

(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
2022年12月1日(01.12.2022)



(10) 国際公開番号

WO 2022/249256 A1

(51) 国際特許分類:
G06F 8/70 (2018.01) G06N 20/00 (2019.01)

京都武蔵野市緑町3丁目9-11 NTT
知的財産センタ内 Tokyo (JP).

(21) 国際出願番号: PCT/JP2021/019661

(22) 国際出願日: 2021年5月24日(24.05.2021)

(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).

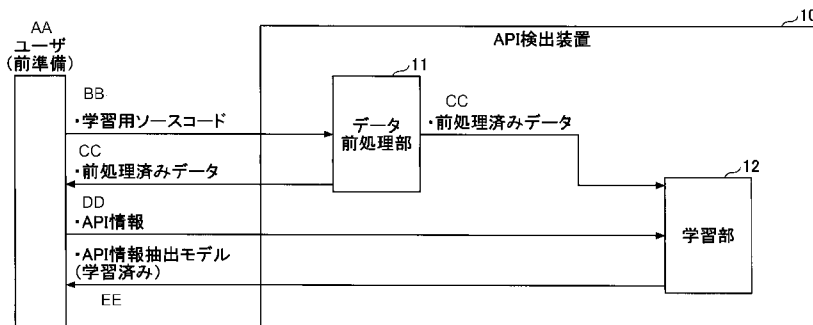
(72) 発明者: 倉林 利行(KURABAYASHI, Toshiyuki); 〒1808585 東京都武蔵野市緑町3丁目9-11 NTT 知的財産センタ内 Tokyo (JP). 岩塚 卓弥(IWATSUKA, Takuya); 〒1808585 東

(74) 代理人: 伊東 忠重, 外(ITO, Tadashige et al.); 〒1000005 東京都千代田区丸の内二丁目1番1号 丸の内 M Y P L A Z A (明治安田生命ビル) 16階 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, 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, IT, JO, JP, 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, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,

(54) Title: API DETECTION DEVICE, API DETECTION METHOD, AND PROGRAM

(54) 発明の名称: API 検出装置、API 検出方法及びプログラム



10 API detection device
11 Data preprocessing unit
12 Training unit
AA User (Preliminary preparation)
BB Source code for training
CC Preprocessed data
DD API information
EE API information extraction model (trained)

(57) Abstract: This API detection device increases the efficiency of API detection by having: a training unit for training a model on the basis of training data in which a prescribed label is added to a token within a token string pertaining to a first source code, the token being an API configuration component, and the model detecting the configuration component from within an inputted token string; and an API detection unit for inputting a token string pertaining to a second source code to the model, thereby causing the model to detect an API configuration component included in the second source code.

(57) 要約: API 検出装置は、第1のソースコードに係るトークン列のうちAPIの構成要素であるトークンに対して所定のラベルが付与された学習データに基づいて、入力されたトークン列の中から前記構成要素を検出するモデルを学習する学習部と、第2のソースコードに係るトークン列を前記モデルへ入力することで、前記第2のソースコードに含まれるAPIの構成要素を前記モデルに検出させるAPI検出部と、を有することで、APIの検出を効率化する。

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))

明 細 書

発明の名称： A P I 検出装置、 A P I 検出方法及びプログラム
技術分野

[0001] 本発明は、 A P I 検出装置、 A P I 検出方法及びプログラムに関する。

背景技術

[0002] 多様化する消費者ニーズに合ったサービスを迅速に提供するために、近年、ソフトウェア開発を短いサイクルで実施し、頻繁にサービスをリリースする開発スタイルが増加している。また、このような短いサイクルでのサービス提供を維持しながら品質を確保することも求められている。ソフトウェア開発において、テストは開発の終盤で行われることが多く、開発終盤で脆弱性が発見されると大きな手戻りが発生するという問題がある。そこで実装段階でテストを前倒しで行う、シフトレフトと呼ばれる考え方が存在する。

[0003] 従来、テスト対象を変更箇所だけに絞ってテストする技術が存在する（非特許文献1）。この技術では、静的解析により、変更箇所とその変更箇所を実行するための A P I を検出してテストを行う。

先行技術文献

非特許文献

[0004] 非特許文献1：岩塚 卓弥，倉林 利行，切貫 弘之、"テストプロセスの変革～セキュリティテストと探索的テスト～"、ビジネスコミュニケーション 2020 Vol.57 No.2、[online]、インターネット<URL : https://www.sic.ecl.ntt.co.jp/mt_assets/bc_202002/bc03.pdf>

発明の概要

発明が解決しようとする課題

[0005] A P I は、ソフトウェアが使用する W e b フレームワーク（以下「FW」という。）によって記法が異なるため、FWごとに A P I を検出するための個別の解析ルールを設定する必要がある。しかしながら、現代社会において、W e b は非常に需要が高く、FWも様々な種類が存在するため、あらゆる

F Wに対応するように個別に解析ルールを設定するためには大きなコストがかかる。またW e b技術は変化の激しい分野であり、新しいF Wや、同一F Wでも新しい記法が登場する可能性が高い。したがって、非特許文献1の技術では、様々なF Wに対応させてA P Iを検出するために膨大なコストを要する可能性が高い。

[0006] 本発明は、上記の点に鑑みてなされたものであって、A P Iの検出を効率化することを目的とする。

課題を解決するための手段

[0007] そこで上記課題を解決するため、A P I検出装置は、第1のソースコードに係るトークン列のうちA P Iの構成要素であるトークンに対して所定のラベルが付与された学習データに基づいて、入力されたトークン列の中から前記構成要素を検出するモデルを学習する学習部と、第2のソースコードに係るトークン列を前記モデルへ入力することで、前記第2のソースコードに含まれるA P Iの構成要素を前記モデルに検出させるA P I検出部と、を有する。

発明の効果

[0008] A P Iの検出を効率化することができる。

図面の簡単な説明

[0009] [図1]本発明の実施の形態におけるA P I検出装置10のハードウェア構成例を示す図である。

[図2]本発明の実施の形態におけるA P I検出装置10の学習時の機能構成例を示す図である。

[図3]A P I検出装置10が学習時に実行する処理手順の一例を説明するためのフローチャートである。

[図4]本発明の実施の形態におけるA P I検出装置10のA P Iの検出時の機能構成例を示す図である。

[図5]A P I検出装置10がA P I検出時に実行する処理手順の一例を説明するためのフローチャートである。

発明を実施するための形態

[0010] 本実施の形態は、非特許文献1においてWebフレームワーク（以下「FW」という。）におけるWebAPI（以下、単に「API」という。）の記法を学習することで、少ないコストで様々なFWにおけるAPIの検出を可能とする。FWのAPI記法は大きく2つの種別にわけることができ、1つ目はアノテーションで記述する方法（以下、「アノテーション記法」という。）、2つ目は関数で記述する方法（以下、「関数記法」という。）である。アノテーション記法に分類されるFWのAPIの記法には類似する点が存在する。例えば、アノテーション記法のFWの1つであるMicronautは、以下に示すようにAPIを記述する。

```
@app.route("/test", methods=["GET"])
```

```
def test():
```

このコントローラは、`/test`宛てにGETリクエストを受けたときに呼び出される。すなわち、パス及びHTTPメソッドがAPIの構成要素である。したがって、パスとして、`/test`が検出でき、HTTPメソッドとして`GET`が検出できればAPIを検出できることになる。

[0011] 同じくアノテーション記法のFWの1つであるSymfonyでは、以下のようにAPIを記述する。

```
/**
```

```
* @Route("/test", methods={"GET"})
```

```
*/
```

```
public function test(){
```

それぞれ細部は異なるものの、「route」というキーワードの後にパス、「methods」というキーワードの後にHTTPメソッドが記述されている。その他、パスに関するキーワードとして「Controller」や「Path」、HTTPメソッドに関するキーワードとして「GET」や「POST」などが存在する。

[0012] このように、相互に異なる複数のAPI記法には類似性が存在するため、

アノテーション記法に分類される複数のFWでAPI記法を学習すれば、学習外のアノテーション記法に分類されるFWでもAPIが検出できると考えられる。関数記法も同様である。本実施の形態では、この類似性に着目し、アノテーション記法と関数記法のそれぞれのAPI記述を学習することで、幅広いFWでのAPI検出を可能とする。

[0013] 以下、図面に基づいて本発明の実施の形態を説明する。図1は、本発明の実施の形態におけるAPI検出装置10のハードウェア構成例を示す図である。図1のAPI検出装置10は、それぞれバスBで相互に接続されているドライブ装置100、補助記憶装置102、メモリ装置103、CPU104、インタフェース装置105、表示装置106、及び入力装置107等を有する。

[0014] API検出装置10での処理を実現するプログラムは、CD-ROM等の記録媒体101によって提供される。プログラムを記憶した記録媒体101がドライブ装置100にセットされると、プログラムが記録媒体101からドライブ装置100を介して補助記憶装置102にインストールされる。但し、プログラムのインストールは必ずしも記録媒体101より行う必要はなく、ネットワークを介して他のコンピュータよりダウンロードするようにしてもよい。補助記憶装置102は、インストールされたプログラムを格納すると共に、必要なファイルやデータ等を格納する。

[0015] メモリ装置103は、プログラムの起動指示があった場合に、補助記憶装置102からプログラムを読み出して格納する。CPU104は、メモリ装置103に格納されたプログラムに従ってAPI検出装置10に係る機能を実現する。インタフェース装置105は、ネットワークに接続するためのインタフェースとして用いられる。表示装置106はプログラムによるGUI (Graphical User Interface) 等を表示する。入力装置107はキーボード及びマウス等で構成され、様々な操作指示を入力させるために用いられる。

[0016] 図2は、本発明の実施の形態におけるAPI検出装置10の学習時の機能構成例を示す図である。図2において、API検出装置10は、前処理部1

1 及び学習部 12 を有する。これら各部は、API 検出装置 10 にインストールされた 1 以上のプログラムが、CPU 104 に実行させる処理により実現される。

[0017] 学習時には、学習用ソースコードが用意され、API 検出装置 10 に入力される。学習用ソースコードのデータ構造を BNF (Backus-Naur form) 記法に基づく形式によって記すと以下の通りである。

学習用ソースコード ::= ソースコードファイル +

すなわち、学習用ソースコードは、学習対象として決定された任意の複数の FW の API を利用する（呼び出す）実際の或るプログラムのプロジェクトデータにおけるソースコードファイルの集合である。ここでは、相互に API の記法が異なる複数のプロジェクトデータが用意されることとする。したがって、学習用ソースコードは、アノテーション記法と関数記法とのそれぞれについて用意される。

[0018] 図 3 は、API 検出装置 10 が学習時に実行する処理手順の一例を説明するためのフローチャートである。図 3 の処理手順は、アノテーション記法に係る学習用ソースコードと、関数記法に係る学習用ソースコードとのそれぞれについて実行される。その結果、アノテーション記法に係る API 検出モデルと、関数記法に係る API 検出モデルとの 2 つのモデルが学習（生成）される。なお、図 3 では、アノテーション記法及び関数記法について共通の処理を示すため、記法の区別は行わない。

[0019] 図 3 の処理手順は、学習用ソースコードに含まれるソースコードファイルごとのループ処理 L1 を含む。以下、ループ処理 L1 において処理対象とされているソースコードファイルが格納するソースコードを「対象コード」という。ループ処理 L1 は、ループ処理 L2 を含む。ループ処理 L2 は、対象コードが含むブロックごとのループ処理である。以下、ループ処理 L2 において処理対象とされているブロックを「対象ブロック」という。

[0020] なお、ブロックとは、インデント長（字下げ幅）が同じである連続する行の集合をいう。例えば、

```
Def hoge():
```

```
    A=1
```

```
    B=2
```

というソースコードでは、「Def hoge():」が1つのブロックとなり、「A=1
B=2」が2つ目のブロックとなる。

- [0021] ステップS101において、前処理部11は、対象ブロックをトークナイズすることで、対象ブロックのトークン列を生成する。例えば、対象ブロックが以下の「入力」の通りである場合、以下の「出力」がトークナイズの結果であるトークン列として得られる。

```
入力：@Route("/test", methods={"GET"})
```

```
出力：@,Route,(, "/test",,, methods,=,{, "GET",},,)]
```

すなわち、トークン列は、対象ブロックに含まれる各トークンが、対象ブロックにおける出現順に配列されたデータであり、各トークンはカンマで区切られている。

- [0022] 続いて、前処理部11は、当該トークン列の各トークンについて抽象化を実行する（S102）。抽象化を実行することで学習の精度を上げるためである。抽象化には任意の方法が用いられればよい。以下の例では、小文字化と文字列のマスク処理とが施されている。マスク処理とは、例えば、プログラムの文法において予約されていない文字列（例えば、`"""`によって囲まれた文字列やコメント等）を所定の文字列（以下の例では`<string>`）に置換する処理をいう。以下は、入力として抽象化前のトークン列を示し、出力として抽象化後のトークン列を示す。

```
入力：[@,Route,(, "/test",,, methods,=,{, "GET",},,)]
```

```
出力：[@,route,(, <string>,,, methods,=,{, <string>,},,)]
```

続いて、前処理部11は、抽象化されたトークン列の各トークンに対するラベル付けを行うことで、トークンごとに、API情報を生成する（S103）。以下にラベル付けの一例を示す。

```
入力：[@,route,(, <string>,,, methods,=,{, <string>,},,)]
```

出力：[None, None, None, Path, None, None, None, None, HttpMethod, None, None]

すなわち、ラベル付けは、各トークンに対して、None、Path又はHttpMethodのラベルを付与することである。APIの構成要素であるパス及びHTTPメソッドのいずれでもないトークンに対しては、Noneが付与される。APIの構成要素であるトークンに対しては、所定のトークンが付与される。具体的には、パスであるトークンに対しては、Pathが付与される。HTTPメソッドであるトークンに対しては、HttpMethodが付与される。なお、API情報のデータ構造をBNF記法に基づく形式によって記すと以下の通りである。

API情報 ::= (トークン, ラベル) +

ラベル ::= (None | Path | HttpMethod)

すなわち、API情報とは、トークンと当該トークンに付与されたラベルとの組である。なお、ラベル付けは、例えば、ラベル付けのためのプログラム（スクリプト）等を用いて自動的に行われてもよいし、人手で行われてもよい。人手で行われる場合、例えば、前処理部11は、各トークンに対するラベルの入力をユーザから受け付けてもよい。

[0023] ループ処理L2が対象コードの全てのブロックについて終了し、ループ処理L1が学習用ソースコードの全てのソースコードファイルについて実行されると、学習部12は、ループ処理L1において生成されたAPI情報の系列（API情報がトークンの並び順で整列されたデータ）を学習データとして、API情報におけるトークンとラベルとの対応関係をAPI検出モデルに学習させる（S104）。API検出モデルは、任意のモデル（ニューラルネットワーク）を用いてよいが、ソースコードは時系列データであるため、LSTMやTransformer（Encoder部分）などがAPI検出モデルとして好適である。トークンの埋め込みにはWord Embeddingなどが用いられればよい。このようなモデルを利用することで、各トークンについて、当該トークンとラベルとの対応関係だけでなく、前後の文脈にも着目して各トークンに対応するラベルを推定可能とすることがで

きる。

[0024] 図3の処理手順が、アノテーション記法と関数記法のそれぞれについて終了すると、それぞれの記法に対応した学習済みのAPI検出モデルが生成される。

[0025] 図4は、本発明の実施の形態におけるAPI検出装置10のAPIの検出時の機能構成例を示す図である。図4において、API検出装置10は、変更ファイル抽出部13、前処理部14及びAPI検出部15を有する。これら各部は、API検出装置10にインストールされた1以上のプログラムが、CPU104に実行させる処理により実現される。

[0026] 図5は、API検出装置10がAPI検出時に実行する処理手順の一例を説明するためのフローチャートである。

[0027] ステップS201において、変更ファイル抽出部13は、アプリケーションのソースコードファイルの中から、開発者によって変更されたソースコードファイル（以下、「変更ファイル」という。）を抽出する。変更ファイルは、新バージョンについて旧バージョンに対する変更を含むファイルであってもよいし、例えば、開発の進行過程において、或る時点に対して行われた変更を含むファイルであってもよい。又は、新規に作成されたソースコードのファイルは、それ自体が変更ファイルである。

[0028] 続いて、変更ファイルごとに、ループ処理L3が実行される。以下、ループ処理L3において処理対象とされている変更ファイルが含むソースコードを「対象コード」という。ループ処理L3は、ループ処理L4を含む。ループ処理L4は、対象コードが含むブロックごとのループ処理である。以下、ループ処理L4において処理対象とされているブロックを「対象ブロック」という。

[0029] ステップS202において、前処理部14は、対象ブロックをトークナイズすることで、対象ブロックのトークン列を生成する。この処理内容は、図3のステップS101と同様である。

[0030] 続いて、前処理部14は、当該トークン列の各トークンについて抽象化を

実行する（S203）。この処理内容は、図3のステップS102と同様である。但し、ステップS203において、前処理部14は、抽象化されたN番目のトークンについて、抽象化前のトークンとNとを対応付けて補助記憶装置102又はメモリ装置103に保存しておく。すなわち、抽象化前後の対応関係が保持される。

[0031] ステップS202及びS203の実行により、対象ブロックが以下の入力の通りであれば、ステップS203では、以下の出力が得られる。

入力：`@app.route("/test", methods=["GET"])`

出力：`[@, app, route, (, <string>, , methods, =, [, <string>, ], )]`

この例では、`"/test"`が`<string>`に置換され、`"GET"`が`<string>`に置換されている。したがって、トークン列における`"/test"`の順番と`"/test"`との対応関係と、トークン列における`"GET"`の順番と`"GET"`との対応関係とが補助記憶装置102又はメモリ装置103に保存される。

[0032] 続いて、API検出部15は、ステップS202において得られた抽象化されたトークン列を学習済みのAPI検出モデルに入力することで、当該トークン列を構成する各トークンについて当該API検出モデルから出力されるラベルを取得する（S203）。この際、API検出部15は、当該トークン列（同一のトークン列）を、アノテーション記法について学習済みのAPI検出モデルと、関数記法について学習済みのAPI検出モデルとのそれぞれに入力し、それぞれから各トークンのラベルを取得する。API検出部15は、トークンごとに、当該トークンについて各API検出モデルから出力された2つのラベルのOR（論理和）を当該トークンのラベルとして判定する。2つのラベルのORとは、2つのラベルのうち少なくともいずれか一方が`Path`又は`HttpMethod`であり、他方が`None`である場合に、`Path`又は`HttpMethod`が採用されることをいう。2つのAPI検出モデルが使用されるのは、FWによってはアノテーション記法と関数記法の両方が存在するためである。以下に、ステップS203の入力と出力との例を示す。

入力 : [@, app, route, (, <string>, ,, methods, =, [, <string>,],)]

出力 : [None, None, None, None, Path, None, None, None, None, HttpMethod, None, None]

これによって、トークン列の各トークンに対するラベルが得られる。換言すれば、いずれのトークンがAPIの構成要素であるかが分かる。

[0033] 続いて、API検出部15は、当該トークン列において抽象化されたトークンを抽象化前の文字列に復元する(S205)。例えば、上記の入力が、以下のように復元される。

[@, app, route, (, "/test", ,, methods, =, [, "GET",],)]

続いて、API検出部15は、抽象化前の状態に復元されたトークン列のうち、ラベルがPathであるトークンとラベルがHttpMethodであるトークンとの組を含むAPI情報を生成し、当該API情報をテスト用APIへ追加する(S206)。テスト用APIのデータ構造をBNF記法に基づく形式によって記すと以下の通りである。

テスト用API ::= API情報 +

API情報 ::= <Path>, <HttpMethod>

すなわち、テスト用APIは、API情報のリストである。API情報は、Pathのラベルが推定されたトークンと、HttpMethodのラベルが推定されたトークンとの組である。例えば、上記の例において、API情報は、「"/test", "GET"」である。

[0034] ループ処理L4が対象コードの全てのブロックについて終了し、ループ処理L3が全ての変更ファイルについて終了すると、API検出部15は、テスト用APIの内容(API情報のリスト)を出力する(S207)。

[0035] ユーザは、出力されたAPI情報のリストを参照して、テストすべきAPIを特定することができる。例えば、上記の例では、ユーザは、「"/test"宛てにGETリクエストを送る」といったテストを実行すべきであることを認識することができる。このようなテストには、非特許文献1に開示された技術が利用されてもよい。

[0036] 上述したように、本実施の形態によれば、API記法を学習することで様々な種類のFWに対して解析ルールを設定することなくAPIを検出することができる。すなわち、APIの検出を効率化することができる。その結果、様々な種類のFWに対して少ないコストでテストを実施することができ、シフトレフトを実現できる。

[0037] なお、上記では、APIの記法として、アノテーション記法と関数記法を例として説明したが、他の記法が存在するAPIについては、3つ以上の記法に関して学習及び検出が行われてもよい。

[0038]

以上、本発明の実施の形態について詳述したが、本発明は斯かる特定の実施形態に限定されるものではなく、請求の範囲に記載された本発明の要旨の範囲内において、種々の変形・変更が可能である。

符号の説明

[0039]	1 0	API検出装置
	1 1	前処理部
	1 2	学習部
	1 3	変更ファイル抽出部
	1 4	前処理部
	1 5	API検出部
	1 0 0	ドライブ装置
	1 0 1	記録媒体
	1 0 2	補助記憶装置
	1 0 3	メモリ装置
	1 0 4	CPU
	1 0 5	インタフェース装置
	1 0 6	表示装置
	1 0 7	入力装置
	B	バス

請求の範囲

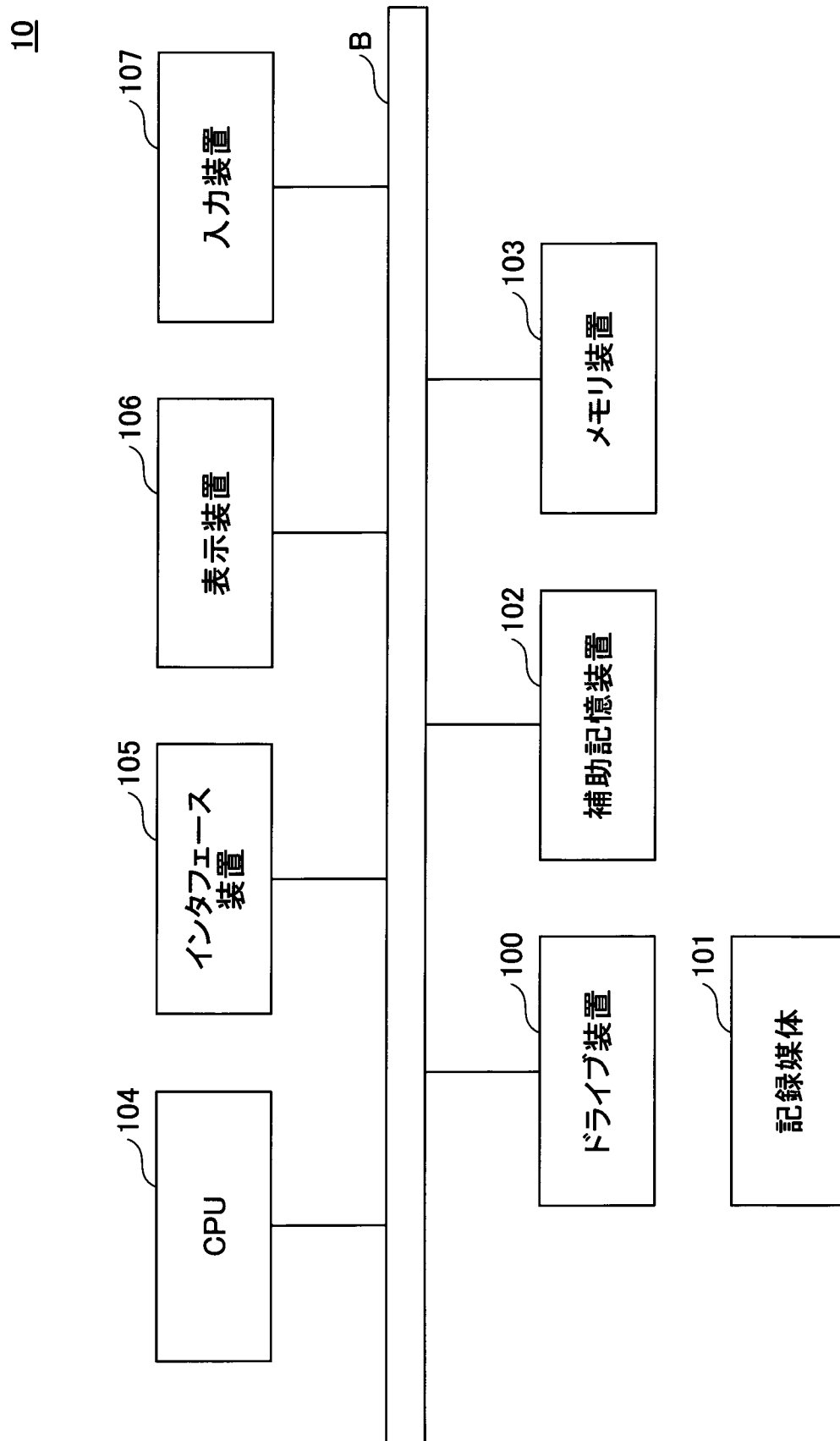
- [請求項1] 第1のソースコードに係るトークン列のうちA P Iの構成要素であるトークンに対して所定のラベルが付与された学習データに基づいて、入力されたトークン列の中から前記構成要素を検出するモデルを学習する学習部と、
- 第2のソースコードに係るトークン列を前記モデルへ入力することで、前記第2のソースコードに含まれるA P Iの構成要素を前記モデルに検出させるA P I検出部と、
- を有することを特徴とするA P I検出装置。
- [請求項2] 前記学習部は、A P Iに関する複数の記法の種別ごとに、当該種別に係る前記学習データに基づいて前記モデルを学習し、
- 前記A P I検出部は、前記第2のソースコードに係るトークン列を複数の前記モデルに入力することで、前記第2のソースコードに含まれるA P Iの構成要素を前記複数のモデルに検出させる、
- ことを特徴とする請求項1記載のA P I検出装置。
- [請求項3] 前記複数の記法の種別は、アノテーション記法又は関数記法を含む、
- ことを特徴とする請求項2記載のA P I検出装置。
- [請求項4] 第1のソースコードに係るトークン列のうちA P Iの構成要素であるトークンに対して所定のラベルが付与された学習データに基づいて、入力されたトークン列の中から前記構成要素を検出するモデルを学習する学習手順と、
- 第2のソースコードに係るトークン列を前記モデルへ入力することで、前記第2のソースコードに含まれるA P Iの構成要素を前記モデルに検出させるA P I検出手順と、
- をコンピュータが実行することを特徴とするA P I検出方法。
- [請求項5] 前記学習手順は、A P Iに関する複数の記法の種別ごとに、当該種別に係る前記学習データに基づいて前記モデルを学習し、

前記 A P I 検出手順は、前記第 2 のソースコードに係るトークン列を複数の前記モデルに入力することで、前記第 2 のソースコードに含まれる A P I の構成要素を前記複数のモデルに検出させる、ことを特徴とする請求項 4 記載の A P I 検出方法。

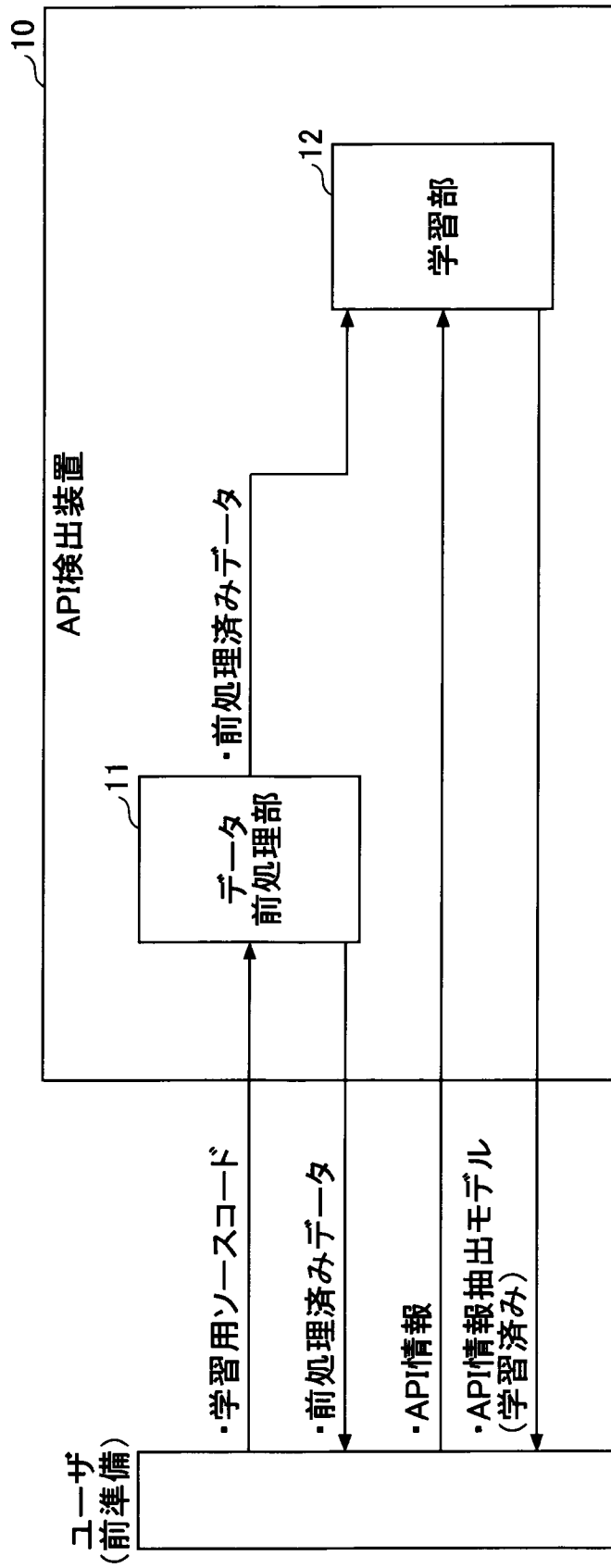
[請求項 6] 前記複数の記法の種別は、アノテーション記法又は関数記法を含む、
ことを特徴とする請求項 5 記載の A P I 検出方法。

[請求項 7] 請求項 4 乃至 6 いずれか一行記載の A P I 検出方法をコンピュータに実行させることを特徴とするプログラム。

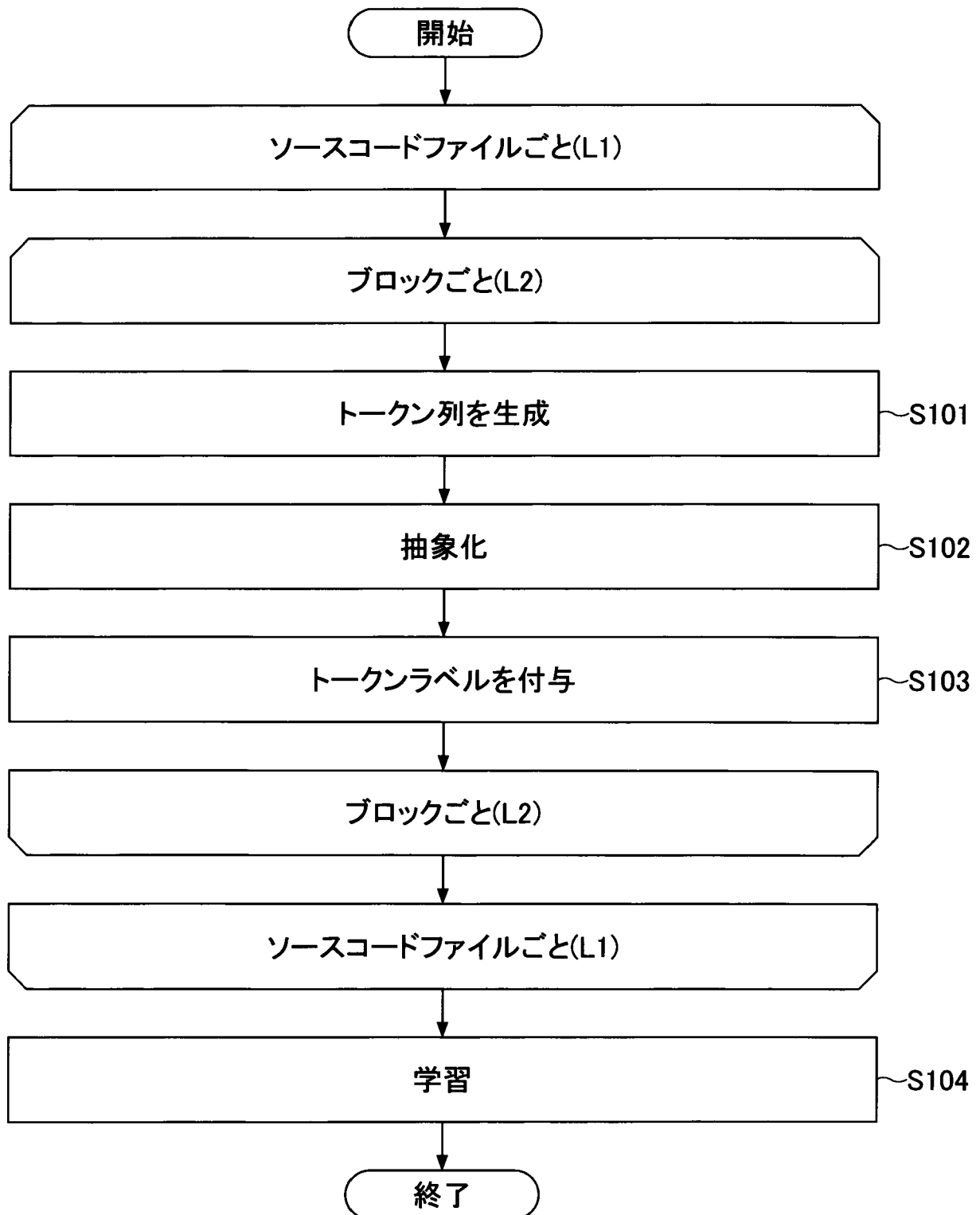
[図1]



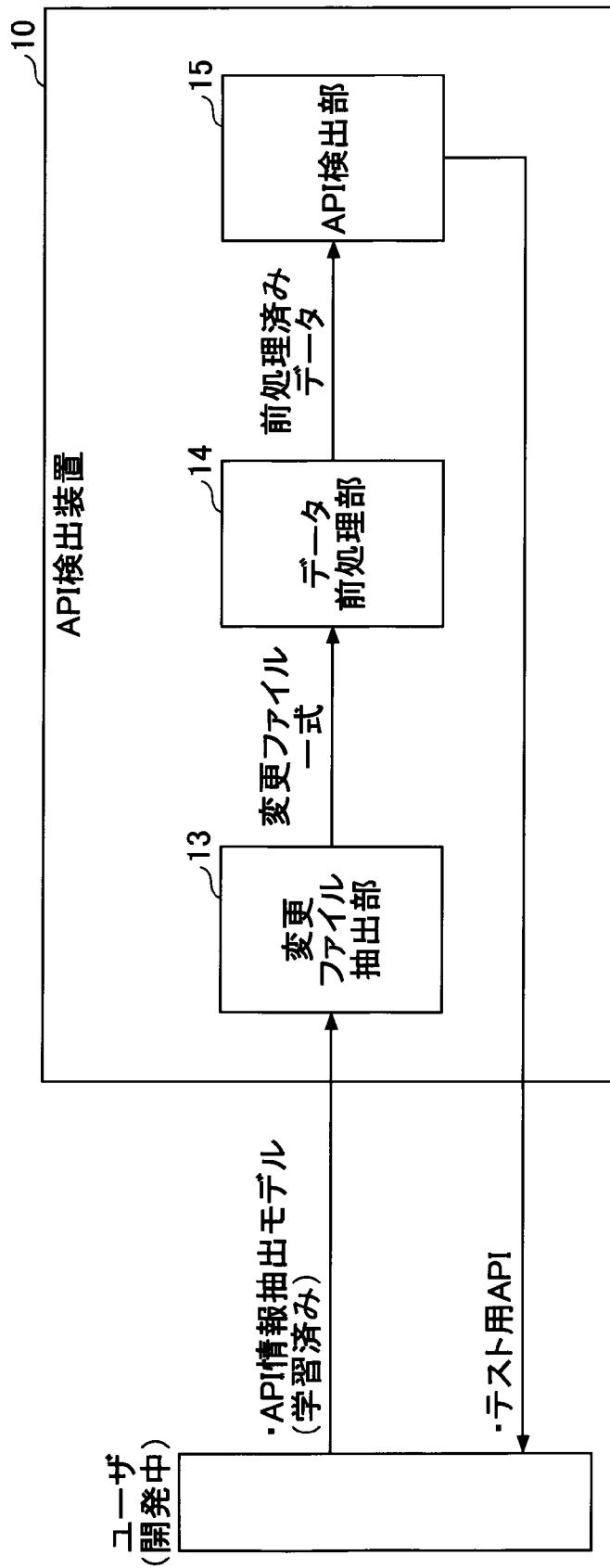
[図2]



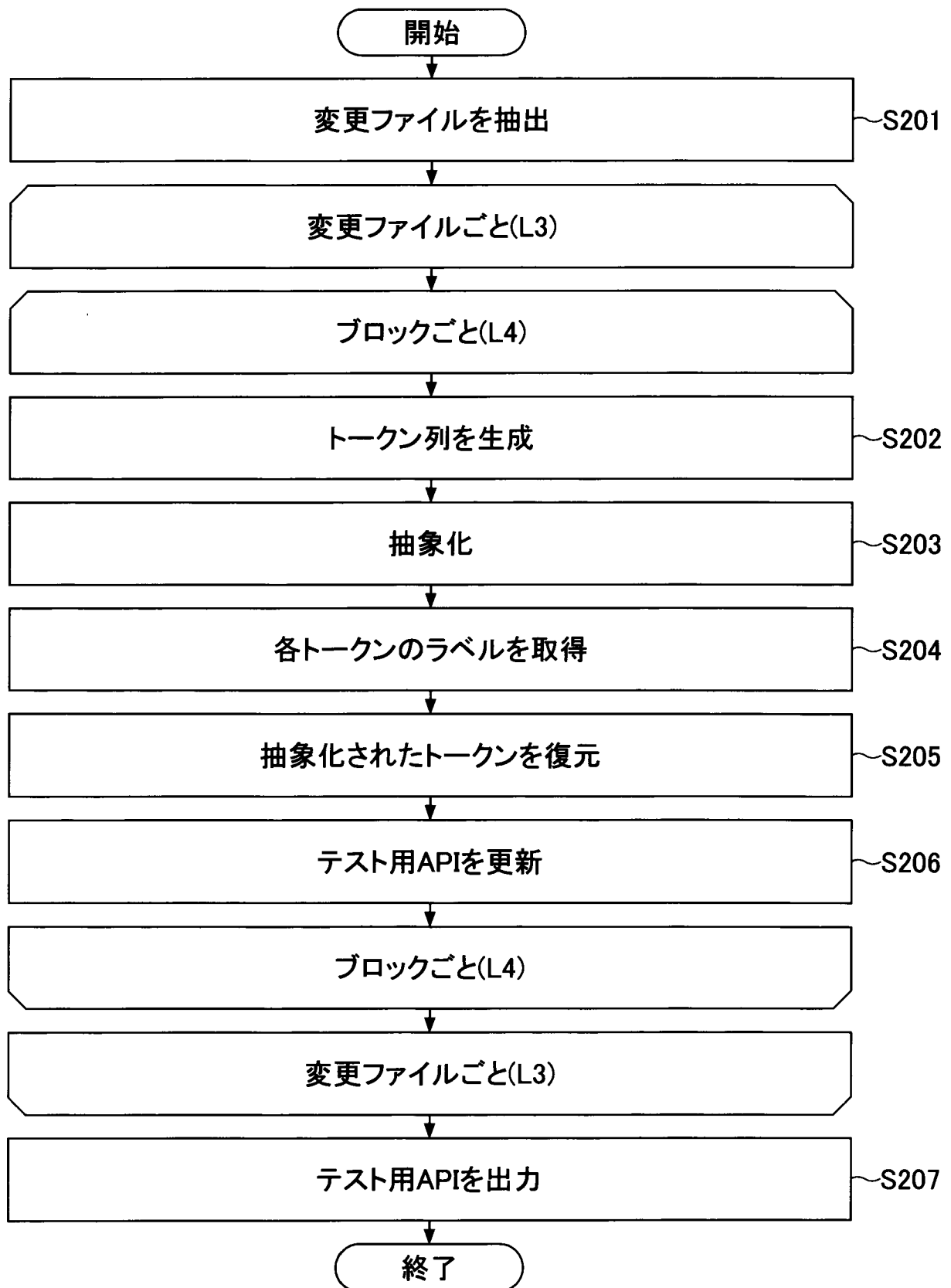
[図3]



[図4]



[図5]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2021/019661

A. CLASSIFICATION OF SUBJECT MATTER

G06F 8/70(2018.01)i; G06N 20/00(2019.01)i

FI: G06F8/70; G06N20/00 130

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)

G06F8/70; G06N20/00

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 1971-2021

Registered utility model specifications of Japan 1996-2021

Published registered utility model applications of Japan 1994-2021

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	JP 2020-91865 A (FUJITSU LTD) 11 June 2020 (2020-06-11) paragraphs [0026]-[0049], fig. 1	1, 4, 7
Y		2, 3, 5, 6
Y	JP 2019-28723 A (NIPPON TELEGRAPH & TELEPHONE) 21 February 2019 (2019-02-21) paragraph [0014]	2, 3, 5, 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
24 June 2021 (24.06.2021)

Date of mailing of the international search report
06 July 2021 (06.07.2021)

Name and mailing address of the ISA/
Japan Patent Office
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/JP2021/019661

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
JP 2020-91865 A	11 Jun. 2020	US 10579372 B1 column 5, line 9 to column 11, line 7	
JP 2019-28723 A	21 Feb. 2019	(Family: none)	

A. 発明の属する分野の分類（国際特許分類（IPC）） G06F 8/70(2018.01)i; G06N 20/00(2019.01)i FI: G06F8/70; G06N20/00 130														
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）） G06F8/70; G06N20/00														
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 <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922 - 1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971 - 2021年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996 - 2021年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994 - 2021年</td> </tr> </table>			日本国実用新案公報	1922 - 1996年	日本国公開実用新案公報	1971 - 2021年	日本国実用新案登録公報	1996 - 2021年	日本国登録実用新案公報	1994 - 2021年				
日本国実用新案公報	1922 - 1996年													
日本国公開実用新案公報	1971 - 2021年													
日本国実用新案登録公報	1996 - 2021年													
日本国登録実用新案公報	1994 - 2021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 2020-91865 A（富士通株式会社）11.06.2020（2020 - 06 - 11） [0026] - [0049], 図1	1, 4, 7												
Y		2, 3, 5, 6												
Y	JP 2019-28723 A（日本電信電話株式会社）21.02.2019（2019 - 02 - 21） [0014]	2, 3, 5, 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。														
<table border="0"> <tr> <td>* 引用文献のカテゴリー</td> <td>“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</td> </tr> <tr> <td>“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</td> <td>“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</td> </tr> <tr> <td>“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</td> <td>“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</td> </tr> <tr> <td>“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）</td> <td>“&” 同一パテントファミリー文献</td> </tr> <tr> <td>“O” 口頭による開示、使用、展示等に言及する文献</td> <td></td> </tr> <tr> <td>“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</td> <td></td> </tr> </table>			* 引用文献のカテゴリー	“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	“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	“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	“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	“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	“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）	“&” 同一パテントファミリー文献	“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	
* 引用文献のカテゴリー	“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													
“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	“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													
“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	“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													
“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）	“&” 同一パテントファミリー文献													
“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														
国際調査を完了した日 24.06.2021	国際調査報告の発送日 06.07.2021													
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） 中村 信也 5B 4058 電話番号 03-3581-1101 内線 3545													

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/JP2021/019661

引用文献			公表日	パテントファミリー文献	公表日
JP	2020-91865	A	11.06.2020	US 10579372 B1 コラム5第59行ーコラム11第 7行	
JP	2019-28723	A	21.02.2019	(ファミリーなし)	