

(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
2025年1月16日(16.01.2025)



(10) 国際公開番号

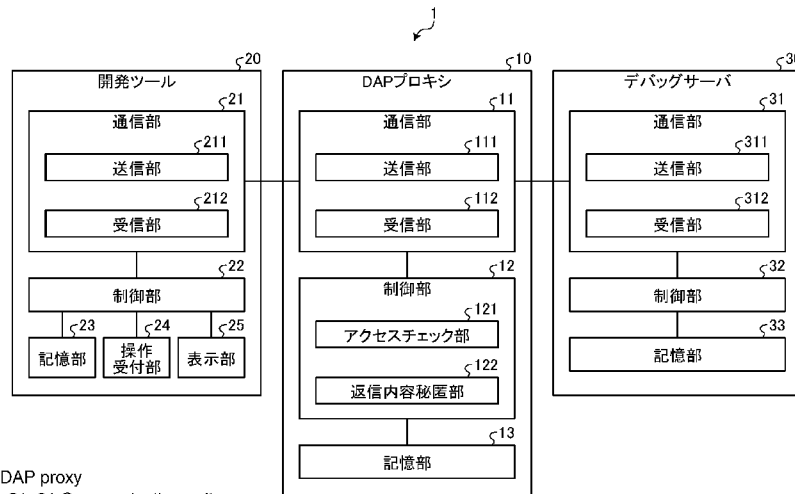
WO 2025/013262 A1

- (51) 国際特許分類:
G06F 21/60 (2013.01) *G06F 21/57* (2013.01)
- (21) 国際出願番号: PCT/JP2023/025785
- (22) 国際出願日: 2023年7月12日(12.07.2023)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人: 日本電信電話株式会社 (NIPPON TELEGRAPH AND TELEPHONE CORPORATION) [JP/JP]; 〒1008116 東京都千代田区大手町一丁目5番1号 Tokyo (JP).
- (72) 発明者: 斎藤 忍(SAITO, Shinobu); 〒1808585 東京都武蔵野市緑町3丁目9-11 N T T 知的財産センタ内 Tokyo (JP).

- (74) 代理人: 弁理士法人酒井国際特許事務所 (SAKAI INTERNATIONAL PATENT OFFICE); 〒1000013 東京都千代田区霞が関3丁目8番1号 虎ノ門ダイビルイースト Tokyo (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, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK,

(54) Title: CONTROL DEVICE, CONTROL METHOD, AND CONTROL PROGRAM

(54) 発明の名称: 制御装置、制御方法および制御プログラム



10 DAP proxy
11, 21, 31 Communication unit
12, 22, 32 Control unit
13, 23, 33 Storage unit
20 Development tool
24 Operation acceptance unit
25 Display unit
30 Debugging server
111, 211, 311 Transmission unit
112, 212, 312 Reception unit
121 Access check unit
122 Reply content concealment unit

(57) Abstract: A DAP proxy (10) comprises: a reception unit (112) that receives, from a development tool (20) for performing a debugging process on program code, a command relating to the debugging process for the code; and an access check unit (121) that, on the basis of the command received from the development tool (20) by the reception unit (112) and content relating to access to the code, skips an operation leading to access to confidential code when an access condition is not satisfied.

[続葉有]

SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA,
UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(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, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, 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, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第21条(3))

(57) 要約: D A P プロキシ (10) は、プログラムのコードに対してデバッグ処理を行うための開発ツール (20) から、コードに対するデバッグ処理に関するコマンドを受信する受信部 (112) と、受信部 (112) によって開発ツール (20) から受信されたコマンドおよびコードへのアクセスに関する内容に基づいて、アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせるアクセスチェック部 (121) と、を有する。

明 細 書

発明の名称： 制御装置、制御方法および制御プログラム

技術分野

[0001] 本発明は、制御装置、制御方法および制御プログラムに関する。

背景技術

[0002] ソフトウェア開発の現状として、フリーランスやクラウドソーシングの活用、リモートワークの推進により、企業の管理区画外（自宅やシェアオフィス等）でのプログラムの編集、デバッグ等の開発の遂行が増加している。例えば、通常はプログラムのコードをデバッグするためのデバッグ装置（プログラムの編集、デバッグ等の開発を行うための開発ツール等）によって提供されるデバッグ支援機能を、デバッグ装置とは独立したサーバから提供可能にするための通信処理等を行うデバッグ処理の公開プロトコル（D A P : Debug Adapter Protocol）が近年公開されている。D A P の活用例として、D A P によって、デバッグ装置等によって提供されるデバッグ処理等の処理を停止する箇所であるブレイクポイント設定や、コードの関数の中に入るステップイン等のデバッグ支援機能を、デバッグ装置から独立したサーバ（以降、デバッグサーバ）からの提供を可能とする例が存在する（例えば、非特許文献1 参照）。

先行技術文献

非特許文献

[0003] 非特許文献1 : Debug Adapter Protocol, [online] , [2023年6月19日検索] , インターネット<<https://microsoft.github.io/debug-adapter-protocol/>>

発明の概要

発明が解決しようとする課題

[0004] しかし、従来技術では、開発者に機密コードを開示することなく、プログラムのコードのデバッグ処理を行わせることができない場合がある。例えば

、従来技術では、企業によって保有されていたり、顧客から委託されたりするプログラムに含まれる機密コードが、リモートでコードのデバッグ処理等を行う開発者によって閲覧されうるセキュリティ上の懸念がある。また、開発者が外部の攻撃者ではない開発プロジェクトに所属するリモート開発者であっても、開発者によるオペレーションミス等により、外部のWebサイトや外部のリポジトリに、機密コードが公開されてしまうコードの漏洩のリスクが高い。

課題を解決するための手段

[0005] 上述の課題を解決するため、本発明は、プログラムのコードに対してデバッグ処理を行うためのデバッグ装置から、当該コードに対する前記デバッグ処理に関するコマンドを受信する受信部と、前記受信部によって前記デバッグ装置から受信された前記コマンドおよび前記コードへのアクセスに関する内容に基づいて、前記アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせる制御部と、を有する。

発明の効果

[0006] 本発明によれば、開発者に機密コードを開示することなく、プログラムのコードのデバッグ処理を行わせることができる。

図面の簡単な説明

[0007] [図1]図1は、制御システムの概要の一例について説明するための図である。

[図2]図2は、機密コードの一例を示す図である。

[図3]図3は、制御システムの概要の一例について説明するための図である。

[図4]図4は、制御システムの概要の一例について説明するための図である。

[図5]図5は、制御システムの構成の一例を示す図である。

[図6]図6は、制御システムによって実行される処理の流れの一例を示すフローチャートである。

[図7]図7は、制御プログラムを実行するコンピュータの構成の一例を示す図である。

[図8]図8は、参考技術に係る制御システムの概要の一例について説明するた

めの図である。

[図9]図9は、参考技術に係る制御システムの概要の一例について説明するための図である。

[図10]図10は、参考技術に係る制御システムの概要の一例について説明するための図である。

発明を実施するための形態

[0008] 以下、図面を用いて本発明の実施形態について説明するが、本発明は、以下の実施形態に限定されない。また、図面の記載は、同一部分には同一の符号を付しており、重複する説明は省略され、同様の機能を有する部材および同様の処理の説明も省略される。

[0009] [0. 参考技術]

[概要の一例]

本実施形態について説明する前に、図8～図10を用いて、参考技術に係る制御システム1#の概要の一例について説明する。図8～図10は、参考技術に係る制御システムの概要の一例について説明するための図である。図8～図10に示される例では、制御システム1#は、コードのデバッグ処理を行う。図8～図10に示される例では、制御システム1#は、デバッグ支援機能（ブレイクポイント設定やステップイン等）を、開発ツール20と、開発ツール20とは独立したデバッグサーバ30と、DAP10#とを有する。DAP10#は、デバッグサーバ30のコードに対するデバッグ処理を提供可能にするための通信処理を行うためのデバッグ処理の公開プロトコルである。

[0010] [デバッグ支援機能の利用の一例]

まず、図8を用いて、制御システム1#によるデバッグ支援機能の利用の一例について説明する。図8に示される例では、制御システム1#のDAP10#は、あらかじめ、参照用のソースコードを開発ツール20およびデバッグサーバ30に送信する。ソースコードとは、例えば、プログラミング言語等の人間が理解、記述しやすい言語やデータ形式によって書き記されたコ

ンピュータプログラムのことをいう。

[0011] 例えば、制御システム 1 #の開発ツール 20は、開発者の端末のGUI (Graphic User Interface) や、およびインターネット等のネットワークを介して、開発者からデバッグ操作を受け付ける (図8の(0)デバッグ操作 (ブレイクポイントの設定、コード実行、ステップイン要求等))。次に、DAP 10 #は、例えば、開発ツール 20から受信したデバッグ処理に関するコマンド (命令) をデバッグサーバ 30に送信する (図8の(1)コマンド送信 (ブレイクポイントの設定、コード実行、ステップイン要求等))。

[0012] 次に、図8に示される例では、デバッグサーバ 30は、コマンドの内容に含まれるコードへのステップインおよびブレイクポイントでの停止等のデバッグ処理を実行し、デバッグ処理の結果を開発ツール 20側 (DAP 10 #) に返信する (図8の(2)デバッグサーバ側で、デバッグ処理を実行し、結果を開発ツール側に提供)。

[0013] 次に、図8に示される例では、DAP 10 #は、デバッグサーバ 30から受信したコマンドに対する返信を開発ツール 20に対して行う (図8の(3)コマンド返信 (停止イベント通知、スタックフレーム等))。例えば、DAP 10 #は、デバッグ処理を停止する停止イベント通知、および、ブレイクポイントの直前までに実行された関数の情報であって、呼ばれた関数の名前等が含まれうるスタックフレーム等に関する返信を開発ツール 20に対して行う。次に、図8に示される例では、開発ツール 20は、コマンドに対する返信内容であるデバッグ情報を表示する (図8の(4)デバッグ情報表示)。

[0014] [DAPを利用した機密コードの分離の概要の一例]

図9を用いて、制御システム 1 #によるDAPを利用した機密コードの分離の概要の一例について説明する。図9に示される例では、制御システム 1 #のDAP 10 #は、機密コードを分離する。例えば、DAP 10 #は、図8に示される例と異なり、開発ツール 20に機密コードが含まれないソースコード (図9のソースコード (機密コードなし)) を送信し、デバッグサー

バ30に機密コードが含まれる全てのソースコード（図9のソースコード（全体））を送信する。例えば、開発ツール20は、DAP10#を介してデバッグサーバ30とコマンドのやりとりを通じてデバッグ処理を実行する。

[0015] ここで、図9に示される例では、デバッグサーバ30は、デバッグ処理を実行するための実行ファイルを生成するため、ソースコード（全体）を保持する必要がある。一方、図9に示される例では、ソースコード（機密コードなし）には、デバッグ処理は実行されないため、開発ツール20は、デバッグ処理を実行するための実行ファイルを生成（ビルド）せず、ソースコード（全体）を保持する必要がある。このため、図9に示される例では、制御システム1#は、DAP10#を利用することによって、開発者側に機密コードを物理的に配置しなくても、デバッグ処理を行わせることができるようになる。

[0016] [課題の一例]

次に、図10を用いて、参考技術の課題の一例について説明する。上述のように制御システム1#がDAP10#を利用した場合、参考技術には、開発者に機密コードを開示することなく、プログラムのコードのデバッグ処理を行わせることができない場合がある。例えば、参考技術では、コマンドの種類によっては、デバッグサーバ30からのコマンドに対する返信内容に機密コードの情報も含まれており、この返信内容から機密コードの情報漏洩が起こってしまう場合がある。

[0017] 一例として、スタックトレースの要求（例えば、予めブレイクポイントが設定されている場合や、プログラムにエラーが発生した場合等により、実行中のコンピュータプログラムが停止した際に、停止の直前まで実行されていた関数やメソッド等の履歴情報を表示させる要求）が含まれるコマンドに対する返信内容がスタックフレームである場合、スタックフレームに機密コードに対するデバッグ処理の実行時の関数の名前等が含まれる場合がある。図10に示される例では、NGClassにおいて停止した場合のスタックフレームでは、直前の機密コードであるNGClassが露見する。それ以外にもデバッグ処理

の停止に至るまでに、例えば、SecretFramework等の機密コードに対するデバッグ処理が実行された場合、このような機密コードも露見してしまう。別の一例として、開発ツール20側に機密コードが存在しなくても、開発者はデバッグ情報から機密コードに対するデバッグ処理の実行時の情報を理解できれば、デバッグサーバ30に実装される機密コードの内容を容易に推測できるため、機密コードの情報が漏洩してしまう。

[0018] 本実施形態は、上述の課題を解決するためになされたものであり、開発者に機密コードを開示することなく、プログラムのコードのデバッグ処理を行わせることができる。

[0019] [1. 実施形態]

[概要の一例]

以下、図1～図4を用いて、実施形態に係る制御システム1の概要の一例について説明する。図1、図3および図4は、制御システムの概要の一例について説明するための図である。図2は、機密コードの一例を示す図である。図1～図4に示される例では、制御システム1は、参考技術に係るDAP10#の代わりに、DAPプロキシ10を有する。

[0020] 図1～図4に示される例では、制御システム1は、DAPプロキシ10によって、開発ツール20からコマンドを受信し、コマンドおよびコードへのアクセスに関する内容に基づいて、アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせる。ここで、アクセスの条件とは、例えば、コードにアクセス可能な条件のことをいう。一例として、アクセスの条件とは、アクセス先のコードが機密コードでないコード（例えば、後述のアクセスすることができるAllow List（許可リスト＝公開可能コードのリスト）として設定されているOKClassおよびOKClass2等）であることをいう。これにより、デバッグ情報の中に機密コードの実行時の情報が含まれなくなるため、制御システム1は、機密コードの実装内容を開発者に見せることも把握させることもなく、デバッグ処理を行わせることができる。

[0021] ここで、DAPプロキシ10とは、例えば、開発ツール20とデバッグサーバー30との中継機構であり、参考技術のDAP10#に対応するものをいう。開発ツール20とは、例えば、プログラムのコードに対してデバッグ処理を行うためのデバッグ装置のことをいう。コマンドとは、例えば、コードに対するデバッグ処理に関するコマンドのことをいう。スキップとは、例えば、機密コードへのアクセスにつながる操作を省略することや、本来やるべき処理をやらないという意味で禁止することをいう。機密コードとは、例えば、開発者に対して、関数の中身等を非公開にすべきコードのことをいう。

[0022] また、図1～図4に示される例では、制御システム1は、DAPプロキシ10によって、コマンドに対する返信の内容に、機密コードの情報漏洩につながる内容が含まれる場合、機密コードの情報漏洩につながる内容を秘匿にするか、または削除する。これにより、デバッグ情報の中に機密コードの実行時の情報が含まれなくなるため、制御システム1は、機密コードの実装内容を開発者に見せることも把握させることもなく、デバッグ処理を行わせることができる。なお、図1～図4に示される例は一例にすぎず、本実施形態に係る制御システム1は、図1～図4に示される例に限定されない。本実施形態に係る制御システム1は、DAPプロキシ10によって、開発ツール20からコマンドを受信し、コマンドおよびコードへのアクセスに関する内容に基づいて、アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせるものであればよい。

[0023] [機密コードの一例]

次に、図2を用いて機密コードの一例について説明する。プログラムの開発では、ソースコードを解析し、コンピュータが直接実行可能な形式のプログラムに変換するコンパイルを行い、プログラムを実行することが要求される。機密コードも、開発過程において、別コードから呼び出される場合がある。

[0024] そこで、図1～図4に示される例では、制御システム1は、機密コードを内部隠ぺいコードと完全隠ぺいコードとの二つに分類する。図2に示される

ように、内部隠ぺいコードとは、例えば、関数の中身が非公開である一方で、関数の呼び出しは許可されているコードのことをいう。すなわち、内部隠ぺいコードとは、例えば、開発者がプログラミングをする上で、呼び出せるコードであり、開発者に対して、関数の存在そのものは秘匿にしないコードのことをいう。完全隠ぺいコードとは、例えば、関数の中身が非公開であることに加えて、関数の呼び出しが許可されていないコードのことをいう。すなわち、完全隠ぺいコードとは、例えば、開発者がプログラミングをする上で、呼び出せないコードであり、開発者に対し、関数の存在そのものも秘匿にするコードのことをいう。図1～図4に示される例では、制御システム1は、どちらのコードであっても、開発過程において実装内容を秘匿することを目的とする。

[0025] [機密コードへのアクセスにつながる操作のスキップの一例]

図1および図3を用いて、制御システム1による機密コードへのアクセスにつながる操作のスキップの一例について説明する。図3は、制御システム1による機密コードへのアクセスにつながる操作のスキップ処理の流れの一例を示すシーケンス図でもある。

[0026] 図1に示される例では、DAPプロキシ10のアクセスチェック部121は、コマンドの送受信内容を解析し、機密コードへのアクセスにつながる操作等、機密コードの漏洩につながる危険な操作をスキップさせる。

[0027] 例えば、アクセスチェック部121は、開発ツール20から受信されたコマンドおよびコードへのアクセスに関する内容をチェックし、アクセスの条件が満たされない場合、デバッグサーバ30に対して、機密コードへのステップイン等の機密コードへのアクセスにつながる操作をスキップさせる。例えば、アクセスチェック部121は、受信部112によって開発ツール20から受信されたコマンドに機密コードへのアクセスにつながる操作が含まれる場合、アクセスの条件が満たされていないと判断し、この操作をスキップさせる。一例として、アクセスチェック部121は、開発ツール20から送信されたコマンドの内容を分析し、コマンドにRefNGClass等の内部隠ぺいコ

ードへのステップイン等の操作が含まれる場合、アクセスの条件が満たされていないと判断し、この操作をスキップさせる。これにより、デバッグ情報の中に機密コードの実行時の情報が含まれなくなるため、アクセスチェック部121は、機密コードの実装内容を開発者に見せることも把握させることもなくデバッグ処理を行わせることができる。

[0028] 次に、アクセスチェック部121は、例えば、DAPプロキシ10に、スキップされた操作について開発ツール20側に返答させずに、デバッグサーバ30に、自動的に次のコードの行（ステートメント）に移動させて、デバッグ処理を実行させる。次に、アクセスチェック部121は、例えば、DAPプロキシ10に、デバッグサーバ30によるデバッグ処理の結果を開発ツール20に返信する。

[0029] 図3に示される例では、DAPプロキシ10には、コードへのアクセスの条件が定義されたアクセス条件定義ファイルによって、アクセスすることができるAllow list（許可リスト＝公開可能コードのリスト）として、OKClassおよびOKClass2があらかじめ設定されている。また、図3に示される例では、DAPプロキシ10には、アクセス条件定義ファイルによって、アクセスできない場合があるExecutable list（実行可能リスト＝内部隠ぺいコードのリスト）として、RefNGClassがあらかじめ設定されている。

[0030] 図3に示される例では、まず、開発ツール20は、DAPプロキシ10にステップインを要求するステップインリクエストの送信を行う。例えば、開発ツール20は、RefNGClassへのステップインリクエストの送信を行う。次に、DAPプロキシ10は、例えば、ステップインリクエストをデバッグサーバ30に転送する。次に、デバッグサーバ30は、例えば、ステップインリクエストに含まれるコードのうち、1行分（1つ）のコードに対してデバッグ処理を実行して、停止イベントをDAPプロキシ10に通知する。

[0031] 次に、DAPプロキシ10は、例えば、スタックトレース情報を取得するためのスタックトレース情報取得リクエストをデバッグサーバ30に送信する。次に、デバッグサーバ30は、例えば、スタックフレームをDAPプロ

キシ10に送信する。一例として、デバッグサーバ30は、RefNGClassにステップインしたので、一番上にRefNGClassがきているスタックフレームをDAPプロキシ10に送信する。次に、DAPプロキシ10のアクセスチェック部121は、例えば、一番上のスタックであるRefNGClassは公開不可なので、ステップインをスキップさせる。次に、DAPプロキシ10は、例えば、強制的に停止されている関数（RefNGClass）の外（呼び出し元=OKClass）に出るステップアウトのリクエストをデバッグサーバ30に送信する。

[0032] 次に、デバッグサーバ30は、例えば、この関数の外に出て、DAPプロキシ10に対して停止イベントの通知を行う。次に、DAPプロキシ10は、例えば、開発ツール20に対して停止イベントの通知を行う。一例として、DAPプロキシ10は、開発ツール20側には、RefNGClassの内部には入らず（ステップインせず）、ステップオーバーを実施したことを通知する。

[0033] [機密コードの情報漏洩につながる内容の秘匿の一例]

次に、図1および図4を用いて、制御システム1による機密コードの情報漏洩につながる内容の秘匿の一例について説明する。図4は、制御システム1による機密コードの情報漏洩につながる内容の秘匿処理の流れの一例を示すシーケンス図でもある。図1に示される例では、DAPプロキシ10の返信内容秘匿部122は、コマンドに対する返信の内容に、機密コードの情報漏洩につながる内容が含まれる場合、機密コードの情報漏洩につながる内容を秘匿にするか、または削除する。例えば、返信内容秘匿部122は、アクセス条件定義ファイルの許可リストに無いNGClass、RefNGClassおよびSecret Framework等の関数のスタックフレームを削除する。

[0034] 図4に示される例では、図3に示される例と同様に、DAPプロキシ10には、アクセス条件定義ファイルによって、アクセスすることができるAllow listとして、OKClassおよびOKClass2があらかじめ設定され、アクセスすることができないExecutable listとして、RefNGClassがあらかじめ設定されている。まず、開発ツール20は、例えば、DAPプロキシ10にブレイクポイントの設定を要求するブレイクポイント設定リクエストの送信を行う。

次に、DAPプロキシ10は、例えば、デバッグサーバ30にブレイクポイント設定リクエストの転送を行う。次に、開発ツール20は、例えば、DAPプロキシ10にデバッグ起動リクエストの送信を行う。

[0035] 次に、DAPプロキシ10は、例えば、デバッグサーバ30にデバッグ起動リクエストの転送を行う。次に、デバッグサーバ30は、例えば、DAPプロキシ10に（ブレイクポイントでの）停止イベントの通知を行う。次に、DAPプロキシ10は、例えば、デバッグサーバ30にスタックトレース情報取得リクエストの送信を行う。次に、デバッグサーバ30は、例えば、スタックフレームの返信を行う。

[0036] 次に、図4に示される例では、DAPプロキシ10の返信内容秘匿部122は、スタックフレームに機密コードが含まれる場合、機密コードを秘匿にするか、または削除する。一例として、返信内容秘匿部122は、許可リストに無いNGClass、RefNGClassおよびSecretFramework等の関数のスタックフレームを削除する。別の一例として、返信内容秘匿部122は、許可リストに無いNGClass、RefNGClassおよびSecretFramework等の関数のスタックフレームを、これらのスタックフレームを覆い隠すマスクによって秘匿にする。次に、DAPプロキシ10は、例えば、開発ツール20に（ブレイクポイントでの）停止イベントの通知を行う。また、DAPプロキシ10は、例えば、開発ツール20にスタックフレームの返信を行う。一例として、DAPプロキシ10は、開発ツール20に機密コードのスタックフレームが削除された情報を返信する。

[0037] [制御システムの構成の一例]

図5を用いて、実施形態に係る制御システム1の構成の一例について説明する。図5は、制御システムの構成の一例を示す図である。図5に示される例では、制御システム1は、DAPプロキシ10と、開発ツール20と、デバッグサーバ30とを有する。

[0038] [DAPプロキシの構成の一例]

DAPプロキシ10は、開発ツール20からコマンドを受信し、コマンド

およびコードへのアクセスに関する内容に基づいて、アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせる制御装置である。図5に示される例では、制御装置は、デバッグ処理の公開プロトコルであるDAPプロキシ10であるが、上述の機能を有するものであれば、特に限定されない。ただし、制御装置は、DAPプロキシ10に適用されることにより、参画プロジェクト(PJ: Project)によって採用されるプログラミング言語や開発環境によらず、PJが変更されても、開発者自身の開発ツールを切り替えずに、デバッグ支援機能を利用できる。図5に示される例では、DAPプロキシ10は、通信部11と、制御部12と、記憶部13とを有する。

[0039] [通信部]

通信部11は、開発ツール20とデバッグサーバ30との間で各種情報を送受信する。通信部11は、例えば、NIC (Network Interface Card) 等によって実現される。図5に示される例では、通信部11は、送信部111と、受信部112とを有する。

[0040] [送信部]

送信部111は、開発ツール20やデバッグサーバ30に各種情報を送信する。一例として、送信部111は、デバッグ処理に関する返信を開発ツール20に対して行う。例えば、送信部111は、機密コードが含まれないコードを開発ツール20に送信してよい。一例として、送信部111は、機密コードが含まれないソースコードを開発ツール20に送信し、機密コードが含まれるソースコードをデバッグサーバ30に送信する。すなわち、送信部111は、例えば、開発ツール20およびデバッグサーバ30の双方にソースコードを保持させつつ、開発ツール20には機密コードを保持させない。

[0041] ここで、開発者は、開発ツール20に全てのソースコード等のコードが存在しなくても、コードが送信されていれば、開発ツール20によってデバッグ処理を実行することができる。このように、機密コードを物理的に開発ツール20内に配置する必要がないため、送信部111は、機密コードが含ま

れないコードを開発ツール２０に送信することによって、機密コードを開発ツール２０から分離して機密コードの漏洩を抑制した上で、開発ツール２０によるデバッグ処理を行わせることができる。

[0042] [受信部]

受信部１１２は、プログラムのコードに対してデバッグ処理を行うための開発ツール２０から、コードに対するデバッグ処理に関するコマンドを受信する。一例として、受信部１１２は、インターネット等のネットワークを介して、開発ツール２０の通信部２１から、コードに対するデバッグ処理に関するコマンドを受信する。例えば、受信部１１２は、デバッグサーバ３０から、デバッグ処理に関する返信を受信してよい。一例として、受信部１１２は、ネットワークおよび言語毎のデバッグアダプタを介して、デバッグサーバ３０の通信部３１からコマンドに対する返信を受信する。

[0043] [制御部]

制御部１２は、ＤＡＰプロキシ１０全体を制御する。例えば、制御部１２は、各種処理手順が規定されたプログラムや、制御情報が格納された内部メモリを有する一以上のプロセッサによって構成され、プロセッサは、プログラムや内部メモリを用いて各処理を実行する。制御部１２は、例えば、ＣＰＵ (Central Processing Unit)、ＭＰＵ (Micro Processing Unit)、ＧＰＵ (Graphics Processing Unit) 等の電子回路やＡＳＩＣ (Application Specific Integrated Circuit)、ＦＰＧＡ (Field Programmable Gate Array) 等の集積回路によって実現される。図５に示される例では、制御部１２は、アクセスチェック部１２１と、返信内容秘匿部１２２とを有する。

[0044] [アクセスチェック部]

アクセスチェック部１２１は、受信部１１２によって開発ツール２０から受信されたコマンドおよびコードへのアクセスに関する内容に基づいて、アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせる。例えば、アクセスチェック部１２１は、受信部１１２に

よって開発ツール20から受信されたコマンドに、機密コードへのアクセスにつながる操作が含まれる場合、アクセスの条件が満たされていないと判断し、この操作をスキップさせる。一例として、アクセスチェック部121は、開発ツール20から送信されたコマンドの内容を分析し、コマンドにRefNGClass等の内部隠ぺいコードへのステップイン等、機密コードへのアクセスにつながる危険な操作が含まれる場合、この操作をスキップさせる。これにより、アクセスチェック部121は、デバッグ処理に伴う開発ツール20とのやりとりにおいて、機密コードの漏洩のリスクを抑制することができる。

[0045] 例えば、アクセスチェック部121は、受信部112によって開発ツール20から受信されたコマンドに、機密コードへのアクセスにつながる操作が含まれる場合、機密コードへのアクセスにつながる操作をスキップして、次のコードへのアクセスに関する内容に基づいて、デバッグ処理に関する制御を実行してよい。一例として、アクセスチェック部121は、コマンドにRefNGClass等の内部隠ぺいコードへのステップイン等の機密コードへのアクセスにつながる操作が含まれる場合、この操作をスキップして、次のコードへのアクセスに関する内容に基づいて、デバッグサーバ30に次のコードへスキップインさせる。これにより、アクセスチェック部121は、開発ツール20とのやりとりにおいて、機密コードのリスクを抑制しつつ、デバッグ処理をさせることができる。

[0046] 例えば、アクセスチェック部121は、受信部112によって開発ツール20から受信されたコマンドおよびコードへのアクセスに関する内容に基づいて、アクセスの条件が満たされる場合、デバッグ装置とは独立したデバッグサーバ30の機密コードが含まれるコードにデバッグ処理の内容を反映させてよい。一例として、アクセスチェック部121は、コマンドおよびコードへのアクセスに関する内容がAllow listとして設定されているOKClassおよびOKClass2へのステップインである場合、アクセスの条件が満たされると判断し、デバッグサーバ30に、デバッグサーバ30のソースコードにおけるOKClassおよびOKClass2へのステップインを実行させる。これにより、

アクセスチェック部121は、機密コードの漏洩のリスクが低い場合、デバッグサーバ30のコードにデバッグ処理の内容を反映させることができる。

[0047] 例えば、アクセスチェック部121は、受信部112によってデバッグサーバ30から受信された返信の内容に、機密コードへのアクセスにつながる操作が含まれる場合、機密コードへのアクセスにつながる操作をスキップさせてよい。一例として、アクセスチェック部121は、受信部112によってデバッグサーバから受信された返信の内容に、内部隠ぺいコードであるRefNGClassにステップインされたことが表されている一番上にRefNGClassがきているスタックフレームが含まれる場合、RefNGClassへのアクセスにつながるステップインをスキップさせる。例えば、アクセスチェック部121は、この場合に、デバッグサーバ30に対して、強制的に停止されている関数（RefNGClass）の外（呼び出し元=OKClass）に出るステップアウトをリクエストする。これにより、アクセスチェック部121は、デバッグ処理に伴うデバッグサーバ30とのやりとりにおいて、機密コードの漏洩のリスクを抑制することができる。

[0048] ここで、機密コードの情報漏洩につながる操作には、上述の例のような機密コードに直接的にアクセスする操作だけではなく、機密コードへの間接的なアクセスを可能にする操作も含まれうる。例えば、プログラムの実行時の情報を取得する機能（リフレクション）を有する関数（以降、リフレクション関数）へのアクセスを可能にする操作（例えば、リフレクション関数のインポート）によって、開発ツール20の開発者は、プログラム実行中の関数名（クラスやメソッド）を取得できるため、リフレクション関数を利用すると機密コード閲覧が可能となる。

[0049] そこで、アクセスチェック部121は、例えば、受信部112によって受信されたコマンドに、機密コードの情報漏洩に間接的につながる操作が含まれる場合、機密コードの情報漏洩に間接的につながる操作をスキップしてよい。一例として、アクセスチェック部121は、上述のリフレクション関数のような禁止関数へアクセスする操作を禁止する。これにより、アクセスチ

ェック部 1 2 1 は、デバッグ処理に伴う開発ツール 2 0 とのやりとりにおいて、機密コードの漏洩のリスクをより抑制することができる。

[0050] [返信内容秘匿部]

返信内容秘匿部 1 2 2 は、受信部 1 1 2 によってデバッグサーバ 3 0 から受信された返信の内容に、機密コードの情報漏洩につながる内容が含まれる場合、機密コードの情報漏洩につながる内容を秘匿にするか、または削除する。例えば、返信内容秘匿部 1 2 2 は、受信部 1 1 2 によってデバッグサーバ 3 0 から受信されたスタックフレームを分析し、許可リストに無い NGClass、RefNGClass および SecretFramework 等の関数のスタックフレームが含まれる場合、スタックフレームを覆い隠すマスクによって秘匿にするか、削除する。

[0051] これにより、返信内容秘匿部 1 2 2 は、デバッグ処理に伴うデバッグサーバ 3 0 とのやりとりにおいて、機密コードの漏洩のリスクを抑制することができる。また、返信内容秘匿部 1 2 2 は、デバッグサーバ 3 0 から受信された返信の内容に機密コードの情報漏洩につながる内容が含まれる場合、機密コードの情報漏洩につながる内容を削除することによって、この内容の存在自体を開発者に見せることも把握させることもないため、機密コードの漏洩のリスクをより抑制することができる。返信内容秘匿部 1 2 2 は、漏洩した場合にリスクが大きい NGClass 等の完全隠ぺいコードが含まれる場合に、完全隠ぺいコードを削除することによって、機密コードの漏洩のリスクを特に抑制することができる。

[0052] 例えば、返信内容秘匿部 1 2 2 は、送信部 1 1 1 によって開発ツール 2 0 に対して行われる返信の内容に、機密コードの情報漏洩につながる内容が含まれる場合、機密コードの情報漏洩につながる内容を秘匿にするか、または削除してよい。一例として、返信内容秘匿部 1 2 2 は、コマンドに対する返信の内容に、アクセス条件定義ファイルの許可リストに無い NGClass、RefNGClass および SecretFramework 等の関数のスタックフレームが含まれる場合、これらの関数のスタックフレームを削除する。

[0053] これにより、返信内容秘匿部 122 は、デバッグ処理に伴う開発ツール 20 とのやりとりにおいて、機密コードの漏洩のリスクを抑制することができる。また、返信内容秘匿部 122 は、デバッグサーバ 30 から受信された返信の内容に機密コードの情報漏洩につながる内容が含まれる場合、機密コードの情報漏洩につながる内容を削除することによって、この内容の存在自体を開発者に見せることも把握させることもないため、機密コードの漏洩のリスクをより抑制することができる。返信内容秘匿部 122 は、漏洩した場合にリスクが大きい NGClass 等の完全隠ぺいコードが含まれる場合に、完全隠ぺいコードを削除することによって、機密コードの漏洩のリスクを特に抑制することができる。

[0054] [記憶部]

記憶部 13 は、各種情報を記憶する。例えば、記憶部 13 は、機密コードが含まれるソースコードおよび DAP プロキシ 10 によって実行される OS (Operating System) や各種プログラム等を記憶する。記憶部 13 は、例えば、HDD (Hard Disk Drive)、SSD (Solid State Drive)、光ディスク等の記憶装置や RAM (Random Access Memory)、フラッシュメモリ、NVS RAM (Non Volatile Static Random Access Memory) 等の情報を書き換え可能な半導体メモリ等によって実現される。

[0055] [開発ツールの構成の一例]

開発ツール 20 は、プログラムのコードに対してデバッグ処理を行うためのデバッグ装置である。開発ツール 20 は、例えば、Vscode 等によって実現される。図 5 に示される例では、デバッグ装置は、プログラムの編集、デバッグ等の開発を行うための開発ツール 20 であるが、上述の機能を有するものであれば、特に限定されない。図 5 に示される例では、開発ツール 20 は、通信部 21 と、制御部 22 と、記憶部 23 と、操作受付部 24 と、表示部 25 とを有する。

[0056] [通信部]

通信部 21 は、DAP プロキシ 10 との間で各種情報を送受信する。通信

部 2 1 は、例えば、N I C 等によって実現される。図 5 に示される例では、通信部 2 1 は、送信部 2 1 1 と、受信部 2 1 2 とを有する。

[0057] [送信部]

送信部 2 1 1 は、D A P プロキシ 1 0 に各種情報を送信する。例えば、送信部 2 1 1 は、コードに対するデバッグ処理に関するコマンドを D A P プロキシ 1 0 に送信する。

[0058] [受信部]

受信部 2 1 2 は、D A P プロキシ 1 0 から各種情報を受信する。一例として、受信部 2 1 2 は、D A P プロキシ 1 0 から機密コードが含まれないソースコードを受信する。別の一例として、受信部 2 1 2 は、D A P プロキシ 1 0 から、許可リストに無い N G C l a s s、R e f N G C l a s s および S e c r e t F r a m e w o r k 等の関数のスタックフレームがマスキングまたは削除された、コマンドに対する返信を受信する。

[0059] [制御部]

制御部 2 2 は、開発ツール 2 0 全体を制御する。例えば、制御部 2 2 は、各種処理手順が規定されたプログラムや、制御情報が格納された内部メモリを有する一以上のプロセッサによって構成され、プロセッサは、プログラムや内部メモリを用いて各処理を実行する。制御部 2 2 は、例えば、C P U、M P U、G P U 等の電子回路や A S I C、F P G A 等の集積回路によって実現される。

[0060] [記憶部]

記憶部 2 3 は、各種情報を記憶する。例えば、記憶部 2 3 は、機密コードが含まれないソースコードおよび開発ツール 2 0 によって実行される O S や各種プログラム等を記憶する。記憶部 2 3 は、例えば、H D D、S S D、光ディスク等の記憶装置や R A M、フラッシュメモリ、N V S R A M 等の情報を書き換え可能な半導体メモリ等により実現される。

[0061] [操作受付部]

操作受付部 2 4 は、各種操作を受け付ける。例えば、操作受付部 2 4 は、

開発者の端末から、コードに対するデバッグ処理に関するコマンドの操作を受け付ける。操作受付部24は、例えば、インターネットを介して開発者の端末からデバッグ処理に関するデバッグ操作を受け付けるものや、GUI等によって実現される。

[0062] [表示部]

表示部25は、各種情報を表示する。例えば、表示部25は、許可リストに無いNGClass、RefNGClassおよびSecretFramework等の関数のスタックフレームがマスキングまたは削除された、コマンドに対する返信の内容を表示する。表示部25は、例えば、端末のデスクトップ等によって実現される。

[0063] [デバッグサーバの構成の一例]

デバッグサーバ30は、デバッグ装置とは独立したサーバである。図5に示される例では、デバッグサーバ30は、開発ツール20とは独立したデバッグ処理を行うためのサーバである。図5に示される例では、デバッグサーバ30は、通信部31と、制御部32と、記憶部33とを有する。

[0064] [通信部]

通信部31は、DAPプロキシ10との間で各種情報を送受信する。通信部31は、例えば、NIC等によって実現される。図5に示される例では、通信部31は、送信部311と、受信部312とを有する。

[0065] [送信部]

送信部311は、DAPプロキシ10に各種情報を送信する。例えば、送信部311は、コマンドに対する返信をDAPプロキシ10に送信する。

[0066] [受信部]

受信部312は、DAPプロキシ10から各種情報を受信する。一例として、受信部312は、DAPプロキシ10から機密コードが含まれるソースコードを受信する。別の一例として、受信部312は、DAPプロキシ10から許可リストに無いNGClass、RefNGClassおよびSecretFramework等の関数のスタックフレームが含まれるコマンドを受信する。

[0067] [制御部]

制御部 32 は、デバッグサーバ 30 全体を制御する。例えば、制御部 32 は、各種処理手順が規定されたプログラムや、制御情報が格納された内部メモリを有する一以上のプロセッサによって構成され、プロセッサは、プログラムや内部メモリを用いて各処理を実行する。制御部 32 は、例えば、CPU、MPU、GPU等の電子回路やASIC、FPGA等の集積回路によって実現される。

[0068] [記憶部]

記憶部 33 は、各種情報を記憶する。例えば、記憶部 33 は、機密コードが含まれるソースコードおよびデバッグサーバ 30 により実行されるOSや各種プログラム等を記憶する。記憶部 33 は、例えば、HDD、SSD、光ディスク等の記憶装置やRAM、フラッシュメモリ、NVS RAM等の情報を書き換え可能な半導体メモリ等により実現される。

[0069] [処理の流れの一例]

以下、図6を用いて実施形態に係る制御システム1のDAPプロキシ10によって実行される処理の流れの一例について説明する。図6は、制御システムによって実行される処理の流れの一例を示すフローチャートである。

[0070] ステップS11において、受信部112は、プログラムのコードに対してデバッグ処理を行うための開発ツール20から、コードに対するデバッグ処理に関するコマンドを受信する。例えば、受信部112は、開発ツール20の操作受付部24によってコマンドの操作が受け付けられた場合に、開発ツール20から、コマンドを受信する。

[0071] アクセスチェック部121は、受信部112によって開発ツール20から受信されたコマンドおよびコードへのアクセスに関する内容に基づいて、アクセスの条件が満たされない場合（ステップS12のYes）、ステップS13において、機密コードへのアクセスにつながる操作をスキップさせる。例えば、アクセスチェック部121は、アクセスチェック部121は、開発ツール20から送信されたコマンドにRefNGClass等の内部隠ぺいコードへのステップイン等の機密コードへのアクセスにつながる危険な操作が含まれる

場合、アクセスの条件が満たされていないと判断し、この操作をスキップさせる。

[0072] アクセスチェック部121は、受信部112によって開発ツール20から受信されたコマンドおよびコードへのアクセスに関する内容に基づいて、アクセスの条件が満たされる場合（ステップS12のNo）、ステップS14において、デバッグ装置とは独立したデバッグサーバ30の機密コードが含まれるコードにデバッグ処理の内容を反映させる。例えば、アクセスチェック部121は、コマンドおよびコードへのアクセスに関する内容がAllow listとして設定されているOKClassおよびOKClass2へのステップインである場合、アクセスの条件が満たされていると判断し、デバッグサーバ30に、デバッグサーバ30のソースコードにおけるOKClassおよびOKClass2へのステップインを実行させる。

[0073] [2. プログラム]

DAPプロキシ10は、パッケージソフトウェアやオンラインソフトウェアとして制御プログラムをコンピュータにインストールさせることにより実装できる。例えば、コンピュータが制御プログラムを実行することにより、DAPプロキシ10として機能させることができる。

[0074] 図7は、制御プログラムを実行するコンピュータの構成の一例を示す図である。図7に示される例では、コンピュータ1000は、メモリ1010とCPU1020とハードディスクドライバインタフェース1030とディスクドライバインタフェース1040とシリアルポートインタフェース1050とビデオアダプタ1060とネットワークインタフェース1070とを有する。これらの各部は、バス1080によって接続される。

[0075] メモリ1010は、ROM (Read Only Memory) 1011およびRAM 1012を含む。ROM1011は、例えば、BIOS (Basic Input Output System) 等のブートプログラムを記憶する。ハードディスクドライバインタフェース1030は、ハードディスクドライブ1090に接続される。ディスクドライバインタフェース1040は、ディスクドライブ1100に

接続される。例えば、磁気ディスクや光ディスク等の着脱可能な記憶媒体が、ディスクドライブ 1100 に挿入される。シリアルポートインタフェース 1050 は、例えば、マウス 1110、キーボード 1120 に接続される。ビデオアダプタ 1060 は、例えば、ディスプレイ 1130 に接続される。

[0076] ハードディスクドライブ 1090 は、例えば、OS 1091、アプリケーションプログラム 1092、プログラムモジュール 1093 およびプログラムデータ 1094 を記憶する。すなわち、DAP プロキシ 10 によって実行される各処理を規定するプログラムは、コンピュータにより実行可能なコードが記述されたプログラムモジュール 1093 として実装される。プログラムモジュール 1093 は、例えば、ハードディスクドライブ 1090 に記憶される。例えば、DAP プロキシ 10 における機能構成と同様の処理を実行するためのプログラムモジュール 1093 が、ハードディスクドライブ 1090 に記憶される。なお、ハードディスクドライブ 1090 は、SSD によって代替されてよい。

[0077] また、上述の実施形態の処理において用いられるデータは、プログラムデータ 1094 として、例えば、メモリ 1010 やハードディスクドライブ 1090 に記憶される。そして、CPU 1020 が、メモリ 1010 やハードディスクドライブ 1090 に記憶されたプログラムモジュール 1093 やプログラムデータ 1094 を、必要に応じて RAM 1012 に読み出して実行する。

[0078] なお、プログラムモジュール 1093 やプログラムデータ 1094 は、ハードディスクドライブ 1090 に記憶される場合に限定されず、例えば、着脱可能な記憶媒体に記憶され、ディスクドライブ 1100 等を介して CPU 1020 によって読み出されてもよい。あるいは、プログラムモジュール 1093 およびプログラムデータ 1094 は、ネットワーク（LAN (Local Area Network)、WAN (Wide Area Network) 等）を介して接続される他のコンピュータに記憶されてもよい。そして、プログラムモジュール 1093 およびプログラムデータ 1094 は、他のコンピュータから、ネットワ

ークインタフェース１０７０を介してＣＰＵ１０２０によって読み出されてもよい。

符号の説明

- [0079] １ 制御システム
- １０ ＤＡＰプロキシ（制御装置）
- １１、２１、３１ 通信部
- １２、２２、３２ 制御部
- １３、２３、３３ 記憶部
- ２０ 開発ツール（デバッグ装置）
- ２４ 操作受付部
- ２５ 表示部
- ３０ デバッグサーバ（サーバ）
- １１１、２１１、３１１ 送信部
- １１２、２１２、３１２ 受信部
- １２１ アクセスチェック部
- １２２ 返信内容秘匿部

請求の範囲

- [請求項1] プログラムのコードに対してデバッグ処理を行うためのデバッグ装置から、当該コードに対する前記デバッグ処理に関するコマンドを受信する受信部と、
- 前記受信部によって前記デバッグ装置から受信された前記コマンドおよび前記コードへのアクセスに関する内容に基づいて、前記アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせる制御部と、
- を有することを特徴とする制御装置。
- [請求項2] 前記制御部は、前記受信部によって前記デバッグ装置から受信された前記コマンドに、前記機密コードへのアクセスにつながる操作が含まれる場合、前記機密コードへのアクセスにつながる操作をスキップさせることを特徴とする請求項1に記載の制御装置。
- [請求項3] 前記制御部は、前記受信部によって前記デバッグ装置から受信された前記コマンドに、前記機密コードへのアクセスにつながる操作が含まれる場合、前記機密コードへのアクセスにつながる操作をスキップして、次のコードへのアクセスに関する内容に基づいて、前記デバッグ処理に関する制御を実行することを特徴とする請求項2に記載の制御装置。
- [請求項4] 前記制御部は、前記受信部によって前記デバッグ装置から受信された前記コマンドおよび前記コードへのアクセスに関する内容に基づいて、前記アクセスの条件が満たされる場合、前記デバッグ装置とは独立したサーバの前記機密コードが含まれるコードに前記デバッグ処理の内容を反映させることを特徴とする請求項1～3のいずれか1項に記載の制御装置。
- [請求項5] 前記受信部は、前記サーバから前記デバッグ処理に関する返信を受信し、
- 前記制御部は、前記受信部によって前記サーバから受信された前記

返信の内容に、前記機密コードへのアクセスにつながる操作が含まれる場合、前記機密コードへのアクセスにつながる操作をスキップし、前記機密コードの情報漏洩につながる内容が含まれる場合、前記機密コードの情報漏洩につながる内容を秘匿にするか、または削除することを特徴とする請求項4に記載の制御装置。

[請求項6] 前記デバッグ処理に関する返信を前記デバッグ装置に対して行う送信部をさらに有し、

前記制御部は、前記送信部によって前記デバッグ装置に対して行われる前記返信の内容に、前記機密コードの情報漏洩につながる内容が含まれる場合、前記機密コードの情報漏洩につながる内容を秘匿にするか、または削除することを特徴とする請求項1に記載の制御装置。

[請求項7] 制御装置によって実行される制御方法であって、

プログラムのコードに対してデバッグ処理を行うためのデバッグ装置から、当該コードに対する前記デバッグ処理に関するコマンドを受信する受信工程と、

前記受信工程によって前記デバッグ装置から受信された前記コマンドおよび前記コードへのアクセスに関する内容に基づいて、前記アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせる制御工程と、

を含むことを特徴とする制御方法。

[請求項8] プログラムのコードに対してデバッグ処理を行うためのデバッグ装置から、当該コードに対する前記デバッグ処理に関するコマンドを受信する受信ステップと、

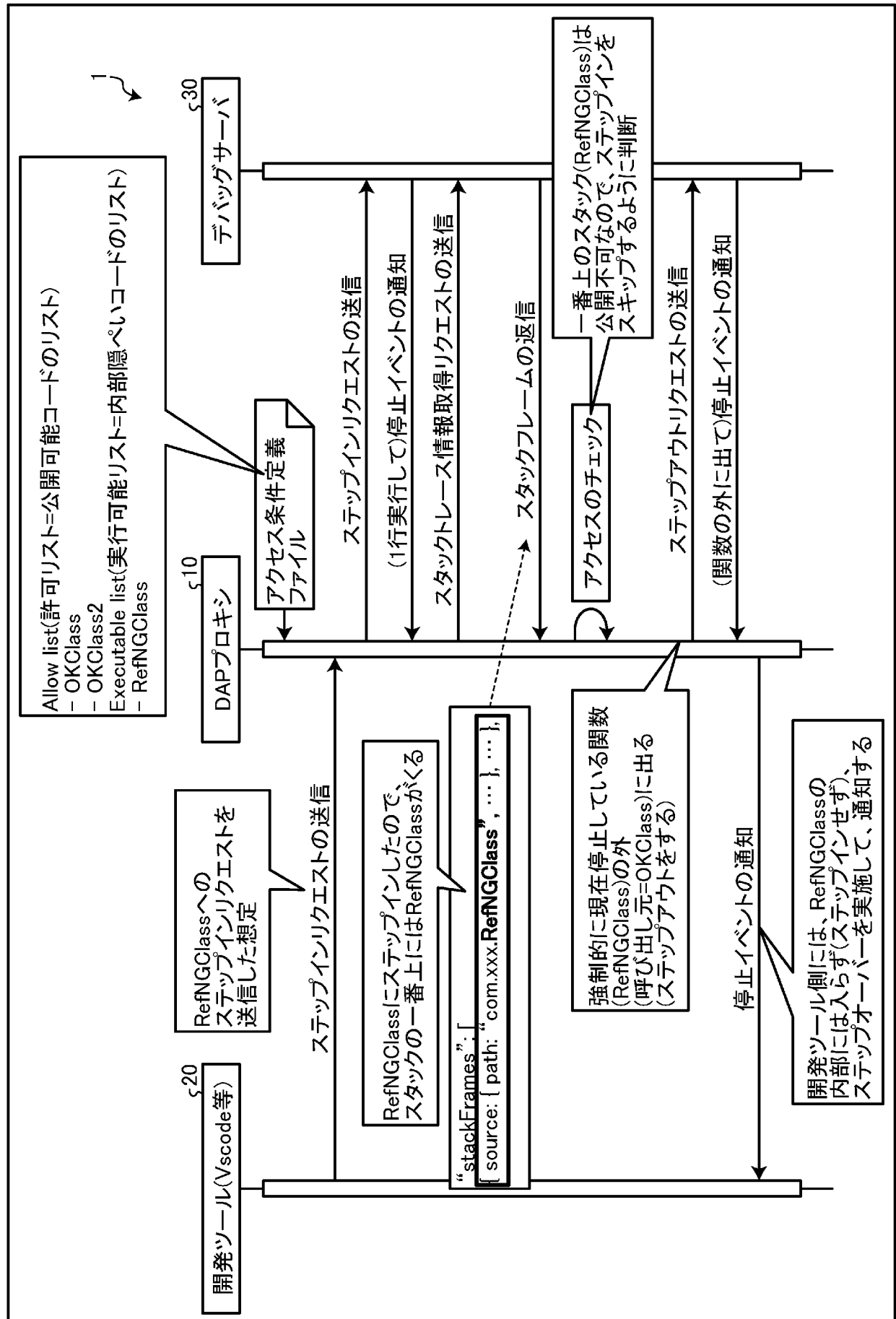
前記受信ステップによって前記デバッグ装置から受信された前記コマンドおよび前記コードへのアクセスに関する内容に基づいて、前記アクセスの条件が満たされない場合、機密コードへのアクセスにつながる操作をスキップさせる制御ステップと、

をコンピュータに実行させことを特徴とする制御プログラム。

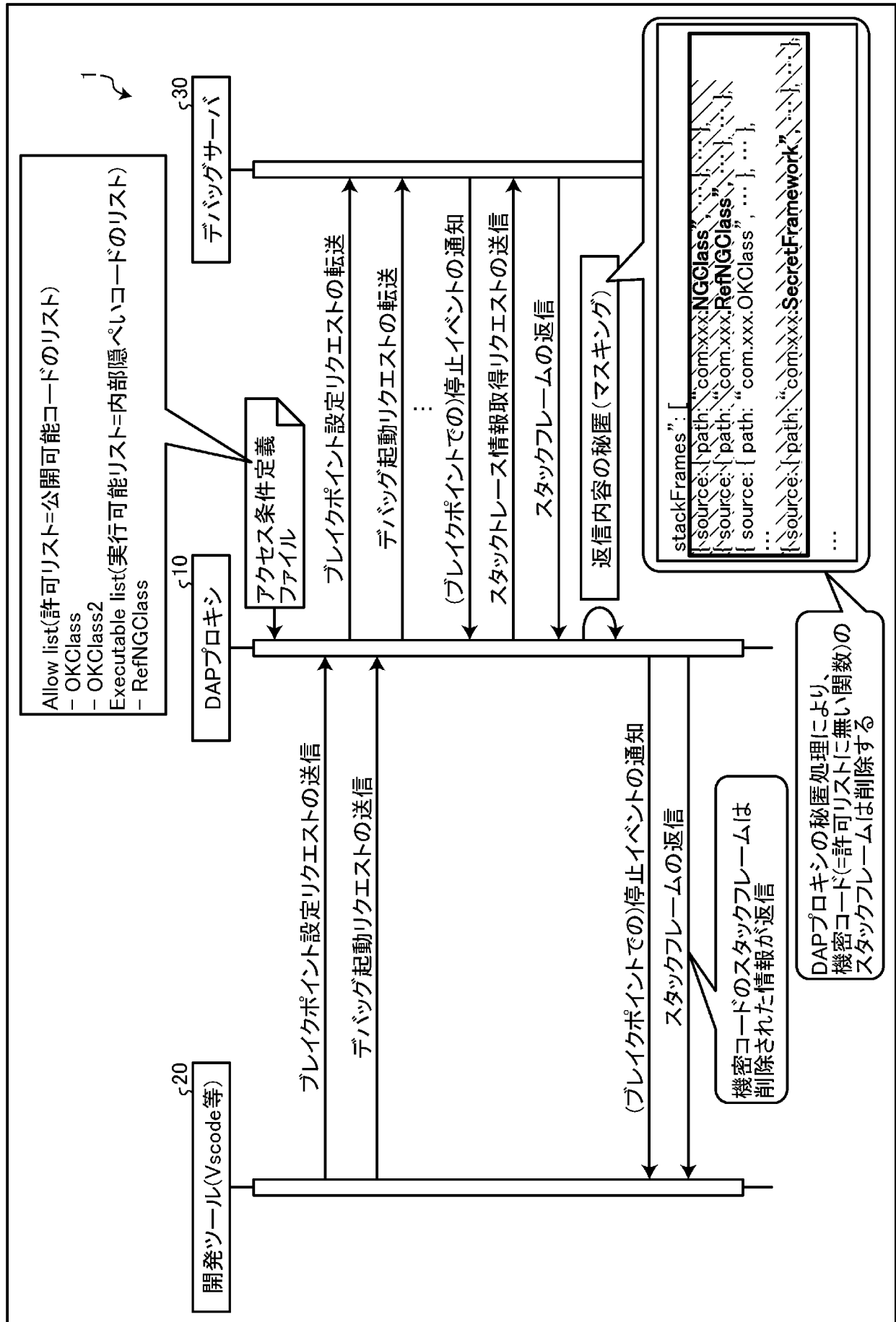
[図2]

機密コードの分類	実装内容の公開	関数のインターフェイス (I/F)の内容公開	コードの説明
内部隠ぺいコード	× (関数の中身が非公開)	○ (関数の呼び出しは許可)	・開発者がプログラミングをする上で、 呼び出せるコード ・開発者に対して、 関数の存在そのものは秘匿にしない
完全隠ぺいコード	× (関数の中身が非公開)	× (関数の呼び出しが不許可)	・開発者がプログラミングをする上で、 呼び出せないコード ・開発者に対して、 関数の存在そのものも秘匿にする

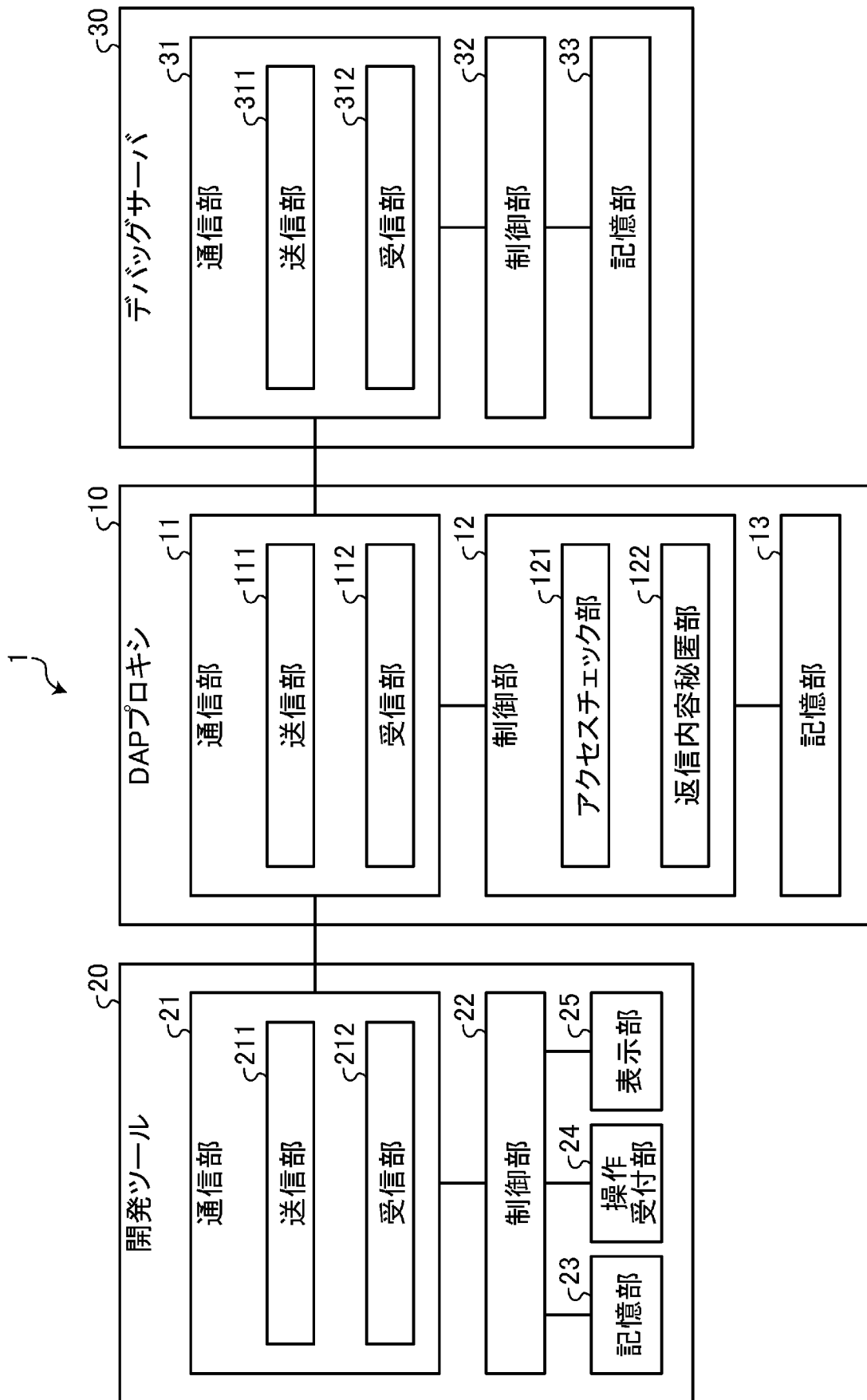
[図3]



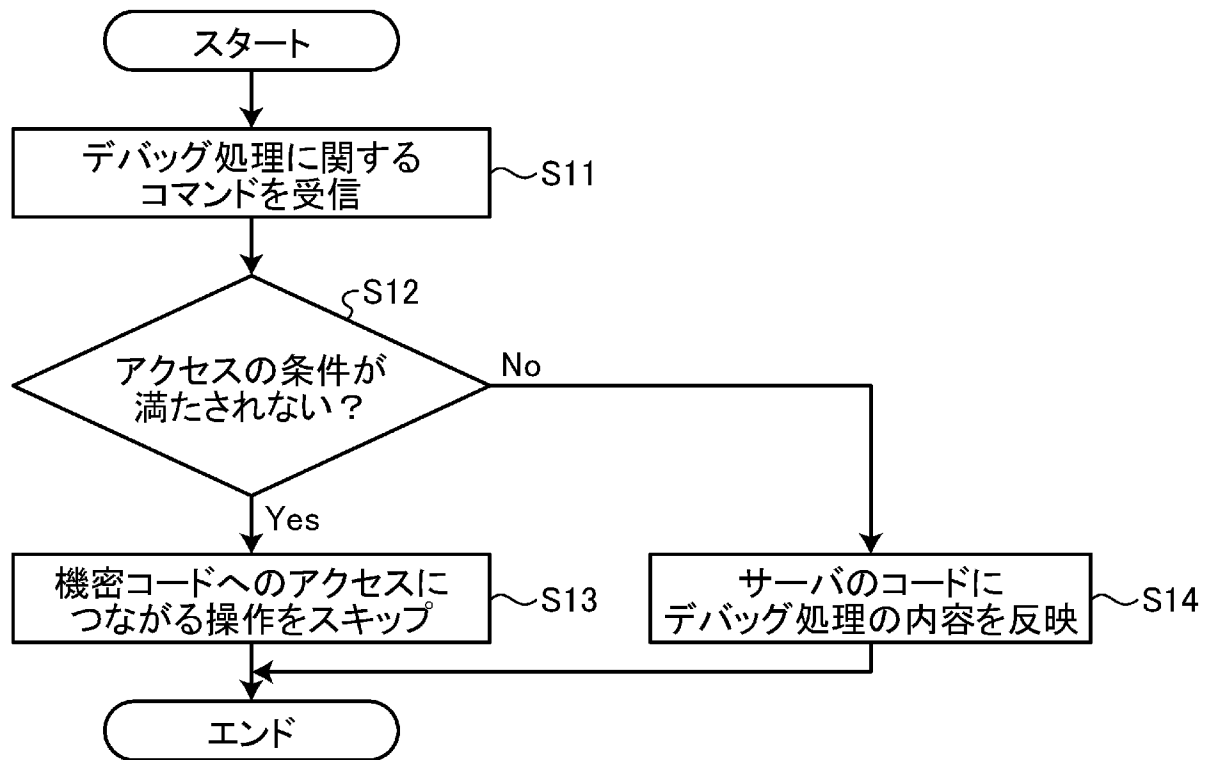
[図4]



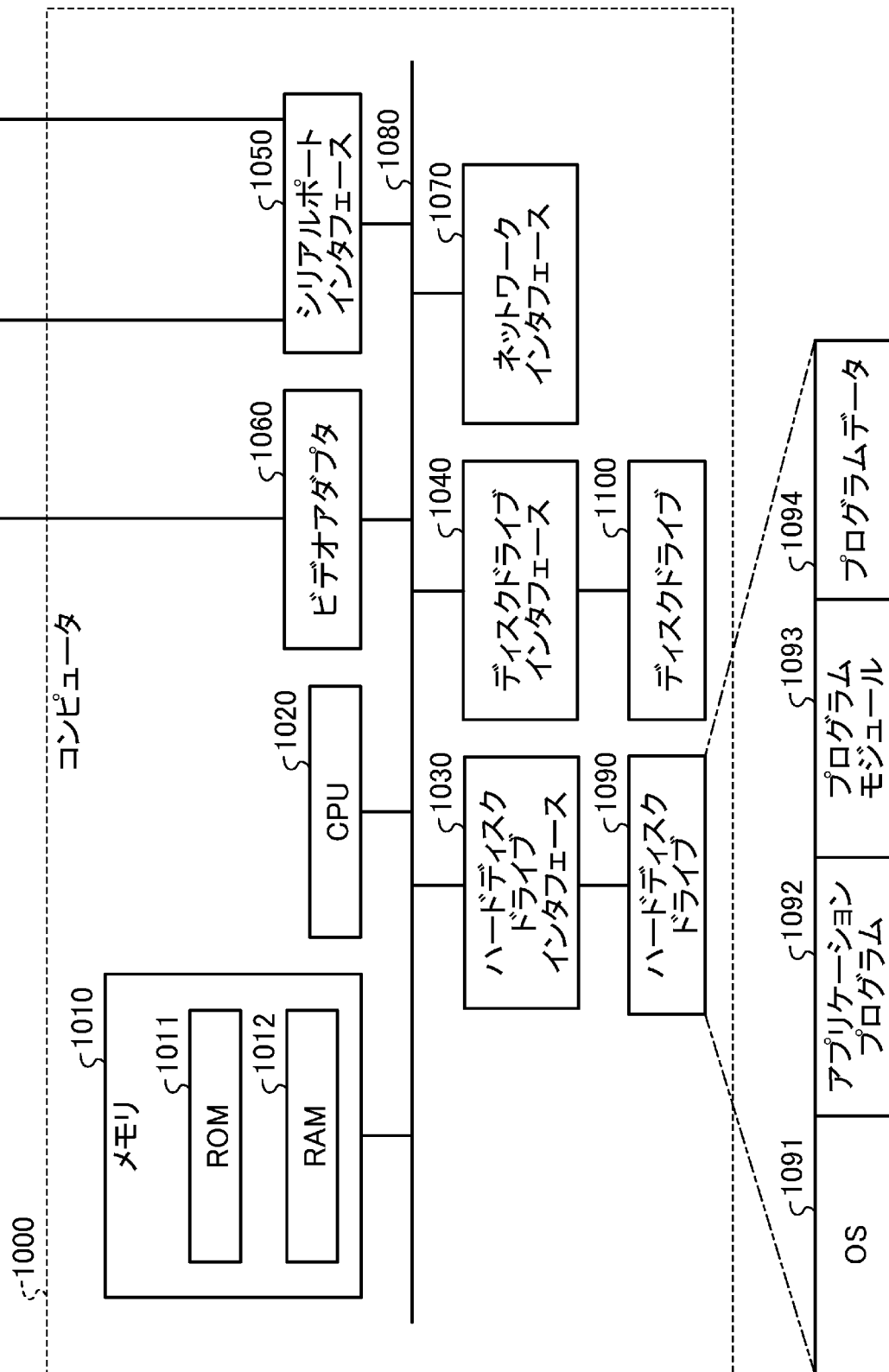
[図5]



[図6]

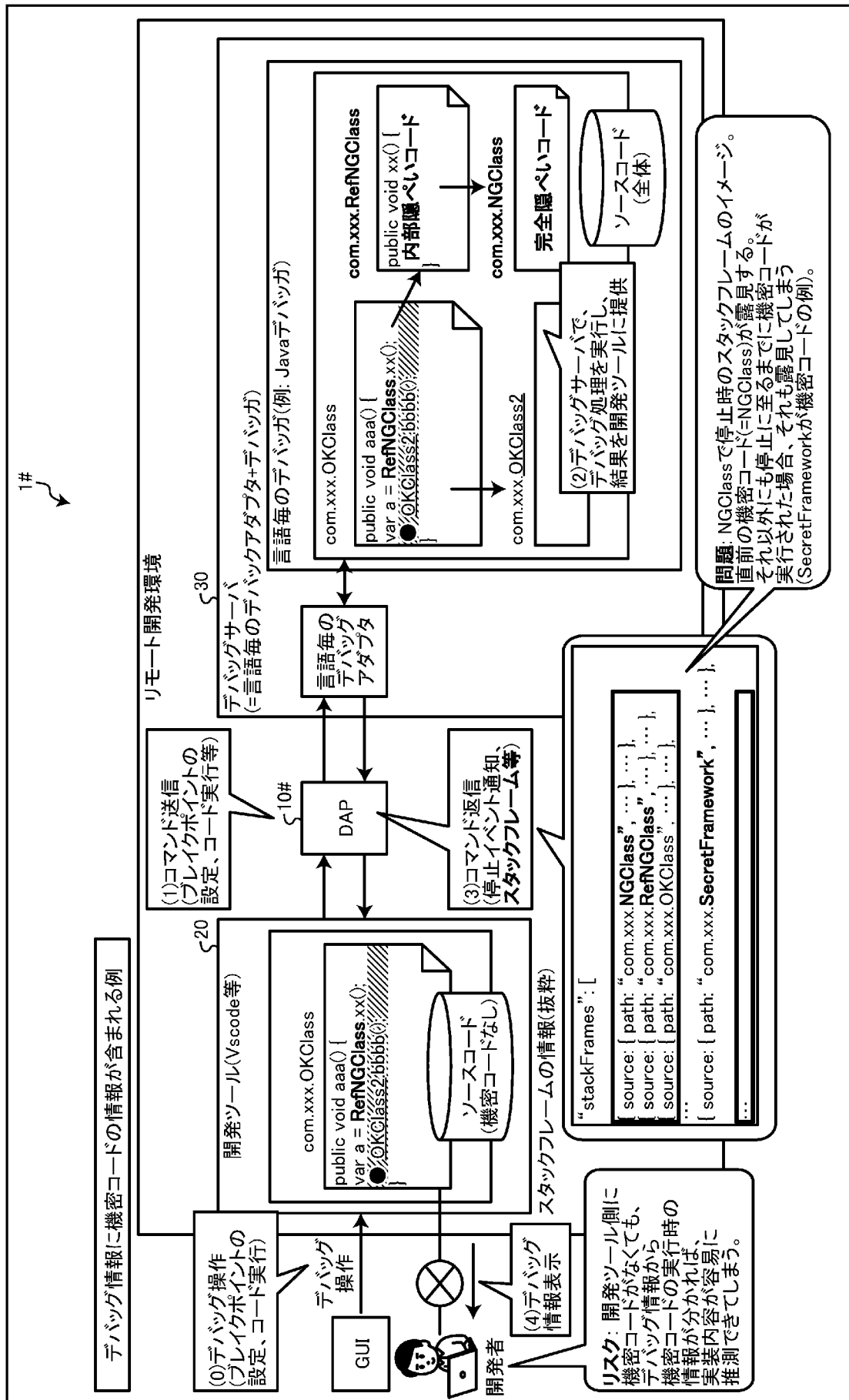


[図7]



[illegible]

[図10]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2023/025785

A. CLASSIFICATION OF SUBJECT MATTER**G06F 21/60**(2013.01)i; **G06F 21/57**(2013.01)i

FI: G06F21/60; G06F21/57

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)

G06F21/57; G06F21/60

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

Published examined utility model applications of Japan 1922-1996
 Published unexamined utility model applications of Japan 1971-2023
 Registered utility model specifications of Japan 1996-2023
 Published registered utility model applications of Japan 1994-2023

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 2009-129326 A (CANON KABUSHIKI KAISHA) 11 June 2009 (2009-06-11) paragraphs [0093], [0098], fig. 20	1-8
A	WO 2022/259311 A1 (NIPPON TELEGRAPH AND TELEPHONE CORPORATION) 15 December 2022 (2022-12-15) abstract	1-8



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

04 September 2023

Date of mailing of the international search report

19 September 2023

Name and mailing address of the ISA/JP

Japan Patent Office (ISA/JP)
 3-4-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8915
 Japan

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/JP2023/025785

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
JP	2009-129326	A	11 June 2009	(Family: none)	
WO	2022/259311	A1	15 December 2022	(Family: none)	

A. 発明の属する分野の分類（国際特許分類（IPC）） G06F 21/60(2013.01)i; G06F 21/57(2013.01)i FI: G06F21/60; G06F21/57		
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）） G06F21/57; G06F21/60 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 日本国実用新案公報 1922 - 1996年 日本国公開実用新案公報 1971 - 2023年 日本国実用新案登録公報 1996 - 2023年 日本国登録実用新案公報 1994 - 2023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 2009-129326 A（キヤノン株式会社）11.06.2009（2009 - 06 - 11） 段落[0093], [0098], 図20	1-8
A	WO 2022/259311 A1（日本電信電話株式会社）15.12.2022（2022 - 12 - 15） [要約]	1-8
☐ 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、一般的な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抵触するものではなく、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 04.09.2023	国際調査報告の発送日 19.09.2023	
名称及びあて先 日本国特許庁(ISA/JP) 〒100-8915 日本国 東京都千代田区霞が関三丁目4番3号	権限のある職員（特許庁審査官） 平井 誠 5S 9071 電話番号 03-3581-1101 内線 3546	

国際調査報告
パテントファミリーに関する情報

国際出願番号

PCT/JP2023/025785

引用文献			公表日	パテントファミリー文献	公表日
JP	2009-129326	A	11.06.2009	(ファミリーなし)	
WO	2022/259311	A1	15.12.2022	(ファミリーなし)	