが行う処理ボリュームやタスクがハードウェアリソースにアクセスする場合はその処理待ち時間を考慮する。また、タスクの実行回数とは、周期的に起動されるタスクの実行回数やエラーリトライするタスクのリトライ回数をいう。

[0025] タスク情報データベースは、これらの優先度、起動周期、実行時間、実行回数の情報を、タスク毎に割り当てられたユニークな識別番号により参照できるデータベースである。なお、タスク毎に割り当てられたユニークな識別

番号はユニークなタスク名でもよい。

[0026] 次に、上記のアプリケーション処理プログラム毎に用いるタスクの実行フローを、予め登録されているフロー定義部品を用いて定義して、プログラム定義データベースとして登録するプログラム定義入力手段について説明する。ここで、アプリケーション処理プログラムとは、例えば、携帯電話の場合には、ブラウザ機能を実現するプログラムや、電子メール機能を実現するプログラムや、通話機能を実現するプログラムなどをいう。アプリケーション処理プログラムを構成するタスクの実行フローとは、アプリケーション処理プログラムを構成する1以上のタスクの実行順序、繰り返し、条件分岐、タスク間のリソースの排他制御などをフローチャートの如く示したものをいう。

[0027] また、フロー定義部品とは、タスクの実行順序、繰り返し、条件分岐、タスク間のリソースの排他制御などを、それぞれシンボル図形化したものであり、繰り返し回数、条件分岐の確率、排他制御のタイムアウトなどのパラメータを有するものである。

また、プログラム定義データベースとは、該プログラムが使用するフロー定義部品とそのパラメータ、該プログラムがアクセスするリソース、該プログラムを構成するタスクの識別番号をプログラム毎に割り当てられたユニークな識別番号により参照できるデータベースである。なお、プログラム毎に割り当てられたユニークな識別番号はユニークなプログラム名でもよい。

[0028] 次に、上記のタスク情報データベースとプログラム定義データベースおよびフロー定義部品が格納されているフロー定義部品データベースを用いて、アプリケーション処理プログラム毎の実行時間をシミュレーションする実行時間分布シミュレーション手段について説明する。アプリケーション処理プログラム毎の実行時間分布のシミュレーションは、該プログラムに対応するプログラム定義データベースから、使用されるタスクと使用されるフロー定義部品を認識し、該タスクに対応するタスク情報データベースから優先度や実行時間などを認識し、また該フロー定義部品に対応するフロー定義部品デ

データベースから処理内容とパラメータを認識する。

[0029] そして、認識したタスクの優先度や実行時間、認識した実行順序などに基づいて、タスクの実行フローにしたがって、各タスクの実行時間を加算していくことにより、アプリケーション処理プログラム毎の実行時間をシミュレーションする。

1つのアプリケーション処理プログラムを搭載するシステムの場合、そのアプリケーション処理プログラムを構成するタスクの実行フローにしたがって、順番に実行時間を加算していく。タスクが処理待ち時間を要するリソースにアクセスする場合は、その処理待ち時間も加算する。

[0030] 一方、複数のアプリケーション処理プログラムを搭載するシステムの場合、アプリケーション処理プログラム毎にタスクの実行フローが複数存在する。実行フローが複数存在する場合、実行可能なタスクが複数存在するケースも有り得る。この場合、例えば、1方の実行フローのタスクでリソースの処理待ち時間が生じた場合や特定条件が成立するまで処理を待つアルゴリズムを有する場合は、他方の実行フローで実行可能なタスクに切り替わり当該タスクが実行されるとして実行時間を加算する。また、オペレーティングシステム（OS）によって優先度の高いものが実行されることから、例えば、1方の実行フローで実行可能なタスクの優先度と他方の実行フローで実行可能なタスクの優先度とを比べて、優先度の高い側のタスクが実行されるとして実行時間を加算する。

[0031] 次に、上記のシステムの性能評価結果を表示する結果表示手段について説明する。シミュレーションした結果のプログラム毎の実行時間分布は、実行最小時間、実行最大時間、発生頻度をグラフや表にして装置に付属する液晶パネルなどの表示画面に結果を表示する。また、シミュレーション結果は、プログラム毎あるいはプログラムのバージョン毎に異なるファイルとして保存される。

[0032] また、上記の計算機ソフトウェアシステム設計支援装置において、アプリケーション処理プログラム毎の性能目標情報として、アプリケーション処理

プログラム毎に予測時間領域を上記のプログラム定義データベースに定義する性能目標情報定義手段を更に備えることが好ましい。

性能目標情報とは、最小実行時間、実行最大時間、平均実行時間、分布度合いである。

[0033] また、上記の計算機ソフトウェアシステム設計支援装置における結果表示手段は、シミュレーション結果の前記実行時間分布が前記予測時間領域と比較して逸脱する場合には、該アプリケーション処理プログラムに対してアラームを出力することが好ましい。

ここで、アプリケーション処理プログラム毎の性能目標となる予測時間領域とは、処理時間の上限を定めるもの、下限を定めるもの、上下限を定めるもの、いくつかの時間を定めるもの、いくつかの時間領域を定めるものが例として挙げられる。

[0034] また、上記の計算機ソフトウェアシステム設計支援装置におけるタスク定義入力手段において、タスク毎に起動オフセット時間、実時間属性およびタイムスライス時間を、タスク情報データベースに更に登録できる手段を備えることが好ましい。

ここで、起動オフセット時間とはタスクが起動するまでの遅れ時間である。主メモリに常駐するタスクの場合、遅れ時間は無視できるが、起動される度にハードディスクから処理コードが読み込まれる非常駐のタスクの場合は遅れ時間も考慮して設計を行うことが、システム全体の性能評価を正しく行えることになる。

また、実時間属性とは、本明細書において、割り込みのようなイベント駆動型のタスク、すなわちリアルタイム性が要求されるタスクか、周期的に処理を行うような周期駆動型のタスク、すなわち周期性が要求されるタスクか、CPU時間を実行可能な全てのタスクに分配するラウンドロビン方式が要求されるタスクか否かのタスクの属性を意味する。

[0035] また、上記の計算機ソフトウェアシステム設計支援装置におけるタスク定義入力手段において、タスク実行時間の分布情報を見積り実行時間とその重

み付けの1セットとして複数セットを分布情報として登録できる手段を備えることが好ましい。

重み付けは、見積り実行時間になる確率で表現してもよい。タスクによっては、常に一定の実行時間で処理を終わるものではないものも存在することに着眼して、タスク実行時間の分布情報を加えるものである。タスク実行時間の分布情報とは、実行時間のバラツキ分布や、実行時間の確率分布や、ガウス分布などの数式で示す分布情報などである。なお、タスク実行時間の分布情報は、タスクの実行時間のバラツキを表すものであれば、これらに限定されずとも構わない。

[0036] また、上記の計算機ソフトウェアシステム設計支援装置におけるタスク定義入力手段において、処理待ち時間が発生するハードウェアリソースの処理待ち時間の見積り時間を登録する処理待ち時間登録手段を更に備えることが好ましい。

ここで、ハードウェアリソースとは、上述したように、共有メモリ、フラッシュメモリなどの不揮発性の半導体メモリ、USBメモリ、ハードディスクなどをいう。また、ハードウェアリソースのアクセス時間とは、リソースにアクセスするための書き込み時間、読み込み時間などの処理待ち時間である。

また、ハードウェアリソースは複数のタスクで共通アクセスされるリソースであることがより好ましい。複数のタスクで共通アクセスされるリソースで排他制御ロジックを要する場合、排他制御の処理待ち時間を含む。なお、排他制御ロジックにはタイムアウト時間を設定しても構わない。複数のタスクで共通アクセスされるハードウェアリソースに対して排他制御を行う場合、通常、フラグやセマフォやトークンなどを用いて排他制御を行う。このフラグの値が変化するまでの時間やセマフォなど取得までの時間が処理待ち時間となる。この処理待ち時間は、タスクの実行状況によって変化するため実行時間分布が生じることになるのである。

[0037] また、上記の計算機ソフトウェアシステム設計支援装置において、アプリ

ケーション定義入力手段は、前記フロー定義部品データベースに登録された部品を用いて、繰り返し処理、条件分岐処理、並列実行処理を少なくとも定義入力できることがより好ましい。

[0038] ここで、条件実行処理は、条件実行される1以上の各タスクに対して、それぞれの実行確率を定義できることが好ましい。

条件実行される1以上の各タスクに対してそれぞれの実行確率が定義されるとは、例えば、条件によって2つの処理に分かれる場合、条件によって実行されるタスクが異なることになることから、条件によって実行時間が変わることになる。そこで、それぞれのタスクの実行確率を定義することで、条件によって処理が分かれるロジックを備えるアプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布をシミュレーションすることにしたのである。

[0039] また、並列実行処理は、並列に実行される全てのタスクが終了した場合のみ、その後の処理が進むパルドタイプ並列実行と、並列実行される全てのタスクが終了しなくても進むスプリットタイプ並列実行と、を備えることが好ましい。

発明の効果

[0040] ソフトウェア、特にデバイス機器に組み込まれる組み込みソフトウェアでは、CPU能力が低い、メモリが少ないなどの厳しいハードウェア制約の中でソフトウェアを設計する必要がある。

本発明によれば、ソフトウェアモジュールの設計・製作・デバイスに組み込みを行った段階で初めてシステム全体の性能が未達成であることが判明するといった事態を回避でき、ソフトウェア製作の膨大な後戻り工程やそれに伴うコストを削減できるといった効果を有する。

図面の簡単な説明

[0041] [図1] 計算機ソフトウェア設計支援フロー図

[図2] アプリケーション処理プログラムの説明図

[図3] アプリケーション処理プログラムの実行フローに用いるタスクの説明図

- [図4]タスクの実行フローの説明図 1
- [図5]タスクの実行フローの説明図 2
- [図6]タスクの実行フローの説明図 3
- [図7]タスクの実行フローの説明図 4
- [図8]タスクの実行フローの説明図 5
- [図9]タスクの実行フローの説明図 6
- [図10]タスクの実行フローの説明図 7
- [図11]映像再生システムの機能ブロック図
- [図12]映像再生システムにおけるプログラムの実行フロー図
- [図13]携帯電話の機能ブロック図 1
- [図14]携帯電話の機能ブロック図 2
- [図15]従来の待ち行列モデルを用いるトランザクションの概念図
- [図16]計算機ソフトウェアシステム設計支援装置の機能ブロック図
- [図17]プロジェクトの作成メニュー
- [図18]プロジェクトの作成画面
- [図19]タスク作成メニュー
- [図20]タスク作成画面
- [図21]アプリケーション処理プログラムの定義画面
- [図22]性能要求解析メニュー
- [図23]シミュレーション情報入力画面
- [図24]性能目標情報一覧画面
- [図25]性能目標情報入力画面
- [図26]タスク情報一覧画面
- [図27]タスク情報入力画面
- [図28]シミュレーション評価結果表示画面（全体情報画面）
- [図29]シミュレーション評価結果表示画面（シーケンス画面）
- [図30]シミュレーション評価結果表示画面（タスクトレース画面）

発明を実施するための最良の形態

[0042] 以下、本発明の実施形態について、図面を参照しながら詳細に説明していく。なお、本発明の範囲は、以下の実施例や図示例に限定されるものではなく、幾多の変更及び変形が可能である。

[0043] 図1は、本発明のコンピュータソフトウェア設計支援方法の処理フローを示している。

本発明のコンピュータソフトウェア設計支援方法は、1以上のアプリケーション処理プログラムを搭載するスタンドアロンのシステムに関する性能予測を以下のステップで行う。

(ステップ1) それぞれのアプリケーション処理プログラムを構成するタスク毎に、優先度、起動周期、実行時間、実行回数を定義する。

(ステップ2) アプリケーション処理プログラム毎に、プログラムを構成するタスクの実行フローを定義する。

(ステップ3) タスク毎に定義された優先度、起動周期、実行時間、実行回数を用いて、それぞれのアプリケーション処理プログラム毎に、タスクの実行フローの実行時間を算出する。

(ステップ4) 全てのアプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布に基づいて、システムの性能予測を行う。

[0044] アプリケーション処理プログラムとタスクの関係について、図2を用いて説明する。

図2は、アプリケーション処理プログラムの一例を示す機能ブロック図である。

図2のアプリケーション処理プログラムでは、処理タスク1と処理タスク2で構成されている。処理タスク1は、メッセージキュー1からメッセージデータ3を取り出し、メッセージデータ3を加工して共有メモリ5にデータを書き込む処理を行うものとする。また、処理タスク2は、共有メモリ5からデータを読み込み、データを加工して、フラッシュメモリ6にデータを書き込む処理と、場合によってはフラッシュメモリ6からデータを読み出して、メッセージキュー2に送る処理を行うものとする。

[0045] このようなアプリケーション処理プログラムにおいては、上記（ステップ 1）に従って、まず処理タスク 1 と処理タスク 2 の 2 つのタスクに関して、優先度、起動周期、実行時間、実行回数を定義することになる。

ここで、処理タスク 1 がメッセージ駆動型のタスクとし、処理タスク 2 が周期的駆動型のタスクとして設計されたとする。処理タスク 1 の起動周期は 0（ゼロ）で、処理タスク 2 の起動周期を例えば 10ms とする。また、メッセージ駆動型の処理タスク 1 は、一般的に、優先度が高く設計される。また、処理が 1 回だけ行われる場合、実行回数は 1 であり、複数回処理が繰り返される場合、実行回数は n 回となる。

[0046] 実行時間に関しては、処理タスク 1 の場合、タスクの行う処理、すなわち、メッセージキュー 1 からデータの取り出し処理、共有メモリ 5 への書き込み処理、それぞれの見積り時間を予測して決定する。また、処理タスク 2 の場合、共有メモリ 5 からデータを読み込み処理、フラッシュメモリ 6 にデータを書き込む処理、メッセージキュー 2 にデータを送る処理、それぞれの見積り時間を予測して決定する。

ここで、共有メモリの書き込み処理、読み込み処理、フラッシュメモリのアクセス処理などに処理待ち時間があればそれらを考慮して見積り時間を予測する。また、処理タスク 1 と処理タスク 2 がそれぞれ書き込み・読み込みを行う共有メモリ 5 に排他制御が行われる場合、制御権を受け取るまでの処理待ち時間も考慮して見積り時間を予測する。

[0047] 図 3（1）は、アプリケーション処理プログラムの実行フローに用いるタスクの種類の一例を示している。タスクは見積り予測が可能な単位に細かく定義する。タスクを繰り返し実行するものや、条件によってケース分けして多数を実行するものや、タスクを並列に実行するものなど定義する。

また、図 3（2）は、ハードウェアリソースの一例を示している。ハードウェアリソースには、リソースにアクセスするためのタスクの処理待ち時間が属性情報として付与されている。

[0048] 次に、アプリケーション処理プログラムを構成するタスクの実行フローに

ついて図4～10を参照して説明する。

まず、繰り返しタスクの実行フローの一例を示す。図4は繰り返しタスクの実行フローの一例を示している。例えば、初期化処理タスク10が $10\mu\text{sec}$ 、プロセスランザクションタスク12が $20\mu\text{sec}$ 、終了セッションタスク14が $5\mu\text{sec}$ の見積り時間で、繰り返しタスク16の繰り返し回数 $n=5$ とした場合、図4に示す実行フローの実行時間は、 $10+20\times 5+5=115\mu\text{sec}$ となる。

[0049] 次に、ケースタスクの実行フローの一例を示す。図5はケースタスクの実行フローの一例を示している。例えば、トランザクション取得タスク20が $15\mu\text{sec}$ 、プロセスAタスク22が $20\mu\text{sec}$ 、プロセスBタスク24が $40\mu\text{sec}$ 、プロセスCタスク26が $25\mu\text{sec}$ の見積り時間で、ケースタスク28の分岐確率 $a\sim c$ がそれぞれ 0.5 、 0.3 、 0.2 とした場合、図5に示す実行フローの実行時間は、最小実行時間が $15+20=35\mu\text{sec}$ 、最大実行時間が $15+40=55\mu\text{sec}$ で、実行時間分布が $35\mu\text{sec}$ 、 $55\mu\text{sec}$ 、 $15+25=40\mu\text{sec}$ がそれぞれの確率で現れる分布となる。

[0050] 次に、パルドタスクの実行フローの一例を示す。図6はパルドタスクの実行フローの一例を示している。例えば、ユーザ入力操作タスク30が $5\mu\text{sec}$ 、リモートクエリー表示タスク32が $15\mu\text{sec}$ 、クエリー結果をスクリーンに表示させるタスク34が $25\mu\text{sec}$ 、入力データを表示データとするタスク36が $10\mu\text{sec}$ 、ディスプレイ表示タスク38が $20\mu\text{sec}$ の見積り時間の場合、図6に示す実行フローの実行時間は、 $5+25+10+20=60\mu\text{sec}$ となる。なお、パルドタスク31は付属するタスク32とタスク34が並列実行され、それらのタスクが全て終了しないと先に処理が進まないロジックを備えるものである。

[0051] 次に、共有メモリを排他制御するタスクの実行フローの一例を示す。図7は、データ入力し共有メモリに書き込む2つのプログラムのタスクの実行フローの一例を示している。2つのアプリケーション処理プログラムは、繰り返し

返しタスク41とデータ入力タスク42と書込みデータタスク43の実行フローで構成されるプログラムと、繰り返しタスク46とデータ入力タスク47と書込みデータタスク48の実行フローで構成されるプログラムである。例えば、書込みデータタスク43の実行時間は $20\ \mu\text{sec}$ で、書込みデータタスク48の実行時間は $25\ \mu\text{sec}$ とすると、それぞれの書込みデータタスクは他方の書込みデータタスクが共有メモリ40に書込み中は処理待ちとなるため、書込みデータタスク43の実行時間は最小実行時間が $20\ \mu\text{sec}$ 、最大実行時間が $20 + 25 = 45\ \mu\text{sec}$ となり、書込みデータタスク48の実行時間は最小実行時間が $25\ \mu\text{sec}$ 、最大実行時間が $25 + 20 = 45\ \mu\text{sec}$ となる。

[0052] また、図8は、共有メモリにデータを書き込む処理と読み込む処理の2つのプログラムのタスクの実行フローの一例を示している。2つのアプリケーション処理プログラムは、繰り返しタスク51とデータ入力タスク52と書込みデータタスク53の実行フローで構成されるプログラムと、繰り返しタスク56と読み込みデータタスク57とスクリーン表示タスク58の実行フローで構成されるプログラムである。共有メモリ50に一方のプログラムがデータを書き込んで、他方のプログラムが書き込まれたデータを読み込んで処理する場合を想定する。例えば、データ入力タスク52の実行時間は $10\ \mu\text{sec}$ 、書込みデータタスク53の実行時間は $20\ \mu\text{sec}$ 、読み込みデータタスク57の実行時間は $15\ \mu\text{sec}$ 、スクリーン表示タスク58の実行時間は $25\ \mu\text{sec}$ とすると、データ入力してそれがスクリーン表示されるまでの最小実行時間は $10 + 20 + 15 + 25 = 70\ \mu\text{sec}$ となる。

[0053] ここで、データ入力タスク52の優先度と読み込みデータタスク57の優先度が異なる場合を考える。仮に、データ入力タスク52が、読み込みデータタスク57より優先される場合、データ入力してそれがスクリーン表示されるまでの最小実行時間は $10 + 20 + 10 + 15 + 25 = 80\ \mu\text{sec}$ となる。書込みデータタスク53が終了した後、繰り返しによって再びデータ入力タスク52が動作しそれが終了するまで、読み込みデータタスク57が動作し

ないからである。なお、データ入力タスク10が、スクリーン表示タスク58より優先される場合も同様となる。

[0054] 次に、フラッシュメモリにアクセスするタスクの実行フローの一例を示す。図9は、データ格納しそれをテンポラリメモリあるいはフラッシュメモリに格納するプログラムのタスクの実行フローの一例を示している。例えば、データ格納タスク60の実行時間は $15\ \mu\text{sec}$ 、テンポラリメモリへ格納タスク62の実行時間は $20\ \mu\text{sec}$ 、フラッシュメモリの処理待ち時間を $50\ \mu\text{sec}$ と見積ってフラッシュメモリへ格納タスク63の実行時間は $70\ \mu\text{sec}$ 、終了セッションタスク65の実行時間は $5\ \mu\text{sec}$ 、ケースタスク61の分岐確率がそれぞれ0.8と0.2とすると、図9に示す実行フローの実行時間は、最小実行時間が $15 + 20 = 40\ \mu\text{sec}$ 、最大実行時間が $15 + 70 + 5 = 90\ \mu\text{sec}$ で、実行時間分布が $40\ \mu\text{sec}$ 、 $90\ \mu\text{sec}$ がそれぞれの確率で現れる分布となる。

[0055] 次に、メッセージキューからメッセージを受け取るタスクの実行フローの一例を示す。図10は、メッセージキュー70からメッセージ受信しサービスを行うプログラムのタスクの実行フローの一例を示している。例えば、メッセージ受信タスク71の実行時間は $10\ \mu\text{sec}$ 、サービス処理タスク72の実行時間は $50\ \mu\text{sec}$ 、終了セッションタスク73の実行時間は $5\ \mu\text{sec}$ とすると、図10に示す実行フローの実行時間は、 $10 + 50 + 5 = 65\ \mu\text{sec}$ となる。

実施例 1

[0056] (映像再生装置)

図11に映像再生装置のアプリケーション処理プログラムの機能ブロック図を示す。図11には、ファイル読み込みタスク、映像復号タスク、映像表示タスク、音声復号タスク、音声再生タスクおよび共有メモリが示され、それぞれのタスクの起動周期および起動遅延、共有メモリの待ち時間およびアクセス単位が示されている。

まず、ファイル読み込みタスクが共有メモリから読み込んだデータを他の共

有メモリに書込む。次に、映像復号タスクと音声復号タスクがそれぞれ共有メモリのデータを取り出して復号し、それぞれ映像表示タスクと音声再生タスクに共有メモリを介して渡す。

図12は、図11の機能ブロックをタスクの実行フローに書き換えたものである。音声処理プログラムと映像処理プログラムの2つのプログラムに分け、それぞれのタスクの実行フローを示している。共有メモリ1~5について排他制御は行っていない。またファイル読み込みタスク(80, 86)は共通のタスクである。

[0057] 例えば、ファイル読み込みタスク(80, 86)の実行時間は $20\mu\text{sec}$ 、音声復号タスク82の実行時間は $10\mu\text{sec}$ 、音声再生タスク83の実行時間は $30\mu\text{sec}$ 、映像復号タスク87の実行時間は $40\mu\text{sec}$ 、映像表示タスク88の実行時間は $60\mu\text{sec}$ とすると、音声処理プログラムの最小実行時間は $60\mu\text{sec}$ 、映像処理プログラムの最小実行時間は $120\mu\text{sec}$ となる。また、最大実行時間は、処理が起動周期時間分遅れるとし、さらに起動遅延時間も加わるとして算出する。音声処理プログラムの場合、 $(2+1) + (4+1) + (16+1) = 25\text{msec}$ を最小実行時間に加算して最大実行時間を算出する。映像処理プログラムの場合、 $(2+1) + (4+1) + (16+1) = 25\text{msec}$ を最小実行時間に加算して最大実行時間を算出する。それぞれのタスクの起動周期が異なるために実行時間にはバラツキが生じることになるのである。

実施例 2

[0058] 実施例2では、計算機ソフトウェアシステム設計支援装置の一態様について説明する。

計算機ソフトウェアシステム設計支援装置は、汎用のデスクトップ型コンピュータやノート型コンピュータで構成される。

図16は、計算機ソフトウェアシステム設計支援装置の機能ブロック図を示している。

計算機ソフトウェアシステム設計支援装置は、図16の破線で囲まれたタ

スク情報データベースとプログラム定義データベースおよびフロー定義部品データベースを備える。これらのデータベースは、通常、計算機ソフトウェアシステム設計支援装置のハードディスクに保存される。

タスク情報データベースは、アプリケーション処理プログラムを構成するタスク毎のタスク名、優先度、起動オフセット、起動周期、実時間属性、実行時間、タイムスライス、実行回数の情報を、タスク毎に割り当てられたユニークなタスク名により参照できるデータベースである。

[0059] また、プログラム定義データベースは、プログラムが使用するフロー定義部品とそのパラメータ、該プログラムがアクセスするリソース、プログラムを構成するタスク名をプログラム毎に割り当てられたユニークなプログラム名により参照できるデータベースである。

[0060] また、フロー定義部品データベースは、繰り返し回数、条件分岐の確率、排他制御のタイムアウトなどのフロー定義部品のパラメータ情報を、タスクの実行順序、繰り返し、条件分岐、タスク間のリソースの排他制御などのフロー定義部品に割り当てられたユニークな番号により参照できるデータベースである。

[0061] また、図16に示すように、計算機ソフトウェアシステム設計支援装置は、タスク毎にタスク情報を定義してタスク情報データベースとして登録するタスク定義入力手段と、プログラム毎に用いるタスクの実行フローを予め登録されているフロー定義部品を用いて定義してプログラム定義データベースとして登録するプログラム定義入力手段とを備える。

また、アプリケーション処理プログラム毎の実行時間分布をシミュレーションするプログラムの実行時間分布シミュレーション手段と、システムの性能評価結果表示手段を備える。

[0062] 以下、計算機ソフトウェアシステム設計支援装置の操作メニューならびに画面を参照しながら説明する。

図17および図18は、それぞれプロジェクトの作成メニューとプロジェクトの作成画面を示している。計算機ソフトウェアシステム設計支援装置は

、設計対象デバイスもしくは設計対象システムをプロジェクト単位に管理する。従って、最初にプロジェクトを作成する。

[0063] 次に、アプリケーション処理プログラムとそれを構成する個々のタスクを定義する画面について説明する。タスクの定義は、UML (Unified Modeling Language) 2.0に沿ったクラス図を描いて定義する。UMLは、オブジェクト指向のソフトウェア開発における、プログラム設計図の統一表記法である。

また、アプリケーション処理プログラムの定義は、アプリケーション処理プログラム毎に用いるタスクの実行フローを、予め登録されているフロー定義部品を用いて定義することである。このタスクの実行フローをソフトウェア実行モデルと称している。

図19と図20は、それぞれタスク作成メニューとUML 2.0に沿ったタスク作成画面を示している。また図21は、アプリケーション処理プログラムの定義画面である。

[0064] 次に、性能要求解析について説明する。図22は性能要求解析メニューを示している。メニューから性能要求解析を選択すると、図23のシミュレーション情報入力画面が表示される。図23のシミュレーション情報入力画面を用いて、シミュレーション動作環境情報を入力する。具体的には、シミュレート時間とスケジューリング方式を入力する。

図23の画面において、画面内の“性能目標情報”タブを選択すると、図24の性能目標情報一覧画面が表示される。性能目標情報として、プログラムの時間制約、ここでは実行時間の上限が表示される。

個々のタスクの性能目標情報は、図25の性能目標情報入力画面を用いて作成する。

[0065] 次に、タスク情報について説明する。

図23の画面において、画面内の“タスク情報”タブを選択すると、図26のタスク情報一覧画面が表示される。タスク情報一覧画面では、タスク毎にタスク名、優先度、起動オフセット、起動周期、実時間属性、実行時間、

タイムスライス、実行回数の情報が一覧となって表示される。

個々のタスク情報は、図 27 のタスク情報入力画面を用いて作成する。

[0066] 次に、シミュレーションによる性能評価の結果表示画面について説明する。

全体情報としては、図 28 に示すように、シミュレート時間、実時間属性のタスク実行時間合計、CPU 使用率、性能目標未達成のプログラム一覧が全体情報画面として表示される。

図 28 の画面において、画面内の“シーケンス”タブを選択すると、図 29 のシーケンス画面が表示される。図 28 のグラフ情報として、プログラム名、性能目標、最小実行時間、最大実行時間が表示される。シーケンス画面によれば、アプリケーション処理プログラムの実行時間分布を把握することができる。

図 28 又は図 29 の画面において、画面内の“タスクトレース”タブを選択すると、図 30 のタスクトレース画面が表示される。タスクトレース画面によれば、アプリケーション処理プログラムを構成するタスクの実行順序と実行時間を把握することができる。図中のグラフでは、1つのプログラムが 5 個のタスク (DecodeAudio タスク, DecodeVideo タスク, Timing タスク, FileReader タスク, Playbutton タスク) から構成され、FileReader タスクの実行後、Timing タスクが実行される様子が示されている。

産業上の利用可能性

[0067] 本発明の計算機ソフトウェア設計支援方法および計算機ソフトウェア設計支援装置は、携帯電話端末や情報家電などスタンドアロンの計算機搭載デバイスのソフトウェア設計方法や設計支援装置として、また、自動車車載システムやロボット制御システムなどの複数の計算機で構成されるシステムに搭載されるソフトウェアの設計支援方法や設計支援装置として有用である。

符号の説明

[0068] 1, 2 メッセージキュー

3, 4	データ
5	共有メモリ
6	フラッシュメモリ
11, 12	タスク

請求の範囲

- [請求項1] 少なくとも1つのアプリケーション処理プログラムを搭載したシステムの性能予測を行う計算機ソフトウェア設計支援方法であって、
 アプリケーション処理プログラムを構成するタスク毎に少なくとも優先度、起動周期、実行時間、実行回数を定義するステップと、
 アプリケーション処理プログラムを構成するタスクの実行フローを定義するステップと、
 前記タスク毎に定義された優先度、起動周期、実行時間、実行回数を用いて、タスクの実行フローの実行時間を算出するステップと、
 アプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布からシステムの性能予測を行うステップと、
 を備えたことを特徴とする計算機ソフトウェア設計支援方法。
- [請求項2] 前記タスクの実行時間において、タスク実行時間の分布情報が付加されたことを特徴とする請求項1に記載の計算機ソフトウェア設計支援方法。
- [請求項3] 前記タスク実行時間の分布情報は、見積り実行時間とその重み付けで定義されることを特徴とする請求項2に記載の計算機ソフトウェア設計支援方法。
- [請求項4] 前記タスク実行時間の分布情報は、該タスクがハードウェアリソースにアクセスする場合に、ハードウェアリソースのアクセス時間を含むことを特徴とする請求項2に記載の計算機ソフトウェア設計支援方法。
- [請求項5] 前記タスク実行時間の分布情報は、該タスクが排他制御ロジックを含む場合に、排他制御の予想待ち時間を含むことを特徴とする請求項2に記載の計算機ソフトウェア設計支援方法。
- [請求項6] 前記タスク実行時間は、該タスクに用いられるシステムソフトウェアのアプリケーションインタフェース関数の見積り実行時間を積み上げたものを参考指標値として用いることを特徴とする請求項2に記載

の計算機ソフトウェア設計支援方法。

[請求項7] 前記アプリケーション処理プログラム毎の予測時間領域を定義し、シミュレーション結果の前記実行時間分布が前記予測時間領域と比較して逸脱する場合には、該アプリケーション処理プログラムに対してアラームを出力することを特徴とする請求項1～6のいずれかに記載の計算機ソフトウェア設計支援方法。

[請求項8] 前記アプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布のシミュレーションは、条件実行される1以上の各タスクに対して、それぞれの実行確率が定義されることにより、実行時間分布が生じることを特徴とする請求項1に記載の計算機ソフトウェア設計支援方法。

[請求項9] 前記アプリケーション処理プログラムの実行最小時間、実行最大時間、実行時間分布のシミュレーションは、複数のタスクで共通アクセスされるハードウェアリソースに対して、排他制御を行うことにより、実行時間分布が生じることを特徴とする請求項1に記載の計算機ソフトウェア設計支援方法。

[請求項10] アプリケーション処理プログラムを構成するタスク毎の優先度、起動周期、実行時間、実行回数を定義してタスク情報データベースとして登録するタスク定義入力手段と、

アプリケーション処理プログラム毎に用いるタスクの実行フローを、予め登録されているフロー定義部品を用いて定義して、プログラム定義データベースとして登録するプログラム定義入力手段と、

前記タスク情報データベースとプログラム定義データベースおよび前記フロー定義部品が格納されているフロー定義部品データベースを用いて、アプリケーション処理プログラム毎の実行時間分布をシミュレーションする実行時間分布シミュレーション手段と、

システムの性能評価結果を表示する結果表示手段と、

を備えたことを特徴とする計算機ソフトウェアシステム設計支援装

置。

- [請求項11] アプリケーション処理プログラム毎の性能目標情報として、アプリケーション処理プログラム毎に予測時間領域を前記プログラム定義データベースに定義する性能目標情報定義手段を更に備えたことを特徴とする請求項10に記載の計算機ソフトウェアシステム設計支援装置。
- [請求項12] 前記結果表示手段は、シミュレーション結果の前記実行時間分布が前記予測時間領域と比較して逸脱する場合には、該アプリケーション処理プログラムに対してアラームを出力することを特徴とする請求項11に記載の計算機ソフトウェアシステム設計支援装置。
- [請求項13] 前記タスク定義入力手段において、タスク毎に起動オフセット時間、実時間属性およびタイムスライス時間を、タスク情報データベースに更に登録できる手段を備えたことを特徴とする請求項10～12のいずれかに記載の計算機ソフトウェアシステム設計支援装置。
- [請求項14] 前記タスク定義入力手段において、タスク実行時間の分布情報を見積り実行時間とその重み付けの1セットとして複数セットを分布情報として登録できる手段を備えたことを特徴とする請求項10～12のいずれかに記載の計算機ソフトウェアシステム設計支援装置。
- [請求項15] 前記タスク定義入力手段において、見積り実行時間とその重み付けとを1セットとして、複数セットをタスク実行時間の分布情報として登録できるタスク実行時間分布情報入力手段を更に備えたことを特徴とする請求項10～12のいずれかに記載の計算機ソフトウェアシステム設計支援装置。
- [請求項16] 前記タスク定義入力手段において、処理待ち時間が発生するハードウェアリソースの処理待ち時間の見積り時間を登録する処理待ち時間登録手段を更に備えたことを特徴とする請求項15に記載の計算機ソフトウェアシステム設計支援装置。
- [請求項17] 前記ハードウェアリソースが、複数のタスクで共通アクセスされる

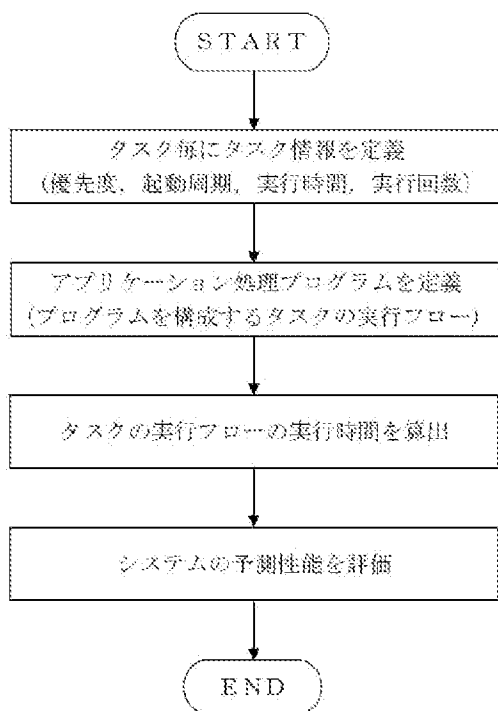
リソースであることを特徴とする請求項 16 に記載のコンピュータソフトウェアシステム設計支援装置。

[請求項18] 前記アプリケーション定義入力手段は、前記フロ一定義部品データベースに登録された部品を用いて、繰り返し処理、条件分岐処理、並列実行処理を少なくとも定義入力できることを特徴とする請求項 10 ~ 12 のいずれかに記載のコンピュータソフトウェアシステム設計支援装置。

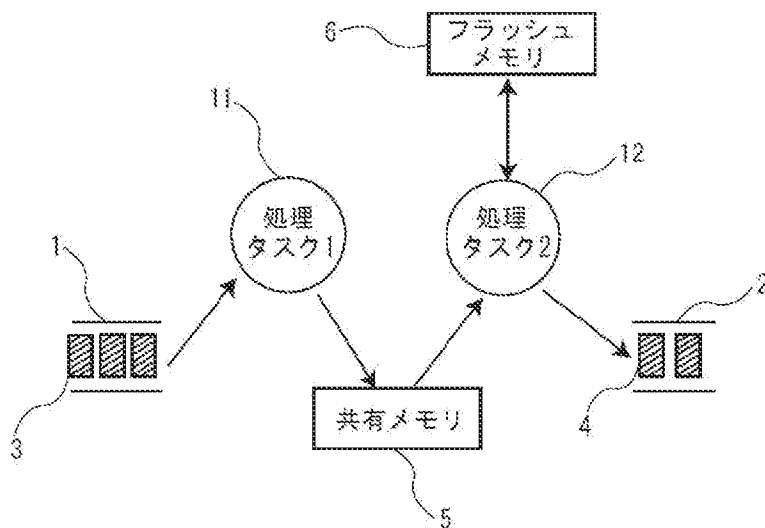
[請求項19] 前記条件実行処理は、条件実行される 1 以上の各タスクに対して、それぞれの実行確率を定義できることを特徴とする請求項 18 に記載のコンピュータソフトウェアシステム設計支援装置。

[請求項20] 前記並列実行処理には、並列に実行される全てのタスクが終了した場合のみ、その後の処理が進むパルドタイプ並列実行と、並列実行される全てのタスクが終了しなくても進むスプリットタイプ並列実行と、を備えたことを特徴とする請求項 18 に記載のコンピュータソフトウェアシステム設計支援装置。

【図1】


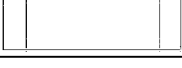






【図2】

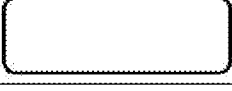
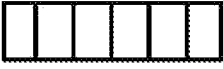


[図3]

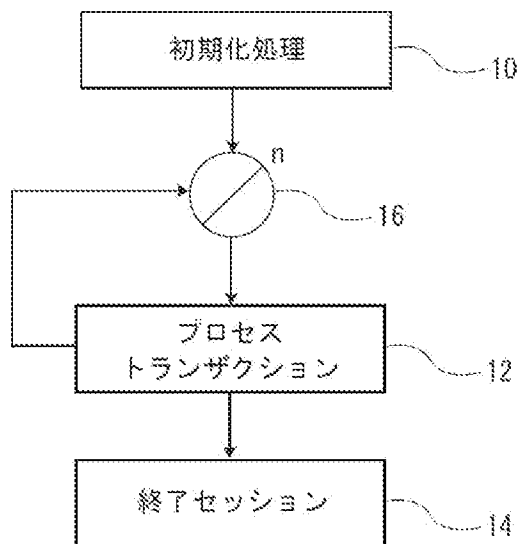
(1)

タスク名	シンボル	説明
基本タスク		現在の開発ステージに適している最も低いレベルの詳細で処理ステップを表現
拡張タスク		サブグラフに対応する
繰り返しタスク		n回対応しているタスクが繰り返し実行される ループはこのタスクに対して戻り矢印がある
ケースタスク		条件により実行される要素を定義 付属するタスクが条件によって実行される 各タスクは実行確率をもつ
バルドタスク		付属するタスクは並列実行される 全てのタスクが終了しないと進まない
スプリットタスク		付属するタスクは新しい処理スレッドを表現する 全てのタスクが終了しなくても進む

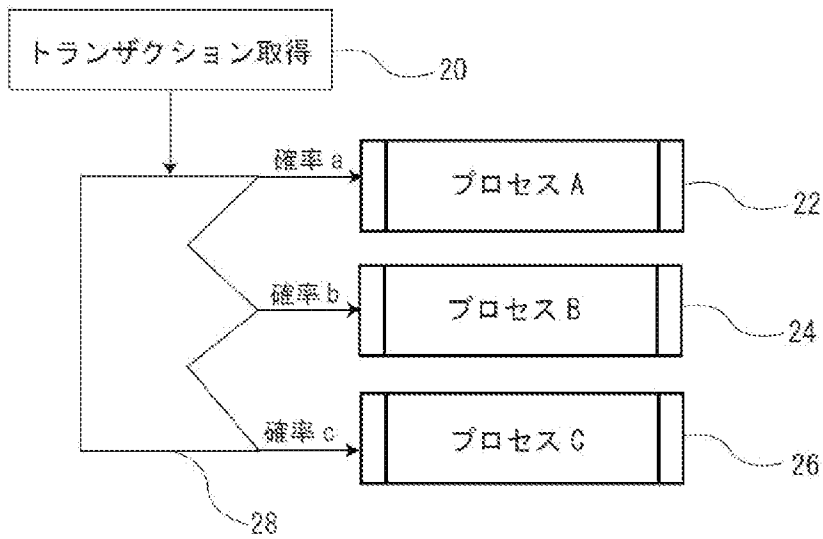
(2)

共有データ		複数の実行処理で共通アクセスされるデータ。タイムアウト指定付きの排他制御を必要とする。
メッセージ通信		ある実行処理から別の処理に渡されるメッセージ。タイムアウト指定など可能。

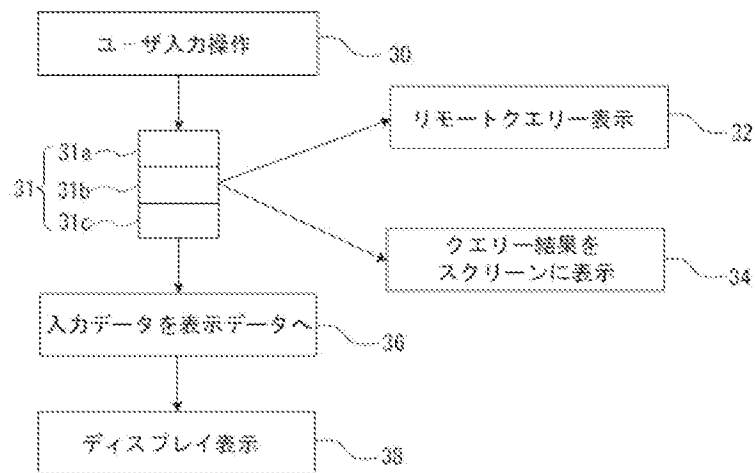
[図4]



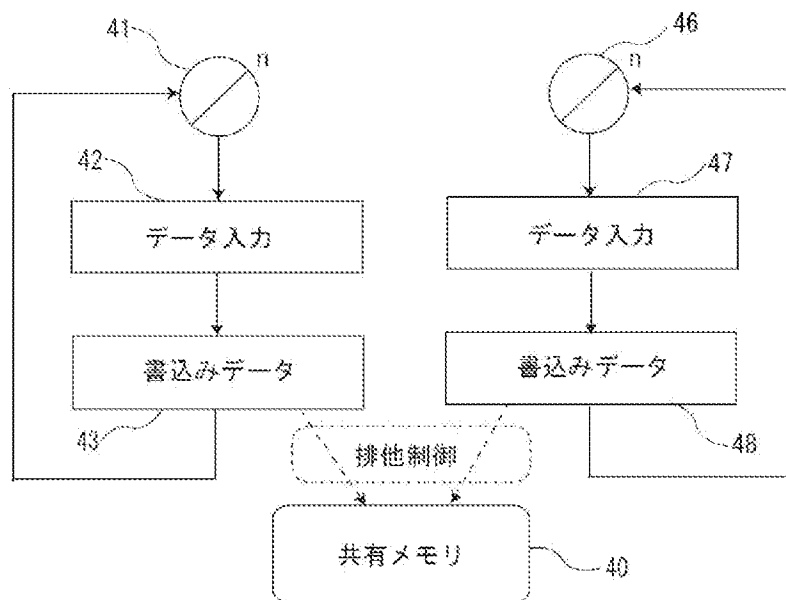
【図5】



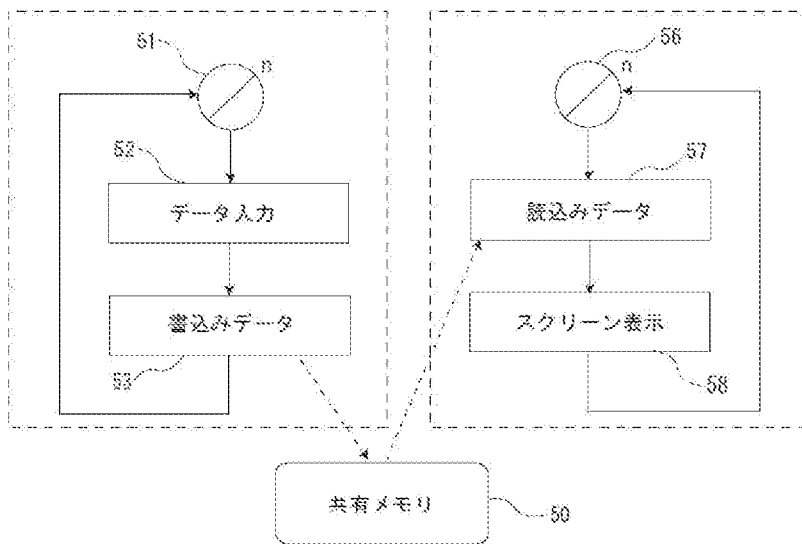
【図6】



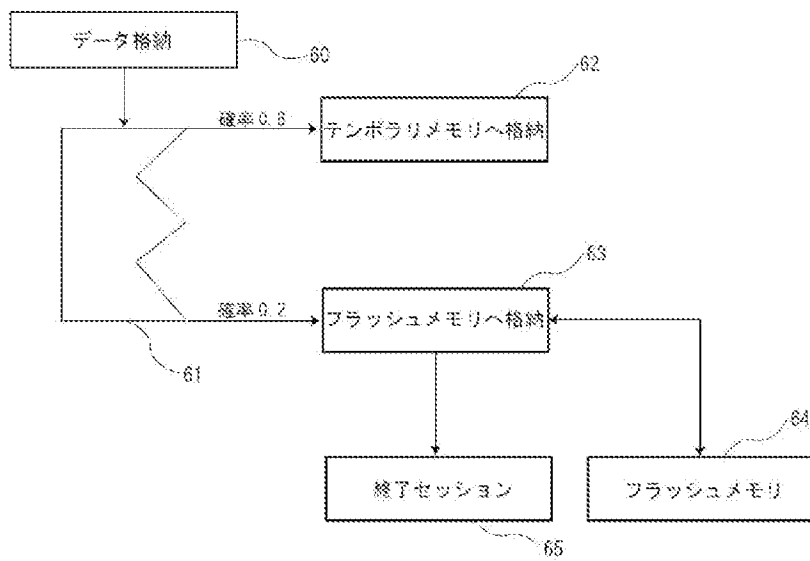
【図7】



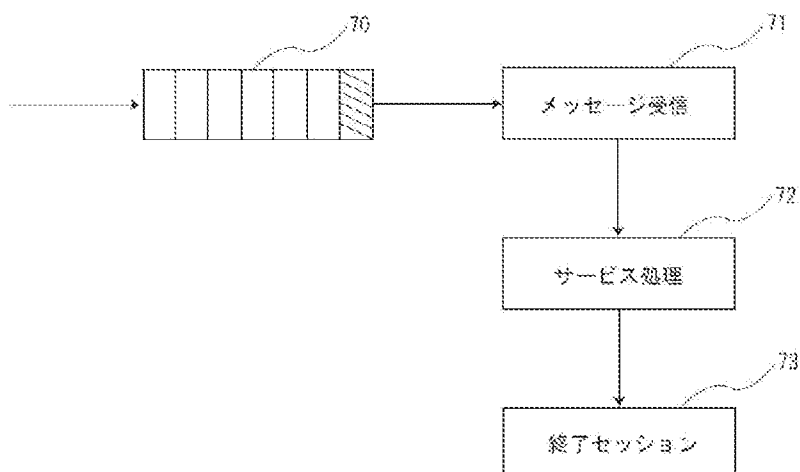
【図8】



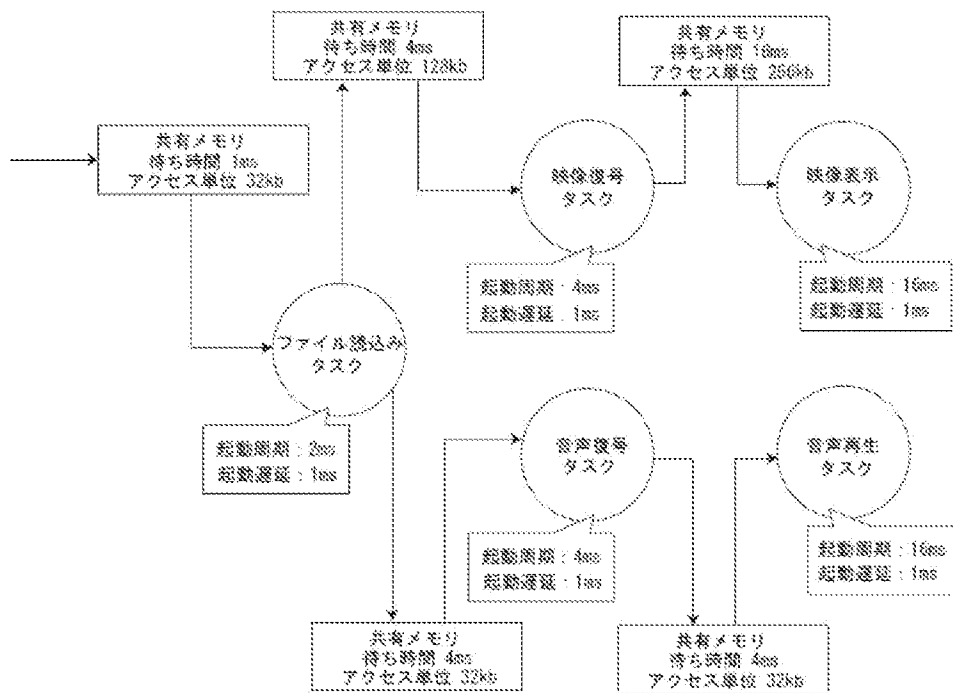
【図9】



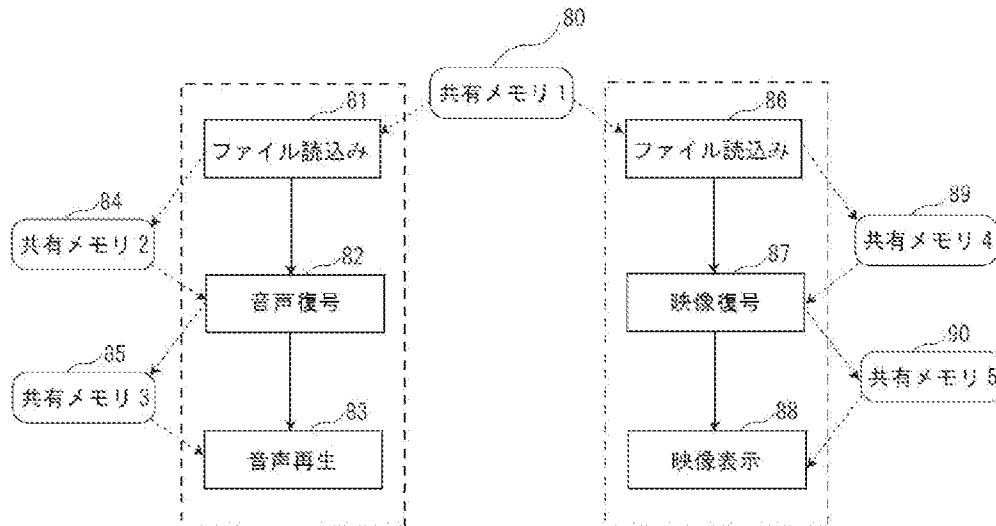
【図10】



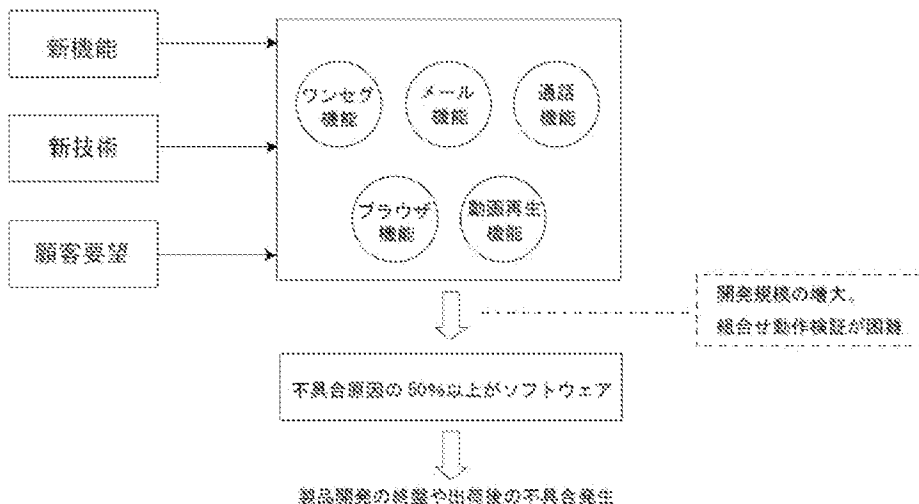
[図11]



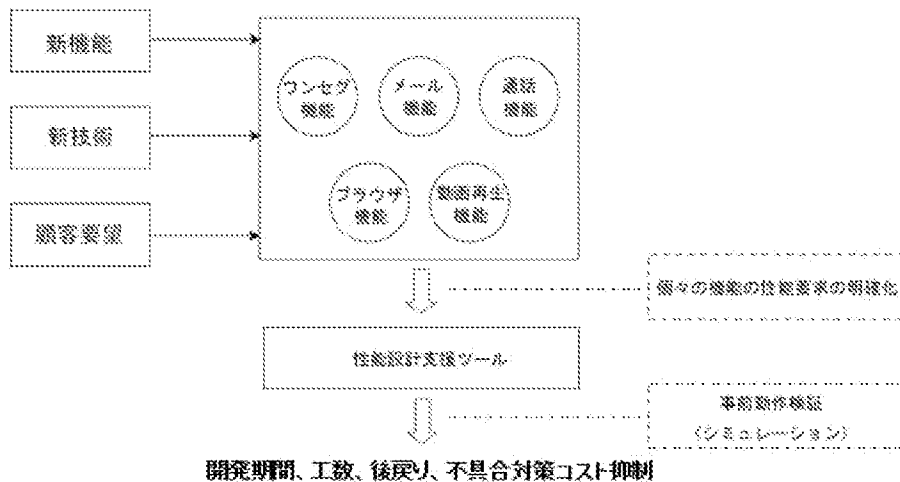
[図12]



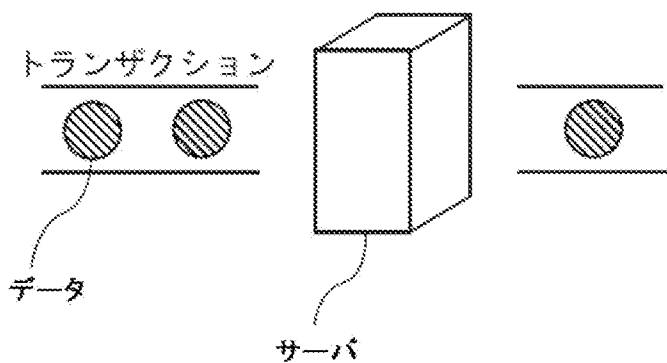
[図13]



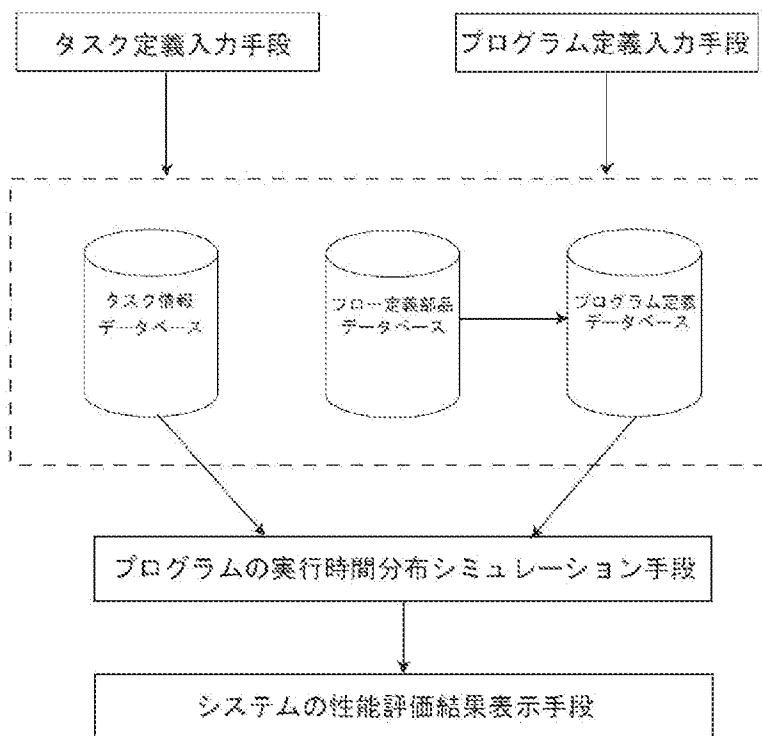
[図14]



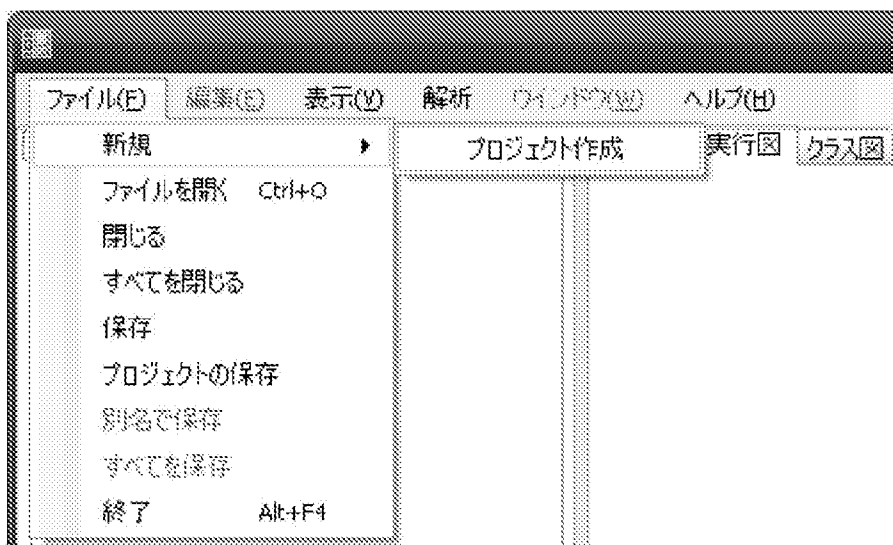
[図15]



[図16]



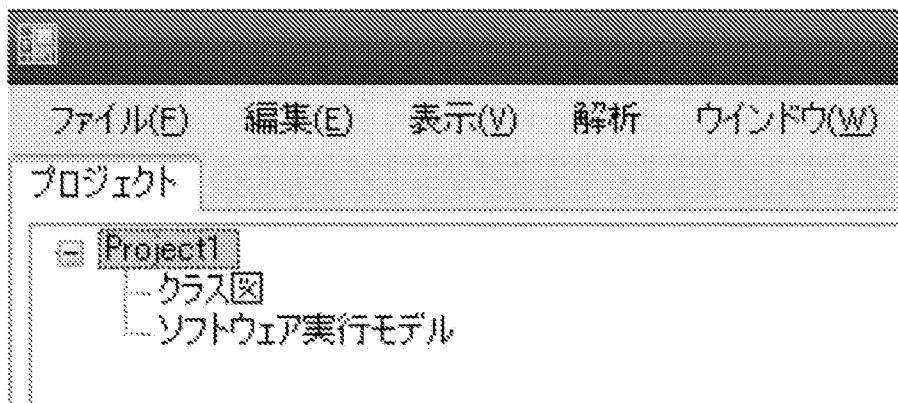
[図17]



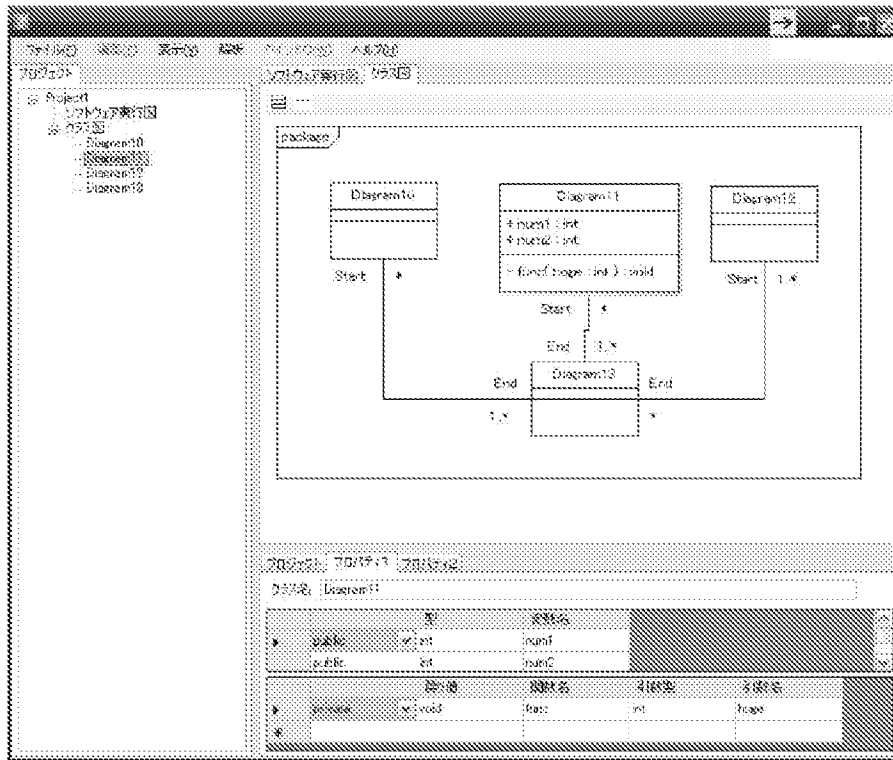
[図18]



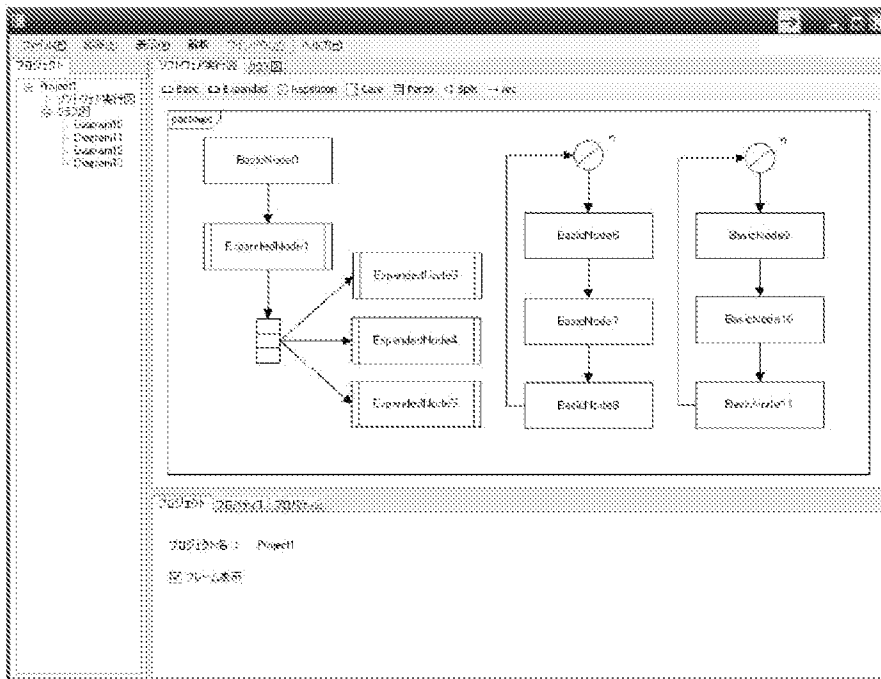
[図19]



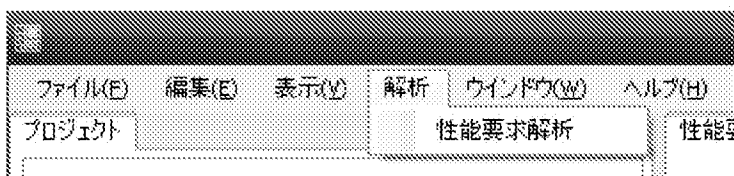
[図20]



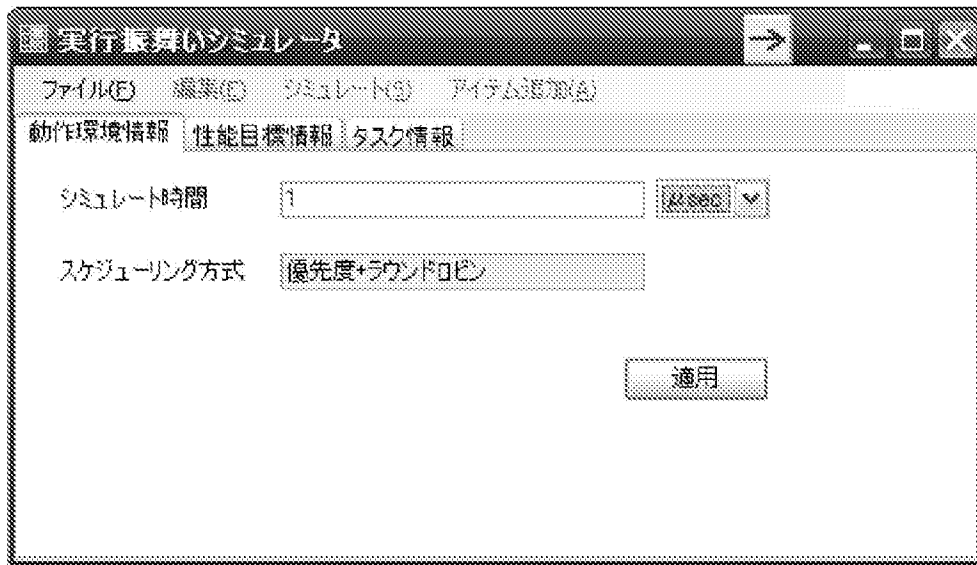
[図21]



[図22]



[図23]



[図24]

性能目標情報		
プログラム名	時間制約	タスク情報
program 1	30000000	Playbutton, FileReader
program 2	80000	DecodeVideo
program 3	1800000	DecodeAudio
program 4	33333	Timing

[図25]

性能目標情報

プログラム名

時間制約 ▼

プログラムに含まれるタスク情報

Playbutton
FileReader

[図26]

タスク情報

タスク名	優先度	起動オフセット	起動周期	属性	実行時間	タイムスライス	実行回数
Playbutton	2	0 μ sec	0 μ sec	リアルタイム	2.481	0 [μ sec]	1
FileReader	1	Playbutton	0 μ sec	リアルタイム	21743176.77	0 [μ sec]	1
DecodeVideo	0	FileReader	20000 μ sec	ラウンドロビン	68736.52844	5000 [μ sec]	1
DecodeAudio	0	FileReader	180000 μ sec	ラウンドロビン	442.8258125	5000 [μ sec]	1
Timing	1	FileReader	33333 μ sec	リアルタイム	645.58225	0 [μ sec]	1

[図27]

タスク情報

タスク名

優先度

起動同期 ▼

実行時間と分布情報

	実行時間	単位	分布情報
	20	usec ▼	0.6
	12	usec ▼	0.3
▶	31	usec ▼	0.1
*			

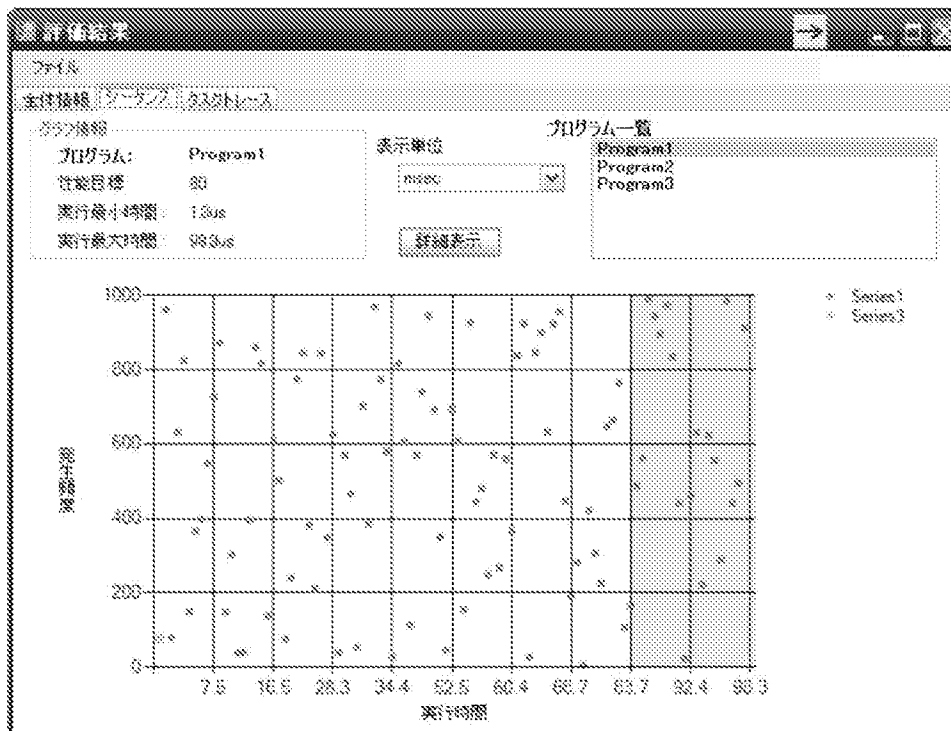
実行回数

OK キャンセル

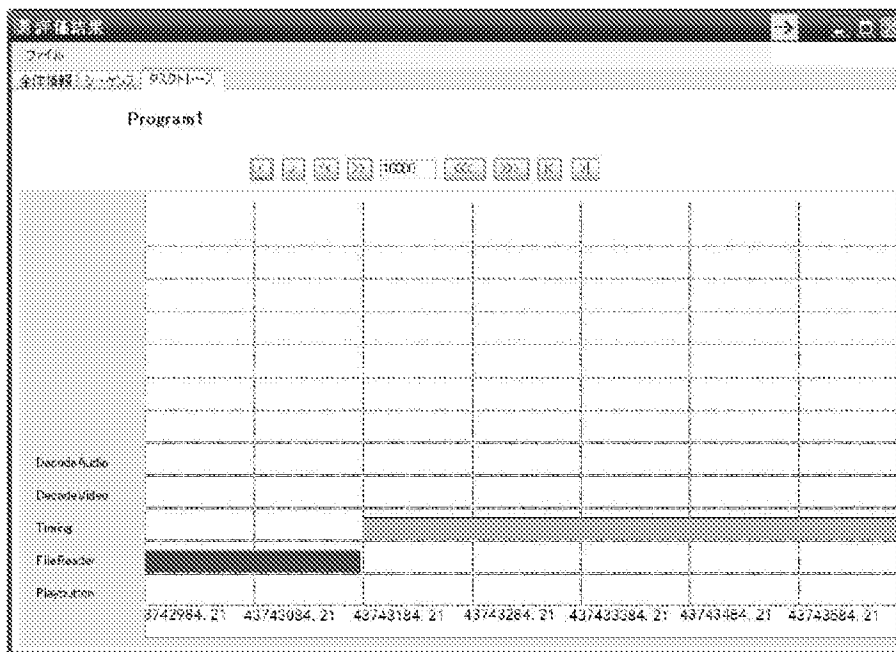
[図28]



[図29]



[図30]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/001478

A. CLASSIFICATION OF SUBJECT MATTER

G06F11/36(2006.01) i, G06F11/34(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F11/36, G06F11/34

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2012
Kokai Jitsuyo Shinan Koho	1971-2012	Toroku Jitsuyo Shinan Koho	1994-2012

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2002-268879 A (Mitsubishi Electric Corp.), 20 September 2002 (20.09.2002), paragraphs [0006] to [0010], [0032] to [0043], [0045] to [0049], [0062] to [0065] (Family: none)	1-20
A	JP 2000-47884 A (Toshiba Corp.), 18 February 2000 (18.02.2000), abstract; paragraphs [0021] to [0022] (Family: none)	1-20
A	JP 2006-185055 A (Toshiba Corp. et al.), 13 July 2006 (13.07.2006), paragraphs [0026] to [0031] (Family: none)	1-20

 Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
23 March, 2012 (23.03.12)Date of mailing of the international search report
03 April, 2012 (03.04.12)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/001478

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Yukikazu NAKAMOTO, Noriyoshi ICHINOSE, Kazuhiro TERAMINE, Tomomi UENO, Masahiro IWANAGA, Nobukazu MIYATA, Mikio NAKAYAMA, Kiyoshi YONEDA, "Real Time Seino Kaiseki System ProBA/RTH", NEC Technical Journal, 15 July 1994 (15.07.1994), vol.47, no.6(whole no.304), pages 42 to 43	1-20

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int.Cl. G06F11/36(2006.01)i, G06F11/34(2006.01)i

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int.Cl. G06F11/36, G06F11/34

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国实用新案公報	1922-1996年
日本国公開实用新案公報	1971-2012年
日本国实用新案登録公報	1996-2012年
日本国登録实用新案公報	1994-2012年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2002-268879 A (三菱電機株式会社) 2002.09.20, 第6~10, 32~43, 45~49, 62~65段落 (ファミリーなし)	1-20
A	JP 2000-47884 A (株式会社東芝) 2000.02.18, 要約欄, 第21~22段落 (ファミリーなし)	1-20
A	JP 2006-185055 A (株式会社東芝 外1名) 2006.07.13, 第26~31段落 (ファミリーなし)	1-20

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
 「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
 「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
 「O」口頭による開示、使用、展示等に言及する文献
 「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献
 「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
 「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
 「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
 「&」同一パテントファミリー文献

国際調査を完了した日

23.03.2012

国際調査報告の発送日

03.04.2012

国際調査機関の名称及びあて先

日本国特許庁 (ISA/J P)
 郵便番号100-8915
 東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

坂庭 剛史

電話番号 03-3581-1101 内線 3545

5B

9288

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	中本幸一、市瀬規善、寺峯和裕、上野智美、岩永雅洋、宮田宣和、中山幹大、米田 潔, リアルタイム性能解析システム P r o B A / R T H, N E C 技報, 1994.07.15, V o l . 4 7, N o . 6 (通巻304号), p p . 4 2 ~ 4 3	1-20