(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
2020年12月30日(30.12.2020)

WIPO | PCT

(10) 国際公開番号
WO 2020/262456 A1

(51) 国際特許分類:
G06F 8/00 (2018.01) *G06F 21/57* (2013.01)
G06F 8/41 (2018.01)

(21) 国際出願番号： PCT/JP2020/024799

(22) 国際出願日： 2020年6月24日(24.06.2020)

(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 2019-119106 2019年6月26日(26.06.2019) JP

(71) 出願人: コネクトフリー株式会社 (**CONNECTFREE CORPORATION**) [JP/JP]; 〒6008009 京都府京都市下京区四条烏丸西入ル函谷鉾町８３番地 Kyoto (JP).

(72) 発明者: 帝都 久利寿 (**TATE, Kristopher Andrew**); 〒6008009 京都府京都市下京区四条烏丸西入ル函谷鉾町８３番地 コネクトフリー株式会社内 Kyoto (JP).

(74) 代理人: 特許業務法人深見特許事務所(**FUKAMI PATENT OFFICE, P.C.**); 〒5300005 大阪府大阪市北区中之島三丁目２番４号 中之島フェスティバルタワー・ウエスト Osaka (JP).

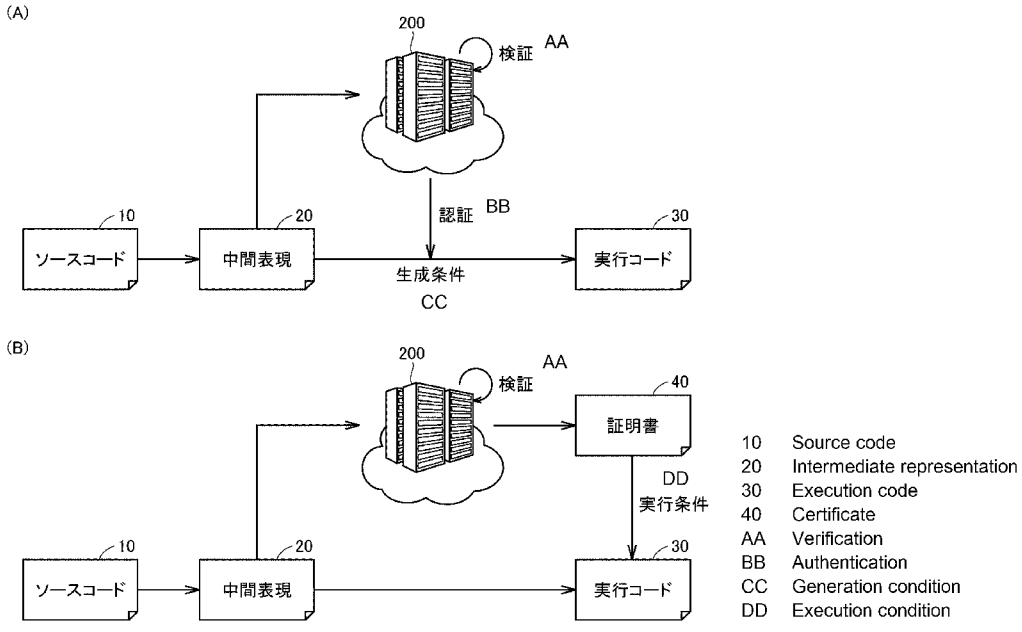
(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ,

WO 2020/262456 A1

(54) **Title:** EXECUTION CODE PROVISION METHOD AND SOFTWARE DEVELOPMENT SYSTEM

(54) 発明の名称： 実行コード提供方法およびソフトウェア開発システム

FIG.1



(57) **Abstract:** The present invention provides a solution with which it is possible to assure safety of an application program. This execution code provision method comprises: a step for preparing a source code in accordance with a user operation; a step for generating an intermediate representation from the source code; a step for verifying whether or not the intermediate representation satisfies a prescribed rule or protocol; and a step for implementing the execution of an execution code generated from the intermediate representation in the case when the intermediate representation meets the prescribed rule or protocol.

［続葉有］

NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, 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, GH, GM, KE, LR, LS, MW, MZ, NA, RW, 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, 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:

一 国際調査報告(条約第21条(3))

(57) 要約: アプリケーションプログラムの安全性を担保できるソリューションが提供される。実行コード提供方法は、ユーザ操作に従ってソースコードを作成するステップと、ソースコードから中間表現を生成するステップと、中間表現が所定の規定あるいは規約を満たしているか否かを検証するステップと、中間表現が所定の規定あるいは規約を満たしている場合に、中間表現から生成される実行コードの実行を実現するステップとを含む。

明　細　書

発明の名称 ：

実行コード提供方法およびソフトウェア開発システム

技術分野

[0001] 本開示は、実行コード提供方法およびソフトウェア開発システムに関する。

背景技術

[0002] 近年の情報通信技術（Information and Communication Technology：ＩＣＴ）の進歩は目覚ましく、インターネットなどのネットワークに接続されるデバイスは、従来のパーソナルコンピュータやスマートフォンといった情報処理装置に限らず、様々なモノ（things）に広がっている。このような技術トレンドは、「ＩｏＴ（Internet of Things；モノのインターネット）」と称され、様々な技術およびサービスが提案および実用化されつつある。将来的には、地球上の数十億人と数百億または数兆のデバイスとが同時につながる世界が想定されている。このようなネットワーク化された世界を実現するためには、よりシンプル、より安全、より自由につながることができるソリューションを提供する必要がある。

[0003] ＩｏＴで利用されるデバイス（「エッジデバイス」とも称される。）のインテリジェント化に伴って、エッジデバイスでは様々な種類のアプリケーションプログラムが実行可能になる。一方で、エッジデバイスに対する安全性は、リソース制約などによって、サーバやパーソナルコンピュータなどに対する安全性と同程度まで高めることは容易ではない。そのため、エッジデバイスに組み入れられるアプリケーションプログラムに対する検証などが重要になってくる。

[0004] 例えば、特開２０１４－１７４５７７号公報（特許文献１）は、Ｗｅｂアプリケーションの脆弱性をソースコードレベルで検証する静的解析ツールの改良を開示する。

先行技術文献

特許文献

[0005] 特許文献1：特開２０１４－１７４５７７号公報

発明の概要

発明が解決しようとする課題

[0006] エッジデバイスなどで利用されるアプリケーションプログラムは、ソースコードの形態ではなく、実行コード生成前の中間表現（intermediate representation）として流通することも多い。このような中間表現を採用することで、ソースコードに含まれるノウハウなどを秘匿化するとともに、データ量を低減できるという利点がある。

[0007] 上述の特許文献１に開示されるように、ソースコードを解析してセキュリティホールや脆弱性などを評価することは一般的であるが、ソースコード以外の形式でセキュリティホールや脆弱性などを評価することは容易ではない。

[0008] 本開示の典型的な目的は、エッジデバイスなどに好適な、アプリケーションプログラムの安全性を担保できるソリューションを提供することである。

課題を解決するための手段

[0009] 本開示のある形態に従う実行コード提供方法は、ユーザ操作に従ってソースコードを作成するステップと、ソースコードから中間表現を生成するステップと、中間表現が所定の規定あるいは規約を満たしているか否かを検証するステップと、中間表現が所定の規定あるいは規約を満たしている場合に、中間表現から生成される実行コードの実行を実現するステップとを含む。

[0010] 実行コードの実行を実現するステップは、中間表現が所定の規定あるいは規約を満たしていることを条件として、中間表現から実行コードを生成するステップを含んでいてもよい。

[0011] 実行コードの実行を実現するステップは、中間表現が所定の規定あるいは規約を満たしていることを条件として、中間表現から生成された実行コード

の実行を許可するステップを含んでいてもよい。

[0012] 実行コード提供方法は、実行コードが実行されるコントローラの起動時に、当該実行コードを生成した中間表現に関連付けて発行された証明書に基づいて、当該実行コードの正当性を評価するステップをさらに含んでいてもよい。

[0013] 検証するステップは、中間表現が所定の規定あるいは規約を満たしている場合に、中間表現に関連付けられる証明書を発行するステップを含み、中間表現から生成された実行コードの実行を許可するステップは、証明書に基づいて実行の可否を判断するステップを含んでいてもよい。

[0014] 検証するステップは、中間表現が所定の規定あるいは規約を満たしていない場合に、所定の規定あるいは規約を満たしていない内容をユーザに通知するステップを含んでいてもよい。

[0015] 本開示の別の形態に従うソフトウェア開発システムは、ソースコードの作成を支援するソフトウェア開発装置と、ソフトウェア開発装置からアクセス可能な検証サーバとを含む。ソフトウェア開発装置は、ソースコードから中間表現を生成する手段を含む。検証サーバは、中間表現が所定の規定あるいは規約を満たしているか否かを検証する手段を含む。中間表現が所定の規定あるいは規約を満たしている場合に、中間表現から生成される実行コードの実行が実現される。

発明の効果

[0016] 本開示によれば、アプリケーションプログラムの安全性を担保できるソリューションを提供できる。

図面の簡単な説明

[0017] [図1]本実施の形態に従うソフトウェア開発システムにおける処理の概要を説明するための図である。

[図2]本実施の形態に従うソフトウェア開発システムにおけるソースコードから実行コードを生成する手順の概要を示す図である。

[図3]本実施の形態に従うソフトウェア開発システムにおけるソースコードか

ら実行コードを生成する手順の概要を示す図である。

[図4]本実施の形態に従うソフトウェア開発装置のハードウェア構成例を示す模式図である。

[図5]本実施の形態に従う検証サーバのハードウェア構成例を示す模式図である。

[図6]本実施の形態に従うソフトウェア開発システムにおける処理手順の一例を示すフローチャートである。

発明を実施するための形態

[0018] 本開示に係る実施の形態について、図面を参照しながら詳細に説明する。なお、図中の同一または相当部分については、同一符号を付してその説明は繰り返さない。

[0019] ＜Ａ．概要＞

まず、本実施の形態に従うソフトウェア開発システム１における処理の概要を説明する。本実施の形態においては、ソースコード１０から生成される中間表現２０自体を検証し、その検証結果が適切であることを条件に、対応する実行コード３０の生成、あるいは、生成された実行コード３０の実行を許容する。中間表現２０に対する検証において、中間表現２０が所定の規定あるいは規約を満たす場合に適切と判断される。

[0020] 本明細書において、「中間表現」は、コンパイラ基盤において、任意のプログラミング言語（通常は、高級言語）で記載されたソースコードを特定の仮想マシンをターゲットとして生成される、当該仮想マシン用のマシン語を包含する。但し、「中間表現」は、必ずしもバイナリ形式である必要はなく、アセンブラ言語のような自然言語に類似した形式で表現されてもよい。さらに、「中間表現」は、特定の仮想マシンをターゲットとして生成されるものであれば、どのような表現形式を採用してもよい。

[0021] 通常、「中間表現」は、実行コード３０が実行されるプラットフォームおよびアーキテクチャとは独立して生成される。このようなソースコード１０から中間表現２０の生成には、静的コンパイラが用いられてもよい。さらに

、「中間表現」は、特定のプラットフォームおよびアーキテクチャに向けられた実行コード３０としてコンパイルされてもよい。「中間表現」および「実行コード」は、セットとして生成される。「中間表現」は、元になったソースコードに比較してデータ量が削減されるとともに、記述される処理自体も効率化できる。

[0022] ソースコード１０から中間表現２０の生成は、基本的には一方向の変換であり、中間表現２０からソースコード１０を完全に復元することは難しい。中間表現２０と実行コード３０との間は、双方向に変換することも可能である。

[0023] 図１は、本実施の形態に従うソフトウェア開発システム１における処理の概要を説明するための図である。図１（Ａ）には、ソースコード１０から生成される中間表現２０に対する検証が成功したことを、実行コード３０の生成条件とする例を示す。図１（Ｂ）には、ソースコード１０から生成される中間表現２０に対する検証が成功したことを、中間表現２０から生成された実行コード３０の実行条件とする例を示す。

[0024] 図１（Ａ）および（Ｂ）に示すように、ソースコード１０から生成された中間表現２０が任意の主体により検証される。図１（Ａ）および（Ｂ）に示す例では、検証サーバ２００が中間表現２０を検証する例を示すが、これに限られず、任意の主体によって検証されてもよい。

[0025] 図１（Ａ）を参照して、検証サーバ２００による検証結果が適切なものである場合には、検証サーバ２００は、中間表現２０に対して認証を与える。中間表現２０に対する認証が生成条件として、中間表現２０から実行コード３０が生成される。このように、中間表現２０に対する検証を実行コード３０の生成条件に含めるようにしてもよい。このような生成条件を採用することで、生成される実行コード３０についての安全性を高めることができる。

[0026] 図１（Ｂ）を参照して、検証サーバ２００による検証結果が適切なものである場合には、検証サーバ２００は、中間表現２０に対する認証を含む証明書４０を発行する。並行して、中間表現２０から実行コード３０が生成され

る。生成された実行コード３０の実行には、対応する証明書４０の存在を条件としてもよい。このように、中間表現２０に対する証明書４０の存在を実行コード３０の実行条件に含めるようにしてもよい。このような実行条件を採用することで、安全性が損なわれた状態で実行コード３０が実行される可能性を低減できる。

[0027] 以上のように、本実施の形態における一連のプロセスにおいては、中間表現２０が所定の規定あるいは規約を満たす場合（検証結果が適切である場合）に、その中間表現２０に対応する実行コード３０の生成または実行を許可するとの認証が与えられる。逆に、中間表現２０が所定の規定あるいは規約を満たさない場合（検証結果が不適切である場合）には、認証が与えられず、その中間表現２０に対応する実行コード３０の生成または実行ができない。すなわち、中間表現２０が所定の規定あるいは規約を満たしている場合に、中間表現２０から生成される実行コード３０の実行が実現される。

[0028] 実行コード３０の実行を実現する処理としては、図１（Ａ）に示すように、中間表現２０が所定の規定あるいは規約を満たしていることを条件として、中間表現２０から実行コード３０を生成する処理を採用してもよい。あるいは、実行コード３０の実行を実現する処理としては、図１（Ｂ）に示すように、中間表現２０が所定の規定あるいは規約を満たしていることを条件として、中間表現２０から生成された実行コード３０の実行を許可する処理を採用してもよい。このとき、中間表現２０から生成された実行コード３０の実行を許可するか否かは、中間表現２０に関連付けられた証明書４０に基づいて判断してもよい。なお、実行の可否は、実行コード３０の実行環境（プロセッサあるいはプロセッサを含むマイクロコンピュータなど）で判断してもよい。

[0029] このように、本実施の形態においては、ソースコード１０ではなく中間表現２０に対する検証を行うことで、ソースコード１０に含まれるノウハウなどを秘匿化した状態で、アプリケーションプログラムに含まれる安全性についてのリスクをより効率的に発見できる。また、中間表現２０に対する検証

を行うことで、ソースコード１０の記述に用いられたプログラミング言語の種類に応じた複数の検証ルールセットなどを用意する必要がなく、効率的な検証を実現できる。

[0030] 次に、本実施の形態に従うソフトウェア開発システム１を用いた実行コード３０の生成手順の一例について説明する。

[0031] 図２および図３は、本実施の形態に従うソフトウェア開発システム１におけるソースコード１０から実行コード３０を生成する手順の概要を示す図である。

[0032] 図２を参照して、ソフトウェア開発システム１は、１または複数のソフトウェア開発装置１００と、ソフトウェア開発装置１００からインターネット２などを介してアクセス可能な検証サーバ２００とを含む。

[0033] ソフトウェア開発装置１００は、エッジデバイス４に含まれるコントローラ３００で実行されるアプリケーションプログラムの作成を支援する。但し、本発明の技術的範囲は、エッジデバイス４で実行されるアプリケーションプログラムの作成に限定されるものではなく、任意のコンピューティングデバイスで実行されるアプリケーションプログラムの作成に適用可能である。

[0034] ソフトウェア開発装置１００には、統合開発環境（ＩＤＥ：Integrated Development Environment）が提供されており、ユーザは統合開発環境上で任意のアプリケーションプログラムを作成できる。

[0035] エッジデバイス４に含まれるコントローラ３００は、コンピューティングデバイスの一例であり、プロセッサを含む。エッジデバイス４としては、どのようなデバイスであってもよいが、典型的には、工場設備、家庭内の各種装置、社会インフラ設備、車両などの移動体、任意の携帯デバイスなどが想定される。後述するように、コントローラ３００は、プロセッサを有しており、ソフトウェア開発装置１００からのアプリケーションプログラムを実行可能になっている。

[0036] まず、ユーザは、ソフトウェア開発装置１００を用いてソースコードを作成する（（１）ソースコード作成）。そして、作成されたソースコードは、

ソフトウェア開発装置１００においてコンパイルされて中間表現２０が生成される（（２）中間表現生成）。

[0037] 中間表現２０から実行コード３０が生成される前に中間表現２０が検証される。図２に示す例においては、生成された中間表現２０が検証サーバ２００へ送信される（（３）中間表現検証依頼）。そして、検証サーバ２００は、ソフトウェア開発装置１００からの中間表現２０を検証する（（４）中間表現の検証）。検証サーバ２００はソフトウェア開発装置１００へ検証結果を応答する。対象の中間表現２０がすべての検証にパスすれば、検証サーバ２００は、検証結果が適切であったことを示す認証をソフトウェア開発装置１００へ送信することになる。

[0038] ソフトウェア開発装置１００は、検証結果が適切であったことを示す認証を検証サーバ２００から受信すると（（５）検証結果（認証）受信）、中間表現２０から実行コード３０を生成する（（６）実行コード生成）。

[0039] そして、ソフトウェア開発装置１００は、生成した実行コード３０をエッジデバイス４のコントローラ３００へ転送する（（７）実行コード転送）。そして、エッジデバイス４のコントローラ３００は、必要に応じて、転送された実行コード３０を実行する（（８）実行コードの実行）。

[0040] 図２および図３に示すような一連の処理によって、エッジデバイス４のコントローラ３００において、セキュアなアプリケーションプログラムを実行させることができる。

[0041] ＜Ｂ．ハードウェア構成例＞

次に、本実施の形態に従うソフトウェア開発システム１を構成する主要装置のハードウェア構成例について説明する。

[0042] （ｂ１：ソフトウェア開発装置１００）

ソフトウェア開発装置１００は、典型的には汎用コンピュータで実現される。

[0043] 図４は、本実施の形態に従うソフトウェア開発装置１００のハードウェア構成例を示す模式図である。図４を参照して、ソフトウェア開発装置１００

は、主たるコンポーネントとして、プロセッサ１０２と、メインメモリ１０４と、入力部１０６と、ディスプレイ１０８と、ハードディスク１１０と、通信インターフェイス１２２とを含む。これらのコンポーネントは、内部バス１２０を介して接続されている。

[0044] プロセッサ１０２は、例えば、ＣＰＵ（Central Processing Unit）やＧＰＵ（Graphics Processing Unit）などで構成される。複数のプロセッサ１０２が配置されてもよいし、複数のコアを有するプロセッサ１０２を採用してもよい。

[0045] メインメモリ１０４は、ＤＲＡＭ（Dynamic Random Access Memory）やＳＲＡＭ（Static Random Access Memory）などの揮発性記憶装置で構成される。ハードディスク１１０は、プロセッサ１０２で実行される各種プログラムや各種データを保持する。なお、ハードディスク１１０に代えて、ＳＳＤ（Solid State Drive）やフラッシュメモリなどの不揮発性記憶装置を採用してもよい。ハードディスク１１０に格納されたプログラムのうち、指定されたプログラムコードがメインメモリ１０４上に展開され、プロセッサ１０２は、メインメモリ１０４上に展開されたプログラムコードに含まれるコンピュータ可読命令（computer-readable instructions）を順次実行することで、後述するような各種機能を実現する。

[0046] 典型的には、ハードディスク１１０には、ユーザが任意に作成するソースコード１０と、統合開発環境を実現するためのソフトウェア開発プログラム１１４と、ソースコード１０から生成される実行コード３０とが格納される。ソフトウェア開発プログラム１１４は、ユーザが任意に作成するソースコード１０から中間表現２０を介して実行コード３０を生成するものであり、アプリケーションプログラムの開発環境を提供するモジュールを含む。但し、中間表現２０を検証した結果が適切であることを条件に、対応する実行コード３０の生成あるいは実行が許容される。

[0047] 入力部１０６は、ソフトウェア開発装置１００を操作するユーザの入力操作を受け付ける。入力部１０６は、例えば、キーボード、マウス、表示デバ

イス上に配置されたタッチパネル、ソフトウェア開発装置１００の筐体に配置された操作ボタンなどであってもよい。

[0048] ディスプレイ１０８は、プロセッサ１０２での処理結果などを表示する。ディスプレイ１０８は、例えば、ＬＣＤ（Liquid Crystal Display）や有機ＥＬ（Electro-Luminescence)ディスプレイなどであってもよい。

[0049] 通信インターフェイス１２２は、検証サーバ２００とのデータ交換を担当する。通信インターフェイス１２２は、例えば、インターネットを介した通信ができるように、イーサネット（登録商標）ポートを含んでいてもよい。

[0050] なお、ソフトウェア開発装置１００の全部または一部は、コンピュータ可読命令に相当する回路が組み込まれたＡＳＩＣ（Application Specific Integrated Circuit）などのハードワイヤード回路を用いて実現してもよい。さらにあるいは、ＦＰＧＡ（field-programmable gate array）上にコンピュータ可読命令に相当する回路を用いて実現してもよい。また、プロセッサ１０２およびメインメモリ、ＡＳＩＣ、ＦＰＧＡなどを適宜組み合わせて実現してもよい。

[0051] ソフトウェア開発装置１００は、コンピュータ可読命令を含むソフトウェア開発プログラム１１４を格納する非一過性（non-transitory）のメディアから、当該格納しているプログラムなどを読み出すためのコンポーネントをさらに有していてもよい。メディアは、例えば、ＤＶＤ（Digital Versatile Disc）などの光学メディア、ＵＳＢメモリなどの半導体メディアなどであってもよい。

[0052] なお、ソフトウェア開発プログラム１１４は、メディアを介してソフトウェア開発装置１００にインストールされるだけではなく、ネットワーク上の配信サーバから提供されるようにしてもよい。

[0053] （ｂ２：検証サーバ２００）

検証サーバ２００についても、典型的には汎用コンピュータで実現される。

[0054] 図５は、本実施の形態に従う検証サーバ２００のハードウェア構成例を示

す模式図である。図５を参照して、検証サーバ２００は、主たるコンポーネントとして、１または複数のプロセッサ２０２と、メインメモリ２０４と、入力部２０６と、ディスプレイ２０８と、ハードディスク２１０と、通信インターフェイス２２２とを含む。これらのコンポーネントは、内部バス２２０を介して接続されている。

[0055] プロセッサ２０２は、例えば、ＣＰＵやＧＰＵなどで構成される。複数のプロセッサ２０２が配置されてもよいし、複数のコアを有するプロセッサ２０２を採用してもよい。

[0056] メインメモリ２０４は、ＤＲＡＭやＳＲＡＭなどの揮発性記憶装置で構成される。ハードディスク２１０は、プロセッサ２０２で実行される各種プログラムや各種データを保持する。なお、ハードディスク２１０に代えて、ＳＳＤやフラッシュメモリなどの不揮発性記憶装置を採用してもよい。ハードディスク２１０に格納されたプログラムのうち、指定されたプログラムコードがメインメモリ２０４上に展開され、プロセッサ２０２は、メインメモリ２０４上に展開されたプログラムコードに含まれるコンピュータ可読命令を順次実行することで、後述するような各種機能を実現する。

[0057] 典型的には、ハードディスク２１０には、外部からの依頼を受けて評価処理を実行するための評価エンジン２１２と、評価エンジン２１２による評価結果に応じて証明書を発行するための証明書発行エンジン２１４と、評価エンジン２１２による評価処理に用いられる評価ルール２１６とが格納される。評価エンジン２１２は、中間表現２０が所定の規定あるいは規約を満たしているか否かを検証する。証明書発行エンジン２１４は、中間表現２０が所定の規定あるいは規約を満たしている場合に、中間表現２０に関連付けられる証明書４０を発行する。これらの要素の動作およびデータ構造などの詳細については後述する。

[0058] 入力部２０６は、コントローラ３００を操作するユーザの入力操作を受け付ける。ディスプレイ２０８は、プロセッサ２０２での処理結果などを表示する。

[0059] 通信インターフェイス２２２は、ソフトウェア開発装置１００とのデータ交換を担当する。通信インターフェイス２２２は、例えば、インターネットを介した通信ができるように、イーサネット（登録商標）ポートを含んでいてもよい。

[0060] なお、検証サーバ２００の全部または一部は、コンピュータ可読命令に相当する回路が組み込まれたＡＳＩＣなどのハードワイヤード回路を用いて実現してもよい。さらにあるいは、ＦＰＧＡ上にコンピュータ可読命令に相当する回路を用いて実現してもよい。また、プロセッサ２０２およびメインメモリ、ＡＳＩＣ、ＦＰＧＡなどを適宜組み合わせて実現してもよい。

[0061] なお、実行に必要なプログラムは、メディアを介して検証サーバ２００にインストールされるだけではなく、ネットワーク上の配信サーバから提供されるようにしてもよい。

[0062] ＜Ｃ．検証処理＞

次に、本実施の形態に従うソフトウェア開発システム１における検証処理について説明する。

[0063] 上述したように、本実施の形態における中間表現２０に対する検証は、中間表現２０が所定の規定あるいは規約を満たすか否かを判断する処理を含む。

[0064] 所定の規定あるいは規約としては、アプリケーションプログラム（中間表現２０および対応する実行コード３０）が安全であるための条件を含む。このような安全であるための条件としては、典型的には、以下のようなものが挙げられる。

[0065] （１）安全性が確認できないメモリ操作（あるいは、危険なメモリ操作）が含まれていないこと（あるいは、そのようなメモリ操作がコンパイル可能ではないこと）

（２）コード中に未定義動作が存在しないこと

（３）エラー処理構文が欠如していないこと（エラー処理構文が適切に含まれていること）

（４）無秩序なエラー処理が含まれていないこと

上記（１）の安全性が確認できないメモリ操作（あるいは、危険なメモリ操作）は、システム領域やセキュア領域に対するメモリアクセスや、メモリ領域に対するブロック消去やブロック書き込みなどのメモリ操作を含む。

[0066] 上記（２）の未定義動作は、コード中に処理が定義されていないプロシージャやファンクションを含む。

[0067] 上記（３）のエラー処理構文は、例外が発生した場合の処理を規定する記述を含む。

上記（４）の無秩序なエラー処理は、例外が発生した場合の過度なエラー処理などを含む。

[0068] さらに、所定の規定あるいは規約としては、アプリケーションプログラム（中間表現２０および対応する実行コード３０）に、いわゆるバックドアが存在しないことを含む。バックドアは、アプリケーションプログラムの利用者が認識しない方法で、（通常は、悪意のある）第三者がアプリケーションプログラムにアクセスするための機能や接続口などを意味する。中間表現２０に対する検証においては、外部からのアクセス経路が中間表現２０に含まれるか否かが判断される。

[0069] 上述したような検証処理は、所定の規定あるいは規約を含む評価ルール２１６（図５）を参照することで実現するようにしてもよいし、いわゆるＡＩ（Artificial Intelligence）などの機械学習を用いて、検証処理を実現するための検証モデルを逐次更新するようにしてもよい。

[0070] 本実施の形態に従うソフトウェア開発システム１における検証処理については、検証サーバ２００で実行するようにしてもよいし、ソフトウェア開発装置１００において実行するようにしてもよい。検証サーバ２００で実行する場合には、クラウドコンピューティングの形態で実装してもよい。さらに、ソフトウェア開発装置１００において検証処理を実行する場合には、外部の認証機関により認証された評価ルール２１６を用いるようにしてもよい。

[0071] 本実施の形態に従うソフトウェア開発システム１における検証処理につい

ては、所定の規定あるいは規約を満たさないと判断された内容の詳細をユーザなどの通知するようにしてもよい。

[0072] ＜Ｄ．処理手順＞

次に、本実施の形態に従うソフトウェア開発システム１における処理手順について説明する。

[0073] 図６は、本実施の形態に従うソフトウェア開発システム１における処理手順の一例を示すフローチャートである。図６に示すソフトウェア開発装置１００が実行する処理については、典型的には、ソフトウェア開発装置１００のプロセッサ１０２がソフトウェア開発プログラム１１４を実行することで実現されてもよい。同様に、検証サーバ２００が実行する処理については、典型的には、検証サーバ２００のプロセッサ２０２が必要なプログラムを実行することで実現されてもよい。

[0074] 図６を参照して、ソフトウェア開発装置１００は、ユーザ操作に従ってソースコード１０を作成する（ステップＳ１００）。ソースコード１０の作成後、ソフトウェア開発装置１００は、ユーザ操作に従って、ソースコード１０をコンパイルして、ソースコード１０から中間表現２０を生成する（ステップＳ１０２）。

[0075] ソフトウェア開発装置１００は、生成した中間表現２０に対する検証を検証サーバ２００へ依頼し（ステップＳ１０４）、検証サーバ２００からの応答を待つ（ステップＳ１０６）。検証サーバ２００は、ソフトウェア開発装置１００からの検証依頼に応答して、依頼された中間表現２０に対する検証を行う（ステップＳ２００）。

[0076] すなわち、ステップＳ１０４およびＳ２００においては、中間表現２０が所定の規定あるいは規約を満たしているか否かを検証する処理が実行される。

[0077] 検証結果が適切であれば（ステップＳ２０２においてＹＥＳ）、ソフトウェア開発装置１００は、中間表現２０に対する認証をソフトウェア開発装置１００へ応答する（ステップＳ２０４）。一方、検証結果が適切でなければ

（ステップＳ２０２においてＮＯ）、ソフトウェア開発装置１００は、中間表現２０に対する認証を与えず、検証結果が不適切であった旨をソフトウェア開発装置１００へ応答する（ステップＳ２０６）。

[0078] 検証サーバ２００から中間表現２０に対する認証が応答されると（ステップＳ１０８においてＹＥＳ）、ソフトウェア開発装置１００は、対象の中間表現２０から実行コード３０を生成する（ステップＳ１１０）。そして、ソフトウェア開発装置１００は、生成した実行コード３０をターゲットのコントローラ３００へ転送する（ステップＳ１１２）。そして、処理は終了する。

[0079] このように、中間表現２０が所定の規定あるいは規約を満たしている場合に、中間表現２０から生成される実行コード３０の実行が実現される。

[0080] 一方、検証サーバ２００から中間表現２０に対する認証が応答されなければ（ステップＳ１０８においてＮＯ）、ソフトウェア開発装置１００は、ソースコード１０または中間表現２０が所定の規定あるいは規約を満たしていない旨をユーザに通知する（ステップＳ１１４）。このように、中間表現２０が所定の規定あるいは規約を満たしていない場合に、所定の規定あるいは規約を満たしていない内容をユーザに通知する処理を採用してもよい。そして、処理は終了する。

[0081] ＜Ｅ．実行環境＞

本実施の形態に従うソフトウェア開発システム１において生成された実行コード３０は、配布先のコントローラ３００を構成するプロセッサあるいはマイクロコンピュータのセキュア領域に格納されることが好ましい。すなわち、中間表現２０に対する認証を条件として生成された実行コード３０に対して、何らかの改ざん行為が行われないように、実行環境において何らかの対処を行うことが好ましい。

[0082] また、本実施の形態に従うソフトウェア開発システム１において生成された中間表現２０に対する証明書４０および中間表現２０から生成される実行コード３０のいずれにも共通の署名（例えば、認証された中間表現２０から

生成されるハッシュ値）を埋め込んでおき、対応する証明書４０が存在することを条件としてもよい。この場合には、実行コード３０を実行する環境（典型的には、マイクロコンピュータのファームウェア）において、実行コード３０に含まれる署名と、対応する証明書４０に含まれる署名との一致を確認する処理を実装してもよい。

[0083] また、実行環境において、実行コード３０を実行するか否かの評価（正当性の判断）は、いずれのタイミングで行ってもよいが、典型的には、実行環境であるコントローラ３００の起動時（ファームウェアの起動中あるいは起動直後のアプリケーション実行開始前）に行うようにしてもよい。すなわち、実行コード３０が実行されるコントローラ３００の起動時に、その実行コード３０を生成した中間表現２０に関連付けて発行された証明書４０に基づいて、その実行コード３０の正当性を評価するようにしてもよい。コントローラ３００の起動時に、アプリケーションプログラムの実行コード３０が認証されていることを確認することで、コントローラ３００などで好ましくない処理が実行される事態を回避できる。

[0084] ＜Ｆ．変形例＞

ソフトウェア開発装置１００から検証サーバ２００へ中間表現２０の検証を依頼する際には、対象の中間表現２０を暗号化してもよい。これによって、ソフトウェア開発装置１００と検証サーバ２００との間で遣り取りされる中間表現２０に対する改ざんなどを防止できる。

[0085] 検証サーバ２００が発行する証明書４０には、認証局からの署名などを付与するようにしてもよい。これによって、検証サーバ２００が発行する証明書４０に対する改ざんなどの不正行為を防止できる。

[0086] 上述の説明においては、ソフトウェア開発装置１００がソースコード１０の生成処理、中間表現２０の生成処理および実行コード３０の生成処理を実行する構成例について説明した、これらの処理を複数のコンピューティングデバイスで分散的に実行するようにしてもよい。

[0087] 本実施の形態に従う中間表現２０に対する検証および検証結果を示す証明

書４０の発行というフレームワークは、アプリケーションを販売あるいは配布する仕組みにも応用可能である。例えば、アプリケーションの開発者がアプリケーションプログラム（実行コード３０）をアプリケーション配布サーバにアップロードする際には、対応する中間表現２０および／または証明書４０を併せてアップロードすることを要求することで、アプリケーション配布サーバ側において、配布予定の実行コード３０を直接的または間接的に検証できる。このように、本実施の形態に従う仕組みを導入することで、バックドアなどが仕組まれたアプリケーションプログラムが配布されるような事態を回避できる。

[0088] ＜Ｇ．利点＞

本実施の形態においては、ソースコード１０ではなく中間表現２０に対する検証を行うことで、ソースコード１０に含まれるノウハウなどを秘匿化した状態で、アプリケーションプログラムに含まれる安全性についてのリスクをより効率的に発見できる。また、中間表現２０に対する検証を行うことで、ソースコード１０の記述に用いられたプログラミング言語の種類に応じた複数の検証ルールセットなどを用意する必要がなく、効率的な検証を実現できる。

[0089] 今回開示された実施の形態はすべての点で例示であって制限的なものでないと考えられるべきである。本発明の範囲は上記した説明ではなくて請求の範囲によって示され、請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

符号の説明

[0090] １ ソフトウェア開発システム、２ インターネット、４ エッジデバイス、１０ ソースコード、２０ 中間表現、３０ 実行コード、４０ 証明書、１００ ソフトウェア開発装置、１０２，２０２ プロセッサ、１０４，２０４ メインメモリ、１０６，２０６ 入力部、１０８，２０８ ディスプレイ、１１０，２１０ ハードディスク、１１４ ソフトウェア開発プログラム、１２０，２２０ 内部バス、１２２，２２２ 通信インターフェ

イス、２００　検証サーバ、２１２　評価エンジン、２１４　証明書発行エンジン、２１６　評価ルール、３００　コントローラ。

請求の範囲

[請求項1] ユーザ操作に従ってソースコードを作成するステップと、

前記ソースコードから中間表現を生成するステップと、

前記中間表現が所定の規定あるいは規約を満たしているか否かを検証するステップと、

前記中間表現が前記所定の規定あるいは規約を満たしている場合に、前記中間表現から生成される実行コードの実行を実現するステップとを備える、実行コード提供方法。

[請求項2] 前記実行コードの実行を実現するステップは、前記中間表現が前記所定の規定あるいは規約を満たしていることを条件として、前記中間表現から前記実行コードを生成するステップを含む、請求項１に記載の実行コード提供方法。

[請求項3] 前記実行コードの実行を実現するステップは、前記中間表現が前記所定の規定あるいは規約を満たしていることを条件として、前記中間表現から生成された前記実行コードの実行を許可するステップを含む、請求項１に記載の実行コード提供方法。

[請求項4] 前記検証するステップは、前記中間表現が前記所定の規定あるいは規約を満たしている場合に、前記中間表現に関連付けられる証明書を発行するステップを含み、

前記中間表現から生成された前記実行コードの実行を許可するステップは、前記証明書に基づいて実行の可否を判断するステップを含む、請求項３に記載の実行コード提供方法。

[請求項5] 前記実行コードが実行されるコントローラの起動時に、当該実行コードを生成した中間表現に関連付けて発行された証明書に基づいて、当該実行コードの正当性を評価するステップをさらに備える、請求項３または４に記載の実行コード提供方法。

[請求項6] 前記検証するステップは、前記中間表現が前記所定の規定あるいは規約を満たしていない場合に、前記所定の規定あるいは規約を満たし

ていない内容をユーザに通知するステップを含む、請求項１～５のいずれか１項に記載の実行コード提供方法。

[請求項7] ソフトウェア開発システムであって、

ソースコードの作成を支援するソフトウェア開発装置と、

前記ソフトウェア開発装置からアクセス可能な検証サーバとを備え、

前記ソフトウェア開発装置は、前記ソースコードから中間表現を生成する手段を備え、

前記検証サーバは、前記中間表現が所定の規定あるいは規約を満たしているか否かを検証する手段を備え、

前記中間表現が前記所定の規定あるいは規約を満たしている場合に、前記中間表現から生成される実行コードの実行が実現される、ソフトウェア開発システム。

[図1]

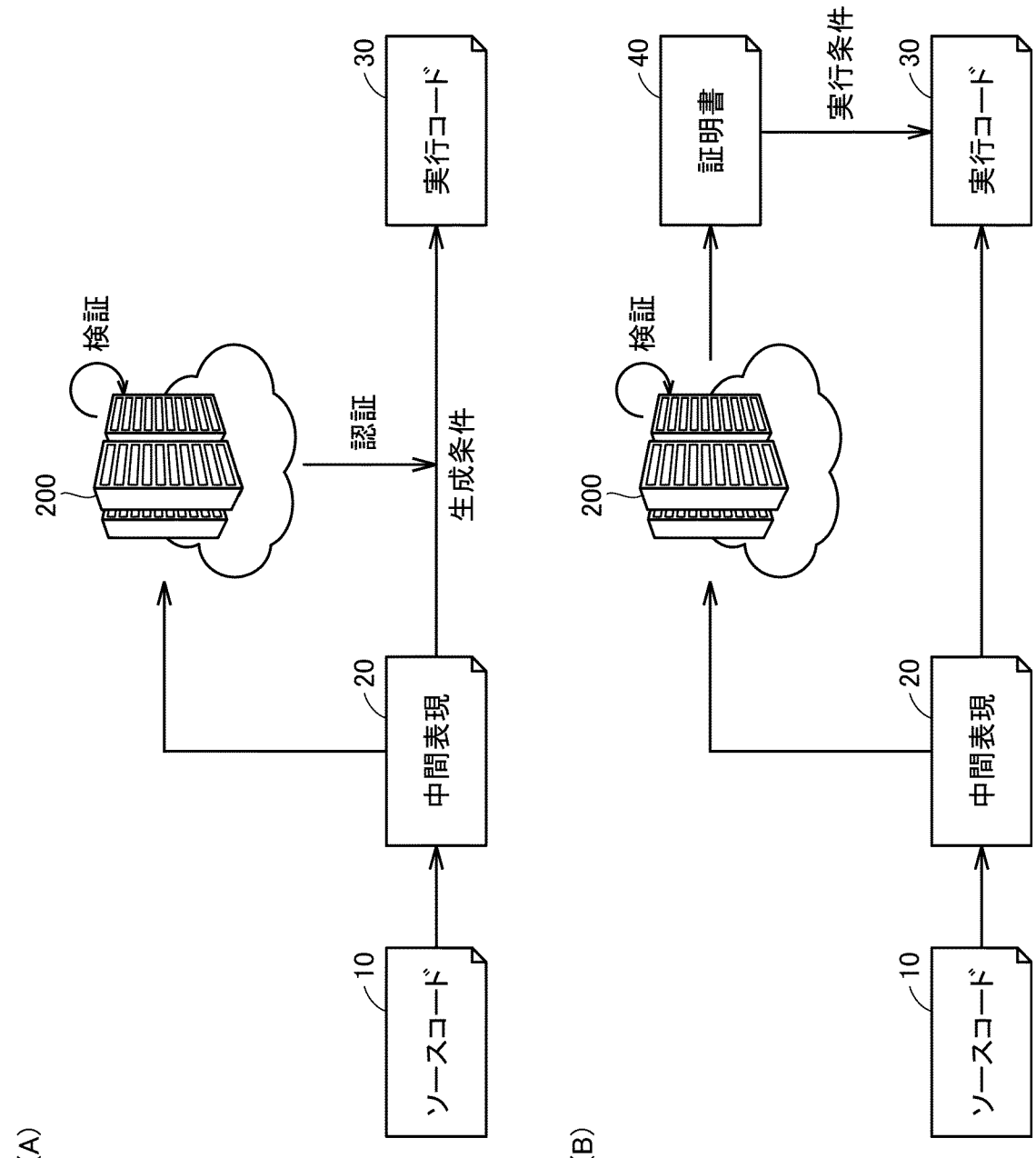

(A)
ソースコード
10
中間表現
20
実行コード
30
検証
200
認証
生成条件
(B)
ソースコード
10
中間表現
20
証明書
40
実行条件
実行コード
30
検証
200

FIG.1

[図2]

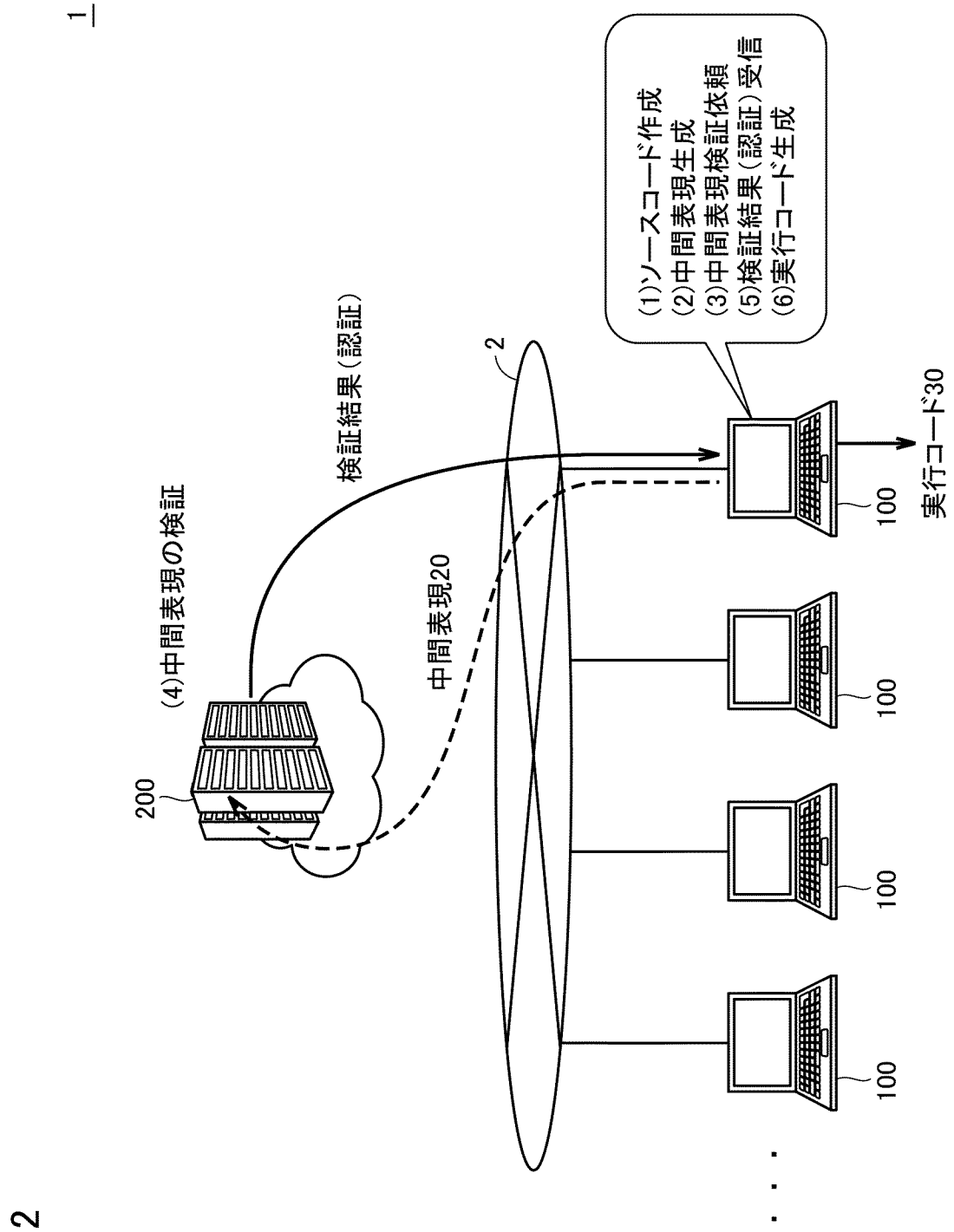
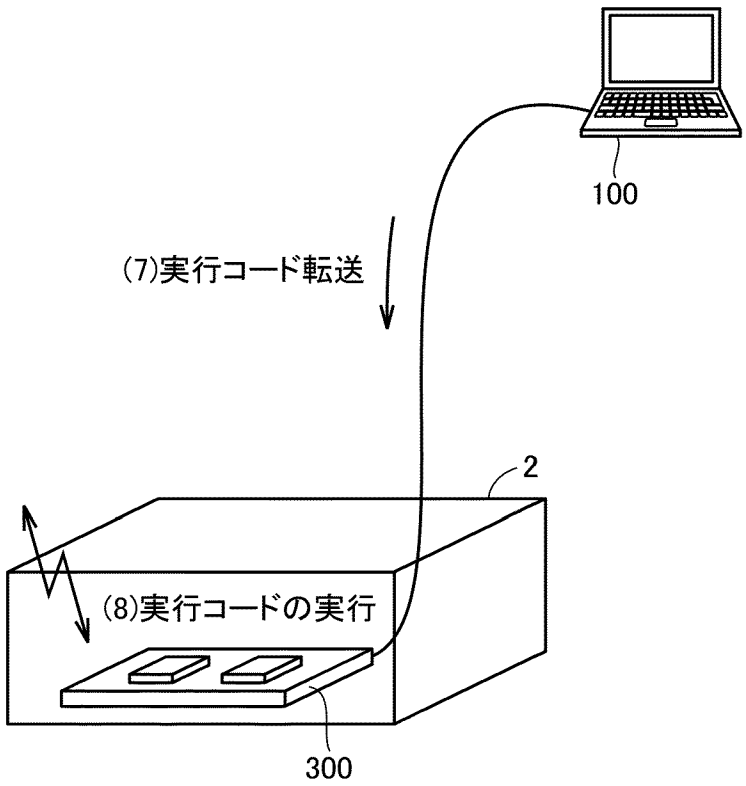

1
(1)ソースコード作成
(2)中間表現生成
(3)中間表現検証依頼
(5)検証結果(認証)受信
(6)実行コード生成
2
検証結果(認証)
(4)中間表現の検証
中間表現20
200
100
100
100
100
実行コード30

FIG.2

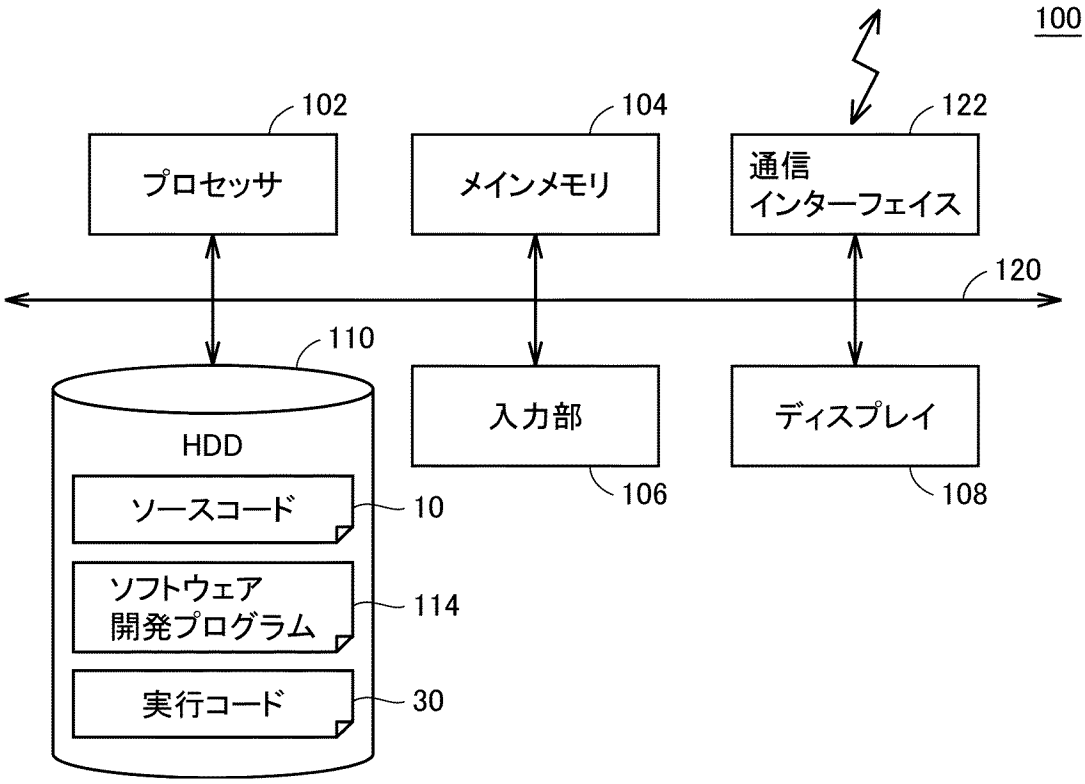
[図3]

FIG.3



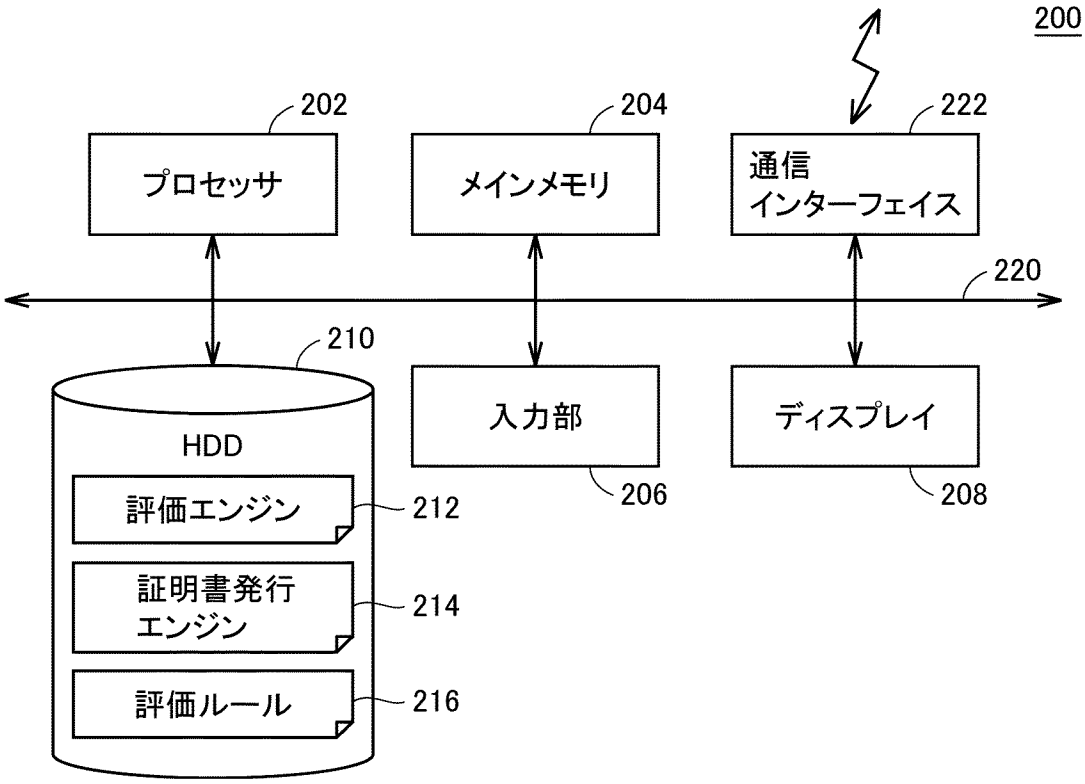
[図4]

FIG.4



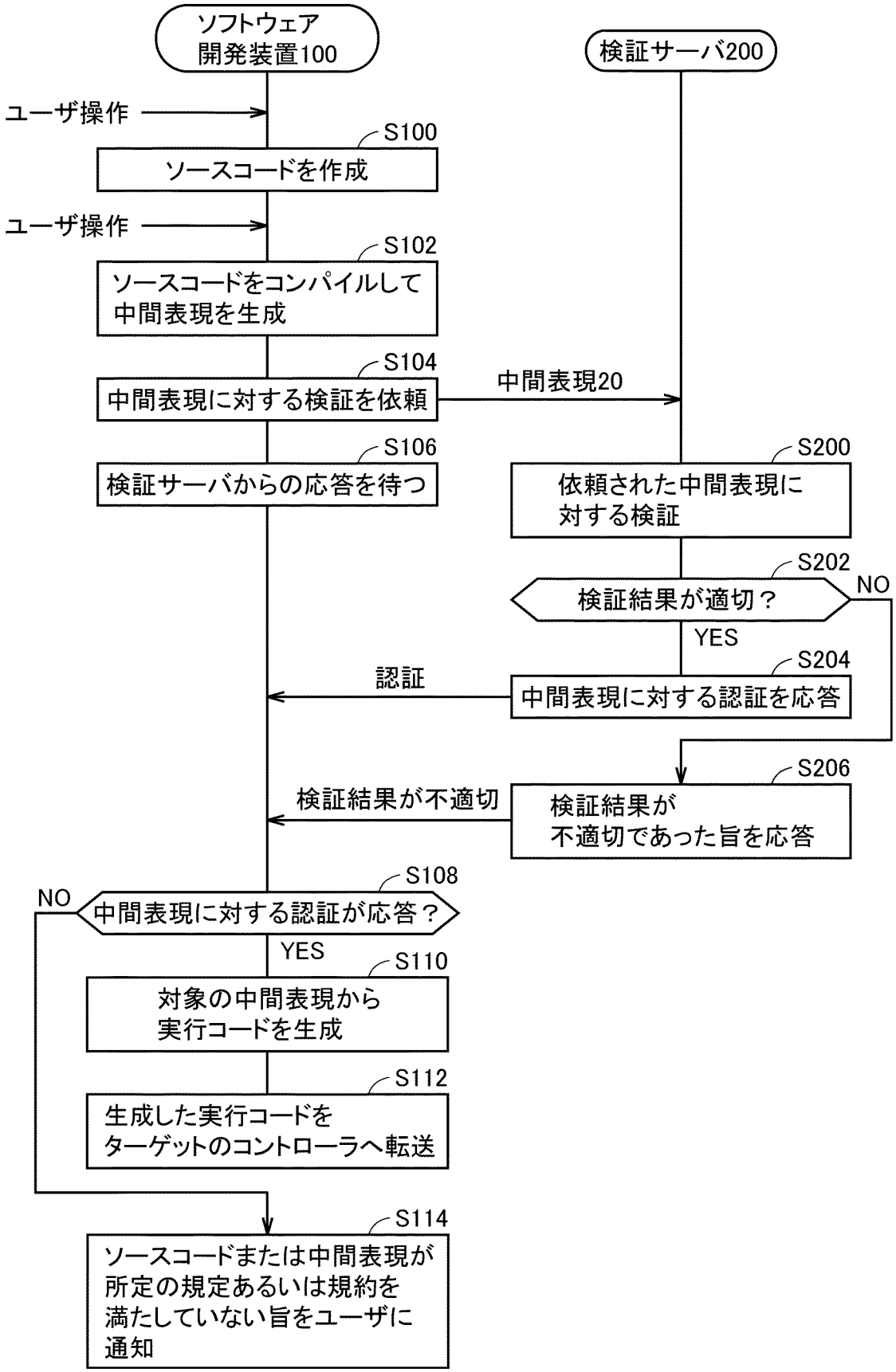
[図5]

FIG.5


200
202
プロセッサ
204
メインメモリ
222
通信
インターフェイス
220
210
HDD
評価エンジン
212
証明書発行
エンジン
214
評価ルール
216
入力部
206
ディスプレイ
208

[図6]

FIG.6


ソフトウェア
開発装置100
検証サーバ200
ユーザ操作
S100
ソースコードを作成
ユーザ操作
S102
ソースコードをコンパイルして
中間表現を生成
S104
中間表現に対する検証を依頼
中間表現20
S106
検証サーバからの応答を待つ
S200
依頼された中間表現に
対する検証
S202
検証結果が適切？
NO
YES
S204
認証
中間表現に対する認証を応答
S206
検証結果が不適切
検証結果が
不適切であった旨を応答
S108
NO
中間表現に対する認証が応答？
YES
S110
対象の中間表現から
実行コードを生成
S112
生成した実行コードを
ターゲットのコントローラへ転送
S114
ソースコードまたは中間表現が
所定の規定あるいは規約を
満たしていない旨をユーザに
通知

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2020/024799

A. CLASSIFICATION OF SUBJECT MATTER
G06F 8/00(2018.01)i; G06F 8/41(2018.01)i; G06F 21/57(2013.01)i
FI: G06F21/57; G06F8/00; G06F8/41

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)
G06F8/00; G06F8/41; G06F21/57

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	1971-2020
Registered utility model specifications of Japan	1996-2020
Published registered utility model applications of Japan	1994-2020

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.
X Y	JP 2001-75814 A (INTERNATIONAL BUSINESS MACHINES CORP.) 23.03.2001 (2001-03-23) paragraphs [0119]-[0126]	1-3,7 4-6
Y	JP 2006-522972 A (GEMPLUS) 05.10.2006 (2006-10-05) paragraphs [0025]-[0026], [0031]-[0034]	4-6
Y	JP 2003-122730 A (NTT COMWARE CORPORATION) 25.04.2003 (2003-04-25) paragraphs [0023]-[0024]	4-6
Y	JP 4-312130 A (NEC CORP.) 04.11.1992 (1992-11-04) column 2, lines 30-32	6

☐ 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	Date of mailing of the international search report
03 September 2020 (03.09.2020)	15 September 2020 (15.09.2020)

Name and mailing address of the ISA/	Authorized officer
Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/JP2020/024799

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
JP 2001-75814 A	23 Mar. 2001	(Family: none)	
JP 2006-522972 A	05 Oct. 2006	US 2007/0043957 A1 paragraphs [0037]-[0038], [0043]-[0046] WO 2004/090718 A1 EP 1616256 A1 CN 1802631 A	
JP 2003-122730 A	25 Apr. 2003	(Family: none)	
JP 4-312130 A	04 Nov. 1992	(Family: none)	

国際調査報告

国際出願番号
PCT/JP2020/024799

A. 発明の属する分野の分類（国際特許分類（ＩＰＣ））

G06F 8/00(2018.01)i; G06F 8/41(2018.01)i; G06F 21/57(2013.01)i
FI: G06F21/57; G06F8/00; G06F8/41

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））

G06F8/00; G06F8/41; G06F21/57

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	１９７１－２０２０年
日本国実用新案登録公報	１９９６－２０２０年
日本国登録実用新案公報	１９９４－２０２０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
X	JP 2001-75814 A (インターナショナル・ビジネス・マシーンズ・コーポレーション) 23.03.2001 (2001-03-23) 段落0119-0126	1-3,7
Y		4-6
Y	JP 2006-522972 A (ジェムプリュス) 05.10.2006 (2006-10-05) 段落0025-0026, 0031-0034	4-6
Y	JP 2003-122730 A (エヌ・ティ・ティ・コムウェア株式会社) 25.04.2003 (2003-04-25) 段落0023-0024	4-6
Y	JP 4-312130 A (日本電気株式会社) 04.11.1992 (1992-11-04) 2欄30-32行	6

☐ 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	国際調査報告の発送日
03.09.2020	15.09.2020

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）
行田　悦資 5S 6304

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/JP2020/024799

引用文献	公表日	パテントファミリー文献	公表日
JP 2001-75814 A	23.03.2001	（ファミリーなし）	
JP 2006-522972 A	05.10.2006	US 2007/0043957 A1 段落0037-0038, 0043-0046 WO 2004/090718 A1 EP 1616256 A1 CN 1802631 A	
JP 2003-122730 A	25.04.2003	（ファミリーなし）	
JP 4-312130 A	04.11.1992	（ファミリーなし）	