

(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
2023年2月2日(02.02.2023)



(10) 国際公開番号
WO 2023/007791 A1

(51) 国際特許分類:

G06F 8/35 (2018.01) **G06F 40/211** (2020.01)
G06F 40/151 (2020.01)

(21) 国際出願番号: PCT/JP2022/008237

(22) 国際出願日: 2022年2月28日(28.02.2022)

(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 2021-123178 2021年7月28日(28.07.2021) JP

(71) 出願人: SOPPRA Digital
transformation 株
会社 (SOPPRA DIGITAL TRANSFORMATION

CORPORATION) [JP/JP]; 〒1000005 東京都千代田区丸の内2丁目5番2号 Tokyo (JP).

(72) 発明者: 白川 基光 (SHIRAKAWA Motomitsu);
〒1000005 東京都千代田区丸の内2丁目5番2
号 SOPPRA Digital transformation 株式会社内 Tokyo (JP).

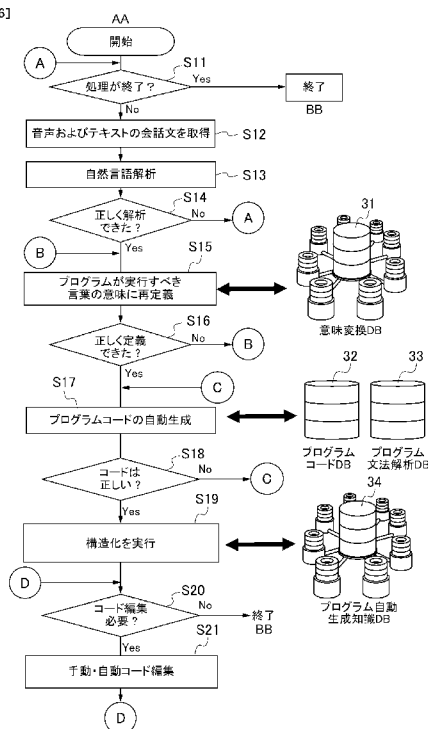
(74) 代理人: 安彦 元 (ABIKO Gen); 〒1020073 東京
都千代田区九段北1-5-10 九段ク
レストビル2階 Tokyo (JP).

(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,

(54) Title: PROGRAM CODE AUTOMATIC GENERATION SYSTEM

(54) 発明の名称: プログラムコード自動生成システム

[図6]



AA... START
31... MEANING CONVERSION DB
32... PROGRAM CODE DB
33... PROGRAM GRAMMAR ANALYSIS DB
34... PROGRAM AUTOMATIC GENERATION KNOWLEDGE DB
S11... HAS PROCESS BEEN FINISHED?
S12... ACQUIRE SOUND AND TEXT CONVERSATION SENTENCE
S13... NATURAL LANGUAGE ANALYSIS
S14... HAS ANALYSIS BEEN MADE CORRECTLY?
S15... REDEFINE MEANING OF WORD TO BE EXECUTED
BY PROGRAM
S16... HAS DEFINITION BEEN MADE CORRECTLY?
S17... AUTOMATICALLY GENERATE PROGRAM CODE
S18... IS CODE CORRECT?
S19... EXECUTE STRUCTURALIZATION
S20... IS IT NECESSARY TO EDIT CODE?
S21... EDIT CODE AUTOMATICALLY/MANUALLY
BB... END

(57) Abstract: [Problem] To automatically generate a short program code in which knowledge and experience of a skilled IT engineer is reflected and which is smooth and streamlined. [Solution] The present invention involves: acquiring text data from a document or a conversation sentence; performing natural language analysis of the acquired text data so as to extract contexts; referring to a program code database DB32 in which one or more program codes are stored in association with each of the contexts, a program grammar analysis database DB33 in which a grammar textbook of a program language

[続葉有]

HR, HU, ID, IL, IN, IR, IS, IT, JM, 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,
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))

is recorded, and a program automatic generation knowledge database DB34 in which a streamlined program is associated with a process operation to be realized; and generating one or more programs according to each of the extracted contexts.

(57) 要約：【課題】熟練のIT技術者の知識や経験が反映された、スムーズで効率化が図られた短いプログラムコードを自動生成する。【解決手段】文書又は会話文からテキストデータを取得し、取得したテキストデータを自然言語解析することにより、文脈を抽出し、各文脈に対して1以上のプログラムコードが関連付けて記憶されるプログラムコードデータベースDB32と、プログラム言語の文法書が記録されたプログラム文法解析データベースDB33と、実現すべき処理動作に対して効率化されたプログラムが関連付けられたプログラム自動生成知識データベースDB34とを参照し、抽出した文脈に応じた1以上のプログラムを生成する。

明 細 書

発明の名称：プログラムコード自動生成システム

技術分野

[0001] 本発明は、プログラム開発の仕組みを根本的に変えるプログラムコード自動生成システムに関するものである。

背景技術

[0002] 従来、プログラム開発を行う場合には、システムエンジニア（SE）やプログラマーを始めとするIT技術者が、仕様書やそれに準ずるドキュメントにも基づいて、様々なプログラム言語によりプログラムコードをコーディングしている。

[0003] 近年では、世界的に深刻なIT技術者不足に対応するため、ローコードあるいはノーコードプログラミングと称するプログラム動作を定義したテンプレートの組み合わせにより簡易な定型プログラムを作成したり、RPAの様な定型作業手順を設定し、業務を行うツール等を活用しプログラム開発工数を削減している。

[0004] しかしながら、このようなツールは何れもIT技術者が設定や構築を行わなければならない、また本格的な業務システムの構築や高度なプログラム開発は、このようなツールでは不可能である。このため、結局のところ、本格的なシステム構築やプログラム開発は、高度な技術をもった熟練のIT技術者に依存せざるを得ず、開発工数がなかなか削減できないのが現状であった。

[0005] このため、近年において、プログラムコードを極めて容易に、しかも人手を介することなく自動的に生成することが可能なプログラムコード自動生成システムが提案されている（例えば、特許文献1参照。）。この特許文献1の開示技術によれば、取得した音声に基づいてテキストデータから指示動作と、これに係り受けする格成分を抽出し、人工知能を利用して、最適な実行動作命令、実行動作対象を探索する。そして、この探索した実行動作命令、実行動作対象をプログラムコードの基本構文に代入することでプログラムコ

ードを生成する。

先行技術文献

特許文献

[0006] 特許文献1：特許第6 8 8 9 4 4 8 号公報

発明の概要

発明が解決しようとする課題

[0007] しかしながら、上述した特許文献1の開示技術では、プログラムを自動生成する上で、単にプログラムの基本構文に実行動作命令に代入しているにすぎないため、実際に複雑な業務処理のプログラムを自動生成させた場合に、できあがるプログラムコード全体が冗長なものとなってしまう場合が多い。プログラムコードが冗長になると、プログラムの読み出しや処理動作に時間を要してしまい、意図している業務処理を円滑に実行させることが難しくなる。このため、冗長なプログラムを自動生成させるのではなく、同じ業務処理であっても、まさしく熟練のIT技術者だけが起案できるような、スムーズで効率化が図られた、短いプログラムコードを自動生成するツールが従来より望まれていた。

[0008] そこで本発明は、上述した問題点に鑑みて案出されたものであり、その目的とするところは、業務処理をコンピュータのシステムにおいて自動的に実行していくために、その業務処理を実行するためのプログラムコードを自動生成する際において、熟練のIT技術者の知識や経験が反映された、スムーズで効率化が図られた短いプログラムコードを自動生成することが可能であり、プログラム開発の仕組みを根本的に変え、近い将来において起こりえる慢性的なSE技術者不足が社会に甚大な影響を与えることを回避できるプログラムコード自動生成システムを提供することにある。

課題を解決するための手段

[0009] 第1発明に係るプログラムコード自動生成システムは、文書又は会話文からテキストデータを取得するテキストデータ取得手段と、上記テキストデー

タ取得手段により取得されたテキストデータを自然言語解析することにより、文脈を抽出する文脈抽出手段と、各文脈に対して1以上のプログラムコードが関連付けて記憶されるプログラムコードデータベースと、プログラム言語の文法書が記録されたプログラム文法解析データベースとを参照し、上記文脈抽出手段により抽出された文脈に応じた1以上のプログラムコードを割り当てるコード割当手段と、プログラムコード又は当該プログラムコードにより実現すべき処理動作に対して、効率化されたプログラムの構造化パターンが関連付けられたプログラム自動生成知識データベースを参照し、上記コード割当手段により割り当てられたプログラムコードを効率化されたプログラムに書き換えるコード効率化手段とを備えることを特徴とする。

[0010] 第2発明に係るプログラムコード自動生成システムは、文書又は会話文からテキストデータを取得するテキストデータ取得手段と、上記テキストデータ取得手段により取得されたテキストデータを自然言語解析することにより、文脈を抽出する文脈抽出手段と、各文脈に対して1以上のプログラムコードが関連付けて記憶されるプログラムコードデータベースと、プログラム言語の文法書が記録されたプログラム文法解析データベースと、実現すべき処理動作に対して効率化されたプログラムが関連付けられたプログラム自動生成知識データベースとを参照し、上記文脈抽出手段により抽出された文脈に応じた1以上のプログラムを生成するコード割当手段とを備えることを特徴とする。

[0011] 第3発明に係るプログラムコード自動生成システムは、第1発明又は第2発明において、上記文脈抽出手段は、上記自然言語解析することにより、単語、指示動作、指示動作及びこれに係り受けする単語の何れかを抽出することにより、上記文脈を抽出することを特徴とする。

[0012] 第4発明に係るプログラムコード自動生成システムは、第3発明において、上記文脈抽出手段は、単語、指示動作、指示動作及びこれに係り受けする単語の何れかと、意味とが互いに関連付けて記憶される意味変換データベースを参照し、新たに抽出した上記単語、指示動作、指示動作及びこれに係り

受けする単語の何れかと関連する意味を探索することにより、上記文脈を抽出することを特徴とする。

[0013] 第5発明に係るプログラムコード自動生成システムは、第1発明又は第2発明において、上記コード割当手段は、上記プログラム文法解析データベースを参照し、生成すべきプログラムの文法を解析することにより、その正誤をチェックすることを特徴とする。

[0014] 第6発明に係るプログラムコード自動生成システムは、第1発明において、上記コード効率化手段は、プログラムコードに対して、効率化されたプログラムが関連付けられたプログラム自動生成知識データベースを参照し、上記コード割当手段により割り当てられたプログラムコードに関連する上記効率化されたプログラムを抽出し、これに書き換えることを特徴とする。

[0015] 第7発明に係るプログラムコード自動生成システムは、第1発明において、上記コード効率化手段は、プログラムコードにより実現すべき処理動作に対して、効率化前の上記プログラムコードから効率化されたプログラムへ変換するためのアルゴリズムが上記構造化パターンとして関連付けられたプログラム自動生成知識データベースを参照し、上記コード割当手段により割り当てられたプログラムコードにより実現すべき処理動作に関連付けられた上記構造化パターンとしてのアルゴリズムに基づいて、上記書き換えを行うことを特徴とする。

[0016] 第8発明に係るプログラムコード自動生成システムは、第3発明において、上記文脈抽出手段は、単語、指示動作、指示動作及びこれに係り受けする単語の何れかと、意味とを教師データとして機械学習されたモデルを用い、入力を単語、指示動作、指示動作及びこれに係り受けする単語の何れかとし、出力を意味とする意味変換データベースを参照することを特徴とする。

[0017] 第9発明に係るプログラムコード自動生成システムは、第1発明～第8発明の何れかにおいて、上記コード割当手段は、文脈とプログラムコードとを教師データとして機械学習されたモデルを用い、入力を文脈とし、出力をプログラムコードとするプログラムコードデータベースを参照することを特徴

とする。

[0018] 第10発明に係るプログラムコード自動生成システムは、第1発明において、上記コード効率化手段は、プログラムコード又は当該プログラムコードにより実現すべき処理動作と、効率化されたプログラムの構造化パターンとを教師データとして機械学習されたモデルを用い、入力をプログラムコード又は当該プログラムコードにより実現すべき処理動作とし、出力を上記構造化パターンとするプログラム自動生成知識データベースを参照することを特徴とする。

発明の効果

[0019] 上述した発明によれば、意図している業務処理をコンピュータ側のシステムにおいて自動的にプログラムコードを生成することができる。このため、極めて容易に、しかも熟練のIT技術者のような技術力がなくても、自動的にプログラムコードを生成することが可能となる。

[0020] これに加えて、効率化されたプログラムの構造化パターンが関連付けられたプログラム自動生成知識データベースを参照し、プログラムコードを効率化されたプログラムに書き換えることができる。その結果、熟練のIT技術者の知識や経験が反映された、スムーズで効率化が図られた短いプログラムコードを自動生成することが可能となる。これにより、プログラム開発の仕組みを根本的に変え、近い将来、慢性的なIT技術者不足が社会に甚大な影響を与えることを回避できる。

図面の簡単な説明

[0021] [図1]本発明を適用したプログラムコード自動生成システムの構成の一例を示す図である。

[図2]意味変換データベース(DB)に格納されているデータについて説明するための図である。

[図3]プログラムコードDBに格納されているデータについて説明するための図である。

[図4]プログラム自動生成知識DBに格納されているデータについて説明する

ための図である。

[図5]プログラム自動生成知識DBに格納されているデータについて説明するための他の図である。

[図6]本発明を適用したプログラムコード自動生成システムの動作例について説明するためのフローチャートである。

[図7]意味変換DBについて人工知能による学習データを利用する例について示す図である。

[図8]プログラムコードDBについて人工知能による学習データを利用する例について示す図である。

[図9]プログラム自動生成知識DBについて人工知能による学習データを利用する例について示す図である。

発明を実施するための形態

[0022] 以下、本発明を適用したプログラムコード自動生成システムについて、図面を参照しながら詳細に説明をする。

[0023] (プログラムコード自動生成システム100)

図1を参照して、本実施形態におけるプログラムコード自動生成システム100の構成の一例について説明する。図1は、本実施形態におけるプログラムコード自動生成システム100の全体の構成を示す模式図である。

[0024] プログラムコード自動生成システム100は、単なる業務補助ではなくプログラム開発の仕組みを根本的に変え、近い将来、慢性的なSE技術者不足が社会に甚大な影響を与えることを回避できるものである。

[0025] すなわち、プログラムコード自動生成システム100は、業務を実行するためのプログラムコードを自動生成することで、家庭や企業、官公庁内における各業務処理（例えば、メールの配信、作業者の進捗状況の収集、ファイル管理、ファイル転送等）の様々なデータ処理をコンピュータ上で自動的に行うことができる。プログラムコード自動生成システム100は、特にこのプログラムコードの自動生成を、音声又はテキストにより設定することができ、システム管理者等のような専門的知識を有しないユーザ（例えばプログ

ラムコード自動生成システム１００を利用して業務を管理する利用者等）においても、様々な業務処理をコンピュータに自動的に行わせるためのプログラムコードの自動生成を容易に実現することが可能となる。

[0026] プログラムコード自動生成システム１００は、例えば図１に示すように、プログラムコード自動生成装置１を備え、ユーザがプログラムコード自動生成装置１を利用してもよい。プログラムコード自動生成システム１００は、例えば通信網４を介してプログラムコード自動生成装置１と接続された端末２を備え、ユーザが端末２を介してプログラムコード自動生成装置１を利用してもよい。プログラムコード自動生成システム１００は、例えば通信網４を介してプログラムコード自動生成装置１と接続されたサーバ３を備え、ユーザがプログラムコード自動生成装置１又は端末２を介してサーバ３との各種情報の送受信により、各手段を実現してもよい。

[0027] <プログラムコード自動生成装置１>

プログラムコード自動生成装置１として、例えばサーバ、パーソナルコンピュータ、スマートフォン、タブレット端末、ウェアラブル端末等の公知の電子機器が用いられる。なお、本実施形態におけるプログラムコード自動生成システム１００において、プログラムコード自動生成装置１は、クラウドサーバの中枢となる。

[0028] プログラムコード自動生成装置１は、サーバ、パーソナルコンピュータ（ＰＣ）、スマートフォン、タブレット端末、ウェアラブル端末等で構成される。プログラムコード自動生成装置１の各機能は、ＣＰＵが、ＲＡＭを作業領域として、保存部等に記憶されたプログラムを実行することにより実現される。また、各機能の一部は、人工知能により制御されてもよい。ここで、いう人工知能は、いかなる周知の人工知能技術に基づくものであってもよい。

[0029] プログラムコード自動生成装置１は、文書又は会話文からテキストデータを取得する。プログラムコード自動生成装置１は、ユーザから会話文が音声により入力された場合、取得部は、公知の音声認識技術（例えば音声認識技

術)を用いて、マイクロフォンを介して受け付けた音声からテキストデータを生成する。なお、音声認識技術は、例えば通信網4を介して、クラウド型の音声認識技術を用いてもよい。プログラムコード自動生成装置1は、文書からテキストデータを取得する場合、紙媒体や電子媒体に記載されている文書中の文字列からテキストデータを取得する。かかる場合には、紙媒体から読み取った文字列をOCR技術で文字認識してテキストデータ化してもよいし、電子媒体からなる文書に含まれる文字列をそのままテキストデータとして取得してもよい。かかる場合において、このプログラムコード自動生成装置1により取得されるテキストデータは、プログラムコード自動生成装置1内にあるデータベース内にあるデータやサーバ3内にあるメール文章、ドキュメントファイル等から取得するようにしてもよい。

[0030] なお、このテキストデータの取得は、プログラムコード自動生成装置1自身が行う場合以外に、端末2側において行うようにしてもよい。かかる場合には、端末2においてマイクロフォンを介して受け付けた音声からテキストデータを取得するための音声認識機能が実装され、或いは紙媒体から文字列を文字認識するためのOCR機能が実装されている必要がある。

[0031] <端末2>

端末2として、例えばサーバ、PC、スマートフォン、タブレット端末、ウェアラブル端末等の公知の電子機器が用いられる。端末2は、例えばプログラムコード自動生成装置1と同様の構成及び機能の少なくとも一部を備えてもよい。端末2は、例えば複数備えてもよく、各端末2がそれぞれ通信網4を介してプログラムコード自動生成装置1と接続されてもよい。

[0032] <通信網4>

通信網4は、プログラムコード自動生成装置1が通信回路を介して接続されるインターネット網等である。通信網4は、いわゆる光ファイバ通信網で構成されてもよい。また、通信網4は、有線通信網のほか、無線通信網等の公知の通信網で実現されてもよい。

[0033] <サーバ3>

サーバ3には、例えば各種情報が記憶される。サーバ3には、例えば通信網4を介してプログラムコード自動生成装置1等から送られてきた各種情報が蓄積される。サーバ3には、例えばプログラムコード自動生成装置1の保存部と同様の情報が記憶され、通信網4を介してプログラムコード自動生成装置1等と各種情報の送受信が行われてもよい。即ち、プログラムコード自動生成システム100では、プログラムコード自動生成装置1又はプログラムコード自動生成装置1の保存部の代わりにサーバ3を用いてもよい。

[0034] また本発明においては、サーバ3において意味変換データベース(DB)31、プログラムコードデータベース(DB)32、プログラム文法解析データベース(DB)33、プログラム自動生成知識データベース(DB)34が構築されている。

[0035] 意味変換データベース(DB)31は、図2に示すように、単語、指示動作、指示動作及びこれに係り受けする単語に対して、意味が関連付けて記憶される。

[0036] 単語とは、テキストデータに含まれる名詞、名詞句、形容詞、副詞、その他動詞を除くあらゆる文の構成要素を構成する単語が含まれる。この単語は、それぞれ指示動作に対して係り受けすることで一つの意図する動作を形成することとなる。

[0037] 指示動作は、テキストデータに含まれる動詞である。この動詞に単語が係り受けすることで、一つの意図する動作を形成することとなる。

[0038] 例えば、図2(a)に示すように、「画面に出力する」であれば、単語が「画面」であり、これが係り受ける指示動作が「出力する」である。同じく「ファイルを登録する」であれば、単語が「ファイル」であり、これが係り受ける指示動作が「登録する」である。

[0039] このような単語と指示動作に対して、それぞれ意味が紐付けられる。ここでいう意味とは、実際に単語と指示動作を通じて特定される意味概念を示すものである。意味は、単なる日常生活において使用される会話の意図を示す意味ではなく、プログラムコードを決定する上で相応しい意味を示すもので

ある。

[0040] 図2(a)の例では、「画面に出力する」(単語:「画面」、指示動作:「出力する」)であれば、意味として「データを見せる」が紐付けられている。また、「ファイルを登録する」(単語:「ファイル」、指示動作:「登録する」)であれば、意味として「書き込む」が紐付けられている。

[0041] なお、この意味変換DB31では、単語とこれに係り受ける指示動作が一つのセットになり、単語と指示動作の組み合わせが意味に紐付けられている場合を例に挙げて説明をしているが、これに限定されるものではなく。単語、又は指示動作の何れか一方のみが少なくとも意味に対して紐付けられていればよい。

[0042] このような指示動作、単語、意味は、何れもテキストデータの文脈の構成要素となる。即ち、指示動作、単語、意味は、テキストデータが何を示しているのか、その文脈につながるものと考えることができる。文脈は、指示動作、単語、意味の3つにより初めて形成されるものに限定されず、指示動作、単語、意味の少なくとも1つにより形成されるものであってもよい。

[0043] 図2(b)の例では、「データを移す」というテキストデータにおける文脈を示している。かかる場合には、単語が「データ」、指示動作が「出力する」となるが、「データを移す」を構成する単語や指示動作からなる一つの組み合わせからは、「データを転送する」、「(データを)コピーする」、「(データを)移動する」等、様々な意味概念を想起することができる。かかる場合には、意味変換DB31には、想起可能な全ての意味を、この指示動作や単語に対して紐付けておく。

[0044] 但し、この図2(b)の例において、指示動作が「移す」ではなく、「転送する」であれば、「データを転送する」という意味に絞り込むことができるし、指示動作又は単語に「バックアップ」という文言が含まれているのであれば、「コピーする」という意味に絞り込むことができる。このような文言が含まれている単語や指示動作には、それぞれ紐付ける意味を絞り込むことができる。

[0045] 図2(c)の例も同様に「出力データを見せて」というテキストデータにおける文脈を示しているが、単語や指示動作からなる一つの組み合わせに対して様々な意味概念を想起することができる。かかる場合には、意味変換DB31には、想起可能な全ての意味を、この指示動作や単語に対して紐付けておく。但し、この図2(c)の例において、「通信回線」や「画面」、「プリンタ」等のように、意味概念を絞り込むことができるような単語が含まれている場合には、そのような文言が含まれている単語や指示動作には、それぞれ紐付ける意味を絞り込むことができる。

[0046] 特に指示動作と単語の組み合わせではなく、指示動作のみ、或いは単語のみの場合には、意味概念を絞り込むことができない場合が多く、想起可能な意味概念は多岐に亘る場合があるが、かかる場合も同様に、その多岐に亘る意味概念を全て紐付けることとなる。

[0047] プログラムコードDB32は、図3に示すように、各文脈に対してプログラムコードが紐付けられているテンプレートが記憶される。つまり、文脈に含まれている、意図する処理動作に応じたプログラムコードが紐付けられている。単なる指示動作又は単語のみで文脈を構成する場合には、図3(a)に示すように、その指示動作又は単語に対してそれぞれプログラムコードが紐付けられている。また、文脈が指示動作、単語、又は指示動作とこれに係り受けする少なくとも単語の組み合わせに加え、更に意味までが割り当てられている場合には、その意味に対してそれぞれプログラムコードが紐付けられている。つまり、このプログラムコードDB32は、文脈が特定できれば、その文脈に含められた、意図する処理動作を実行するために必要なプログラムコードを特定することができるデータベースといえる。

[0048] このプログラムコードDB32においても、文脈が全く同じであっても、また文脈を構成する指示動作、単語、又は指示動作とこれに係り受けする少なくとも単語、意味が全く同じであっても、意図する処理動作を実行するために必要なプログラムコードを複数に亘り想起できる場合がある。かかる場合には、その一の文脈に対して複数のプログラムコードを紐付けるようにし

てもよい。

[0049] 同様に、一のプログラムコードに対して複数の文脈が成り立つ場合もある。かかる場合には、その一のプログラムコードに対して複数の文脈を紐付けるようにしてもよい。

[0050] このようなプログラムコードDB32は、プログラム言語毎に、文脈とプログラムコードを紐付けたテンプレートを予め構築しておく。また、プログラムコードDB32は、プログラム言語毎の全ての命令語（プログラムコード）が、文脈（指示動作や単語、意味等）に対応付けられた状態となっている。その結果、全てのプログラム言語の各プログラムコードが、文脈と対応付けられていることから、逆に文脈に対応するプログラムコードを全てのプログラム言語につき、探索することが可能となる。

[0051] プログラム文法解析DB33は、プログラムコード自動生成装置1において、プログラムコードDB32を参照しつつ生成したプログラムコードが文法的に正しいかを確認するためのデータベースである。プログラム文法解析DB33は、生成したプログラムコードのスペルや記述方法、命令語の配列、必要な記号やマークが挿入されているか否か、引数が正しく挿入されているか否か等、プログラムを正しく動作させる上で必要な情報が格納されている。このプログラム文法解析DB33は、例えば各プログラム言語の文法書の電子データが格納されていてもよい。プログラム言語の文法書には、通常、プログラムコードの記述方法が中心に説明がなされているが、プログラムコードの正しい文法が記述されていることから、これを参照することで、プログラムコード自動生成装置1において生成したプログラムコードが文法的に正しいかを確認することができる。また、プログラム文法解析DB33には、個別のプログラムコード(命令語)に対応した文法の正誤判断が可能なルール設定も記録されていてもよい。かかる場合には、例えばスペル、記載順序、パラメーターの内容等の正誤判断が可能なルールが予め記録されていてもよく、またそのルールに基づいて正誤判断をするためのアルゴリズムが記録されていてもよい。なお、プログラム文法解析DB33では、コンパイラ

やアセンブラにて行う通常の文法チェックに必要なルールや規則、正しい文法に関する情報が予め格納されていてもよい。

[0052] プログラム自動生成知識DB34は、プログラムコード又は当該プログラムコードにより実現すべき処理動作に対して、効率化されたプログラムの構造化パターンが関連付けられたデータベースである。このプログラム自動生成知識DB34は、実現すべき処理動作をプログラムコード化する上で高度な技術をもった熟練のIT技術者であればどのようなプログラムを生成するか、或いはどのようにプログラムコードを構造化するか、についての知識に関する情報が記録されている。プログラム自動生成知識DB34では、このような熟練のIT技術者による構造化パターンのルール(定義)が設定されている。

[0053] 例えば、画面中に繰り返し複数行の表を表示をする場合には、熟練のIT技術者によれば、その繰り返し回数や位置を決めループ処理をすることで、プログラムコードを構造化し、より効率的で短いプログラムコードを起案する場合が多い。このような熟練のIT技術者の経験やテクニックのサンプルをルール(コード)化し、プログラム自動生成知識DB34において登録する事で構造化プログラムのコーディングが可能となる。このプログラム自動生成知識DB34には、基本的にテンプレート化されたいくつもの効率化されたプログラムの構造化パターンが登録されている。ここでいう効率化されたプログラムの構造化パターンは、上述した熟練のIT技術者の経験や知識が反映されている。そして、プログラムコード自動生成装置1において生成したプログラムコード又は当該プログラムコードにより実現すべき処理動作に対応した効率化されたプログラムの構造化パターンが選択可能とされている。かかる場合において、システムで多くのAPIを予め作り込んでおき、処理毎にそのAPIを呼び出すことで上述したプログラムの構造化パターンを選択するようにしてもよい。

[0054] 図4は、プログラム自動生成知識DB34において、画面上に商品毎の売り上げデータ等を表示する場合における、通常のプログラムコードと、効率

化されたプログラムコードが格納されている例を示している。

- [0055] 実現すべき処理動作P 1 が、各商品名について単価、実際に売り上げた数量、売上、更に売上合計を表形式で表示することである場合、一般的なプログラムコードP 2 に示すように、個々の商品毎に、単価、数量売上がそれぞれ集計処理し、これを表示することをそれぞれ実行命令として定義していく。その結果、プログラムの行数が増加してしまい、全体的に長く冗長となってしまう。
- [0056] これに対して、効率化されたプログラムの構造化パターンP 3 では、売上の集計処理や表示処理を個々の商品毎に行うのではなく一括して行い、表示処理についても共通ロジックに基づいて構造化したものを使用し、表示用データについては、最後尾においてitem_listにおいてリスト化しておくものが例示されている。この構造化パターンP 3 に示されるプログラムを実行することで、プログラムコードP 2 と同様に、実現すべき処理動作P 1 としての商品毎の売り上げデータ等を画面上に表示することができる。但し、この構造化パターンP 3 は、熟練のIT技術者の知識や経験が反映されたものであることから、効率的され、しかも短い行数で、同様の実行すべき処理動作P 1 を実現することができる。
- [0057] プログラム自動生成知識DB 3 4 では、このような実現すべき処理動作P 1 及びプログラムコードP 2 に対する、構造化パターンP 3 がそれぞれ記録されている。そして、実現すべき処理動作P 1 が多岐にわたる場合には、その多岐にわたる実現すべき処理動作P 1 毎にそのプログラムコードP 2、更にこれに対応する構造化パターンP 3 が互いに紐付けられて記録されている。
- [0058] 図4 に示す例では、プログラム自動生成知識DB 3 4 において、実現すべき処理動作P 1 と、これに対応するプログラムコードP 2 が紐付けられ、更にこの実現すべき処理動作P 1 とプログラムコードP 2 に対して、構造化パターンP 3 が互いに紐付けられる場合を示しているが、これに限定されるものではない。例えば実現すべき処理動作P 1 があれば、これに対して直接的

に構造化パターンP 3が紐付けられ、一般的なプログラムコードP 2は省略してもよい。また、一般的なプログラムコードP 2と構造化パターンP 3のみが互いに紐付けられ、実現すべき処理動作P 1は特段これらに紐付けられていなくてもよい。

[0059] また、プログラム自動生成知識DB 34では、プログラムコードにより実現すべき処理動作P 1に対して、一般的なプログラムコードP 2から効率化されたプログラムP 3へ変換するためのアルゴリズムが構造化パターンとして関連付けられるものであってもよい。かかる場合には、この実現すべき処理動作P 1を特定することができれば、これに関連付けたアルゴリズムを読み出すことにより、効率化前の一般的なプログラムコードP 2から効率化されたプログラムP 3へ変換することが可能となる。このようなアルゴリズムは、一般的なプログラムコードP 2から効率化されたプログラムP 3へ遷移させる上での様々な法則や習性、知識やノウハウ、パターン等、熟練のIT技術者から取得してアルゴリズム化したものであってもよい。

[0060] また、プログラム自動生成知識DB 34では、実現すべき処理動作P 1から、効率化されたプログラムP 3へ遷移させる上での様々な法則や習性、知識やノウハウ、パターン等の概念が記録されるものであってもよい。例えば、ある実現すべき処理動作P 1から、効率化されたプログラムP 3を作る上で、効率化されたプログラムP 3自身が格納されることは必須ではなく、例えば熟練のIT技術者であればどのような手順からなるプログラムを作成するか、どのようなコンセプトでプログラムの起案するか等、概念的な情報が格納されていてもよい。

[0061] 図5は、実現すべき処理動作P 1が各種処理を行った場合にエラーが発生した場合には、それに対応した処理を実行する例における、効率化前の一般的なプログラムコードP 2と、効率化されたプログラムP 3を示したものである。一般的なプログラムコードP 2では、プログラム中において、いくつもの箇所でエラー処理のコードを記述していることから、その分において全体的に長く冗長化したものとなっている。

- [0062] これに対して、効率化されたプログラムP 3は、共通ロジックを前段で構造化し、最後尾において出力データのみを記述しているが、これにより、効率化が図られ、全体的に短くまとめることができる。
- [0063] この図5の例においても同様に実現すべき処理動作P 1及びプログラムコードP 2に対する、構造化パターンP 3がそれぞれ記録されていてもよい。また実現すべき処理動作P 1があれば、これに対して直接的に構造化パターンP 3が紐付けられていてもよいし、一般的なプログラムコードP 2と構造化パターンP 3のみが互いに紐付けられていてもよい。また、実現すべき処理動作P 1に対して、一般的なプログラムコードP 2から効率化されたプログラムP 3へ変換するためのアルゴリズムが構造化パターンとして関連付けられるものであってもよい。
- [0064] このように全ての実現すべき処理動作P 1又はプログラムコードP 2毎に、それぞれ構造化パターンP 3が、このプログラム自動生成知識DB 34に蓄積されている。このとき、一の処理動作P 1又はプログラムコードP 2に対して複数種の構造化パターンP 3が関連付けられていてもよい。また一の構造化パターンP 3に対して複数種の処理動作P 1又はプログラムコードP 2が関連付けられていてもよい。
- [0065] なお、このプログラム自動生成知識DB 34では、定型的なプログラムコードのパターンが関連付けられている場合のみならず、非定型の何らかのパターンが関連付けされているものであってもよい。
- [0066] 次に本発明を適用したプログラムコード自動生成システム100により、プログラムコードを実際に自動生成する動作について詳細に説明をする。
- [0067] 図6に示すように処理動作を開始させた後、先ずステップS 11に示すように、処理が終了したか否かを確認をする。この確認は、プログラムコード自動生成装置1や端末2を介して直接ユーザに処理を終了させるか否かを問い合わせるようにしてもよいし、プログラムコード自動生成システム100側において終了したか否かを自動判別するようにしてもよい。
- [0068] 次にステップS 12に移行し、文書又は会話文からテキストデータを取得

する。具体的には、音声データとして端末2又はプログラムコード自動生成装置1において会話文が取得された場合には、これを公知の音声認識技術（例えば音素認識技術）を用いテキストデータを生成する。また、電子データ化されたテキストデータをプログラムコード自動生成装置1において取得した場合には、これをそのまま利用することになる。電子データ化されたテキストデータは、例えばメール文や社内又は社外において活用している情報掲示板や各種サイトの書き込みをそのまま取り込むようにしてもよい。

[0069] 端末2又はプログラムコード自動生成装置1は、ユーザの入力するタイミング毎にテキストデータを取得するほか、例えば一定期間に蓄積された複数のテキストデータを一度に取得してもよい。なお、端末2又はプログラムコード自動生成装置1を介して入力された非テキスト形式のデータから、テキストデータを生成する方法は、公知技術を用いることができる。このようにして取得されたテキストデータは、プログラムコード自動生成装置1において一時的に記憶される。

[0070] 次にステップS13に移行し、S12において取得され、プログラムコード自動生成装置1において一時的に記憶されているテキストデータを読み出し、これを自然言語解析することにより文脈を抽出する。この自然言語解析は、主としてプログラムコード自動生成装置1が行う。この自然言語解析では、形態素解析や構文解析を始めとして周知のいかなる自然言語解析技術を利用するようにしてもよい。

[0071] このステップS13における自然言語解析の結果、例えばテキストデータ「データを移す」については、「データ」からなる単語と、「移す」からなる指示動作に分解することができる。またテキスト「出力データを見せて」は、「出力データ」という単語と、「見せて」という指示動作に分解することができる。つまりテキストデータを指示動作からなる動詞に係り受けする単語の組み合わせを通じて文脈を抽出することができる。このステップS13において指示動作とこれに係り受けする単語からなる組み合わせを文脈として抽出する場合に限定されるものではなく、指示動作のみ、又は単語のみ

を文脈を抽出するようにしてもよい。特に「見せて」や、「ファイルA」等のようにテキストデータが動詞のみ、又は単語のみで構成される場合には、指示動作のみ、又は単語のみを抽出することとなる。

[0072] このステップS 1 3において自然言語解析を行った結果、ステップS 1 4へ移行し、正しく解析できた場合にはステップS 1 5へ移行する。これに対して、正しく自然言語解析できなかった場合には、ステップS 1 1へ戻る。

[0073] ステップS 1 5へ移行した場合、意味変換DB 3 1を参照し、ステップS 1 3において抽出した文脈からプログラムが実行すべき言葉の意味に再定義する。

[0074] 実際に文脈を構成する単語や動詞には、日常使用される口語調の文言が含まれている場合があるが、これが、プログラムコードを決定するための意味とは乖離している場合がある。このため、このステップS 1 5において、テキストデータに含まれている文脈をプログラムコードを決定する上で好適な意味に変換する。

[0075] 意味変換は、意味変換DB 3 1を参照し、例えば図2 (a) に示すように、ステップS 1 3において抽出された指示動作が「出力する」であり、単語が「ファイル」であれば、意味として「読み出す」に変換する。同様に指示動作が「照会する」であり、単語が「画面」であれば、意味として「データを見せる」に変換する。また図2 (b)、(c) に示すように一の指示動作と単語に対して複数の意味が紐付けられている場合には、これら紐付けられている全ての意味を探索するようにしてもよいし、これらのうち一部の意味のみを探索するようにしてもよい。また、ステップS 1 3において抽出されたものが指示動作のみ、又は単語のみである場合も同様に意味変換DB 3 1を参照し、指示動作又は単語に紐付けられた意味を抽出することとなる。ステップS 1 5では、このようにして抽出した意味を、テキストデータの文脈として定義し、ステップS 1 6へ移行する。

[0076] なお、このステップS 1 5における、プログラムが実行すべき言葉の意味に再定義する処理動作は必須ではなく、省略するようにしてもよい。かかる

場合には、ステップS 1 3において抽出した、単語、指示動作、指示動作及びこれに係り受けする単語を、テキストデータの文脈として定義し、ステップS 1 6へ移行する。

[0077] ステップS 1 6へ移行した場合には、ステップS 1 5における意味変換を通じて、テキストデータの文脈を正しく定義できたか否かを判別する。その結果、テキストデータの文脈を正しく定義できた場合には、ステップS 1 7へ移行する。これに対してテキストデータの文脈を正しく定義できなかった場合には、ステップS 1 5へと戻る。なお、ステップS 1 5自体を省略した場合、或いはステップS 1 5においてプログラムが実行すべき言葉の意味に再定義する処理動作を行わなかった場合には、このステップS 1 6を省略するようにしてもよい。

[0078] ステップS 1 7に移行した場合には、ステップS 1 5において定義されたテキストデータの文脈に基づいてプログラムコードを自動生成する。

[0079] かかる場合には、図3に示すプログラムコードDB 3 2を参照する。プログラムコードDB 3 2には、文脈（単語、指示動作、指示動作及びこれに係り受けする単語、意味）に対応する各プログラムコードが関連付けられて記憶されている。このため、ステップS 1 7では、プログラムコードDB 3 2を参照し、ステップS 1 5において定義されたテキストデータの文脈に対応するプログラムコードを特定する。このとき、特定したプログラムコードそのものを自動生成すべきプログラムコードとしてもよいが、文脈に応じて、いくつかのプログラムコードを組み合わせ、或いはプログラムコード内にある変数や記号、引数等を適宜流用してもよい。この組み合わせや流用を行う場合においても同様に、プログラムコードDB 3 2を参照する。

[0080] 例えば、図3（a）に示すように、ステップS 1 5において定義されたテキストデータの文脈が「送信」を含むものであれば、これに対応する命令語として「vary response=http.post」等をプログラムコードDB 3 2を参照して抽出し、これに基づいてプログラムコードを自動生成する。ステップS 1 5において定義されたテキストデータの文脈が「画面」を含むものであれ

ば、これに対応する命令語として「text xxxx」等をプログラムコードDB 32を参照して割り当て、これに基づいてプログラムコードを自動生成する。同様にステップS 15において定義されたテキストデータの文脈が「ファイル」を含むものであれば、これに対応する命令語として「import xxxx file.write…」等をプログラムコードDB 32を参照して割り当て、これに基づいてプログラムコードを自動生成する。

[0081] なお、図3(b)、(c)に示すように、一の指示動作や単語から複数の意味が紐付けられている場合もあることから、テキストデータにおいてこのような指示動作や単語で構成されている場合には、これに対応した複数の意味毎に関連付けられたプログラムコードを割り当てることで自動生成するようにしてもよい。かかる場合には、一のテキストデータに対して複数のプログラムコードが自動的に生成されることとなる。

[0082] 例えば、テキストデータが「入力データを画面とUSBに出力して」であり、自然言語解析の結果、指示動作が「出力して」であり、単語が「入力データ」、「画面」、「USB」である場合、図3に示すプログラムコードDB 32を参照し、“入力データの取り込み(Read)”、“画面装置への表示(print xxxx)”、“USBメモリーへのデータ保存(Disk_write)”等の意味に「出力」という語句の意味を解析しプログラムコードを決定し生成することとなる。

[0083] なお、本発明においては、指示動作や単語に対する意味を文脈として定義することは必須ではないことから、仮に意味が文脈として定義されていない場合には、指示動作又は単語に直接紐付けられているプログラムコードをプログラムコードDB 32から読み出し、プログラムコードを自動生成するようにしてもよい。

[0084] 次に、このステップS 17においてはプログラム文法解析DB 33を参照し、自動生成したプログラムコードの文法の正誤をチェックする。上述したようにプログラム文法解析DB 33には、各プログラム言語の文法書の電子データが格納されていてもよいことから、これを参照することで、自動生成

したプログラムコードの文法の正誤をチェックすることができる。実際には、この自動生成したプログラムコードと、文法書の電子データとを比較し、文法書に記載された語法に沿って記述されているのであれば、文法的に正しいものと判断し、文法書の記載から逸脱した形で記述されているのであれば、文法的に誤りであるものと判断することができる。

[0085] 例えば、自動生成したプログラムコードにおいて命令語が連続しているとき、プログラム文法解析DB33に記憶されている文法書の内容において命令語の語法が記述されている箇所を参照する。その結果、文法書には、正しい例として命令語が立て続けに連続するものが記述されていない場合には、文法的に誤りであると判断することができる。また自動生成したプログラムコードにおいて「print」の後に「” ”」が入っていない場合、同じくプログラム文法解析DB33に記憶されている文法書の内容において命令語の語法が記述されている箇所を参照する。その結果、文法書には、正しい例として「print」の後に実際に表示すべき情報を「” ”」で囲む例が記述されていない場合には、文法的に誤りであると判断することができる。

[0086] なお、プログラム文法解析DB33において、個別のプログラムコード(命令語)に対応した文法の正誤判断が可能なルール設定が記録されている場合には、そのルール設定に基づいて正誤判断を行うようにしてもよい。またそのルールに基づいて正誤判断をするためのアルゴリズムが記録されている場合には、そのアルゴリズムに基づいて判断を行うようにしてもよい。

[0087] また、このステップS17では、コンパイラやアセンブラにて行う通常の文法チェックを行うようにしてもよい。かかる場合には、プログラム文法解析DB33において格納されている、コンパイラやアセンブラにて行う通常の文法チェックに必要なルールや規則、正しい文法に関する情報を参照することとなる。

[0088] このプログラム文法解析DB33において正誤をチェックした結果、プログラムコードが文法的に誤っているものと判断された場合には、このステップS17において修正を加えるようにしてもよい。この修正を行う場合にお

いても同様にプログラム文法解析DB33を参照し、正しい文法の記載を引用し、これに基づいてプログラムコードを修正するようにしてもよい。

[0089] なお、このプログラム文法解析DB33を参照することによる文法の正誤のチェックは、生成すべきプログラムの文法を解析することにより行うようにしてもよい。ここでいう生成すべきプログラムとは、プログラムコードDB32に基づいて生成したプログラムコードも含まれるが、これに限定されるものではなく、後述するステップS19において、プログラム自動生成知識DB34を参照して書き換えられた、効率化されたプログラムも含まれる。即ち、プログラム文法解析DB33を参照することによる文法の正誤のチェックは、プログラムコードDB32に基づいて生成したプログラムコード、後述するステップS19において、プログラム自動生成知識DB34を参照して書き換えられた、効率化されたプログラムの何れもが対象になりえる。

[0090] また、このステップS17におけるプログラム文法解析DB33の活用方法としては、上述のような自動生成したプログラムコードの文法の正誤チェックは必須ではない。つまり、プログラムコードDB32に基づいてプログラムを生成する際において、少なくともこのプログラム文法解析DB33を参照していれば、文法の正誤チェックは省略するようにしてもよい。

[0091] かかる場合には、プログラムコードDB32に基づいてプログラムを生成する過程において、プログラム言語の文法書が格納されたプログラム文法解析DB33を参照する。これにより、現在作成中のプログラムにおいて、正しいプログラム文法を反映させることが可能となる。

[0092] ステップS17においては、このプログラム文法解析DB33によるプログラム文法の正誤判断は必須ではなく、省略するようにしてもよい。

[0093] このようなステップS17における各処理が終了した場合には、ステップS18へ移行する。ステップS18では、プログラムコードが正しいか否かを判断し、正しい場合には、ステップS19に、また正しくない場合には、再度ステップS17に戻る。

- [0094] ステップS 1 9に移行した場合には、プログラム自動生成知識DB 3 4を参照し、ステップS 1 7において文法的にチェックされたプログラムコードを、効率化されたプログラムに書き換える。
- [0095] 上述したように、プログラム自動生成知識DB 3 4には、実現すべき処理動作P 1及びプログラムコードP 2に対する構造化パターンP 3、実現すべき処理動作P 1に対する構造化パターンP 3、プログラムコードP 2に対する構造化パターンP 3、実現すべき処理動作P 1に対する、一般的なプログラムコードP 2から効率化されたプログラムP 3へ変換するためのアルゴリズムが記憶されている。
- [0096] つまり、ステップS 1 9に到達した段階で、実現すべき処理動作P 1、プログラムコードP 2の何れかが特定されていれば、この図4に示すようなプログラム自動生成知識DB 3 4を参照することで、構造化パターンP 3を探索することができる状態となっている。
- [0097] ステップS 1 9に到達した時点において、ステップS 1 7にて自動生成したプログラムコードが存在している。このため、ステップS 1 7にて自動生成したプログラムコードがあれば、これに最も類似しているプログラムコードP 2に関連付けられた構造化パターンP 3をこのプログラム自動生成知識DB 3 4から読み出すことができる。構造化パターンP 3自体が、効率化されたプログラムコードで構成されているのであれば、これ自体を新たに書き換えるべきプログラムとすることができる。また、プログラム自動生成知識DB 3 4から読み出した構造化パターンP 3が、変数、式、定義、引数等、一部がステップS 1 7にて自動生成したプログラムコードと相違する場合には、その構造化パターンP 3に、その相違する変数や式等を修正することにより、新たに書き換えるべきプログラムを生成するようにしてもよい。つまり、ステップS 1 9では、ステップS 1 7にて自動生成したプログラムコードに対して、これに最も類似しているプログラムコードP 2に関連付けられた構造化パターンP 3に基づいて修正することでプログラムの書き換えを行うものであってもよい。

- [0098] また、このステップS 1 9におけるプログラムの書き換えは、ステップS 1 7にて自動生成したプログラムコードに最も類似しているプログラムコードP 2に関連付けられた構造化パターンP 3に基づいて行う場合に限定されるものではない。ステップS 1 7にて自動生成したプログラムコードにおいて行おうとしている処理動作を特定し、これをプログラム自動生成知識D B 3 4に記録されている処理動作P 1と照合するようにしてもよい。自動生成したプログラムコードにおいて行おうとしている処理動作を特定する方法としては、予めプログラムの命令語のパターンと、処理動作P 1とを紐付けたデータテーブルを準備しておき、ステップS 1 7にて自動生成したプログラムコードの命令語を抽出し、データテーブルを参照することで行うようにしてもよい。また、意味と、処理動作P 1とを直接紐付けたデータテーブルを準備しておき、ステップS 1 5において特定した意味から、データテーブルを参照することで特定するようにしてもよい。
- [0099] このようにして処理動作P 1が特定できれば、これに関連付けられている構造化パターンP 3に基づいてプログラムの書き換えを行うようにしてもよい。また、これ以外には、処理動作P 1が特定できれば、これに対応するプログラムコードP 2から構造化パターンP 3させるための遷移パターンがプログラム自動生成知識D B 3 4に記憶されていることから、この遷移パターンを読み出し、それに沿ってステップS 1 7にて自動生成したプログラムコードを書き換えるようにしてもよい。
- [0100] また更に、プログラム自動生成知識D B 3 4において、効率化前のプログラムコードP 2から効率化されたプログラムへ変換するためのアルゴリズムが構造化パターンP 3として関連付けられている場合、ステップS 1 7にて自動生成したプログラムコードに最も類似するプログラムコードP 2に関連付けられているアルゴリズムを読み出すようにしてもよい。そして、この読み出したアルゴリズムに基づいて、ステップS 1 7にて自動生成したプログラムコードを効率化して書き換えるようにしてもよい。
- [0101] このプログラム自動生成知識D B 3 4におけるプログラムの書き換えが終

了した後、ステップS 2 0へ移行し、I T技術者やプログラマーによるコード編集が必要か否かを確認する。その結果、コード編集が必要な場合、或いはユーザ側からコード編集を希望する旨が受け付けられた場合には、ステップS 2 1へ移行する。一方、コード編集が不要な場合、或いはユーザ側からコード編集を希望する旨が受け付けられなかった場合には、終了となる。

[0102] ステップS 2 1に移行した場合、ステップS 1 9において書き換えれたプログラムコードに対して、I T技術者やプログラマーによるコード編集が施される。このコード編集は、手動によるプログラムコードの編集に加え、周知の編集ツールにより自動的に編集をするようにしてもよい。このステップS 2 1におけるコード編集が終了した後、再びステップS 2 0に戻り、コード編集がする必要がなくなればそのまま終了となる。

[0103] このように、本発明を適用したプログラムコード自動生成システム1 0 0によれば、ステップS 1 3～S 1 7において説明した処理動作を通じて、意図している業務処理をコンピュータ側のシステムにおいて自動的にプログラムコードを生成することができる。このため、極めて容易に、しかも熟練のI T技術者のような技術力がなくても、自動的にプログラムコードを生成することが可能となる。

[0104] これに加えて、本発明は、ステップS 1 9における構造化の処理を行うことができることから、熟練のI T技術者の知識や経験が反映された、スムーズで効率化が図られた短いプログラムコードを自動生成することが可能とある。これにより、プログラム開発の仕組みを根本的に変え、近い将来、慢性的なI T技術者不足が社会に甚大な影響を与えることを回避できる。

[0105] なお、プログラム自動生成知識D B 3 4において、処理動作P 1やこれに対応するプログラムコードP 2に対して、複数の構造化パターンP 3を関連付けておくことにより、ステップS 1 9において、より効率化された書き換え用のプログラムを複数に亘り生成することができる。その複数に亘り生成された書き換え用のプログラムの中から自動的に、又はユーザの意図に基づいて最適なものを選択することもでき、より高精度な構造化を実現すること

ができる。

[0106] ちなみに、上述したステップS 1 9における構造化プロセスを導入する代替として、例えば、ステップS 1 7におけるプログラムコードの自動生成時において、プログラム自動生成知識DB 3 4を参照することで、当該ステップS 1 9自体を省略するようにしてもよい。

[0107] かかる場合には、ステップS 1 7において、プログラムコードDB 3 2に基づいてプログラムを生成する際において、当初から一般的なプログラムコードを作るのではなく、効率化されたプログラムの作成を行う。即ち、一般的なプログラムを介さずに、直接的に効率化されたプログラムの作成を行う。

[0108] かかる場合には、少なくとも、このプログラム自動生成知識DB 3 4において、実現すべき処理動作P 1に対して効率化されたプログラムが関連付けられることが必要となる。実現すべき処理動作P 1は、上述したように文脈から読み取る。そして、読み取った文脈から、実現すべき処理動作P 1を特定し、この特定した実現すべき処理動作P 1に応じた、効率化されたプログラムを参照する。これにより、いかなる実現すべき処理動作P 2に対して、熟練のIT技術者であればどのようにプログラムを効率化するかについての知識を参照することができる。この知識を参照することで、プログラムコードDB 3 2に基づいてプログラムを生成する際において、その効率化したプログラムの知識を反映させることができ、その結果、一般的なプログラムを介さずに、直接的に効率化されたプログラムの作成を行うことが可能となる。

[0109] このような直接的に効率化されたプログラムの作成を行う場合、作成したプログラムについて、プログラム文法解析DB 3 3を参照することによる文法の正誤のチェックを行うようにしてもよい。即ち、この文法の正誤のチェックを行う対象としての生成すべきプログラムに、この直接的に作成されたプログラムを含めるようにしてもよい。

[0110] なお、本発明は、上述した実施の形態に限定されるものではない。例えば

、意味変換DB31は、図7に示すように、人工知能による学習データを利用するようにしてもよい。入力データは、単語、指示動作、指示動作及びこれに係り受けする単語の何れかで構成され、出力が意味となる。このような学習データを予め蓄積しておくことで意味変換DB31を生成しておく。そして、ステップS13の自然言語解析においてテキストデータから単語、指示動作、指示動作及びこれに係り受けする単語の何れかが抽出された場合には、この学習データを参照し、これに見合う探索解としての意味を抽出する。

[0111] また、プログラムコードDB32は、図8に示すように、人工知能による学習データを利用するようにしてもよい。入力データは、文脈（単語、指示動作、指示動作及びこれに係り受けする単語、意味等）で構成され、出力がプログラムコードとなる。このような学習データを予め蓄積しておくことでプログラムコードDB32を生成しておく。そして、ステップS17のプログラムコードの自動生成時においてこの学習データを参照し、文脈に見合う探索解としてのプログラムコードを抽出する。

[0112] 同様に、プログラム自動生成知識DB34は、図9に示すように、人工知能による学習データを利用するようにしてもよい。入力データは、実現すべき処理動作P1及び／又はプログラムコードP2であり、出力は、構造化パターンP3となる。このような学習データを予め蓄積しておくことでプログラム自動生成知識DB34を生成しておく。そして、ステップS19の構造化の実行時においてこの学習データを参照し、実現すべき処理動作及び／又はプログラムコードに見合う探索解としての構造化パターンP3を抽出する。

[0113] またプログラム文法解析DB33も同様に、人工知能による学習データを利用するようにしてもよいことは勿論である。

符号の説明

- [0114] 1 プログラムコード自動生成装置
2 端末

3 サーバ

4 通信網

100 プログラムコード自動生成システム

請求の範囲

[請求項1] 文書又は会話文からテキストデータを取得するテキストデータ取得手段と、

上記テキストデータ取得手段により取得されたテキストデータを自然言語解析することにより、文脈を抽出する文脈抽出手段と、

各文脈に対して1以上のプログラムコードが関連付けて記憶されるプログラムコードデータベースと、プログラム言語の文法書が記録されたプログラム文法解析データベースとを参照し、上記文脈抽出手段により抽出された文脈に応じた1以上のプログラムコードを割り当てるコード割当手段と、

プログラムコード又は当該プログラムコードにより実現すべき処理動作に対して、効率化されたプログラムの構造化パターンが関連付けられたプログラム自動生成知識データベースを参照し、上記コード割当手段により割り当てられたプログラムコードを効率化されたプログラムに書き換えるコード効率化手段とを備えること

を特徴とするプログラムコード自動生成システム。

[請求項2] 文書又は会話文からテキストデータを取得するテキストデータ取得手段と、

上記テキストデータ取得手段により取得されたテキストデータを自然言語解析することにより、文脈を抽出する文脈抽出手段と、

各文脈に対して1以上のプログラムコードが関連付けて記憶されるプログラムコードデータベースと、プログラム言語の文法書が記録されたプログラム文法解析データベースと、実現すべき処理動作に対して効率化されたプログラムが関連付けられたプログラム自動生成知識データベースとを参照し、上記文脈抽出手段により抽出された文脈に応じた1以上のプログラムを生成するコード割当手段とを備えることを特徴とするプログラムコード自動生成システム。

[請求項3] 上記文脈抽出手段は、上記自然言語解析することにより、単語、指

示動作、指示動作及びこれに係り受けする単語の何れかを抽出することにより、上記文脈を抽出すること

を特徴とする請求項 1 又は 2 記載のプログラムコード自動生成システム。

[請求項4] 上記文脈抽出手段は、単語、指示動作、指示動作及びこれに係り受けする単語の何れかと、意味とが互いに関連付けて記憶される意味変換データベースを参照し、新たに抽出した上記単語、指示動作、指示動作及びこれに係り受けする単語の何れかと関連する意味を探索することにより、上記文脈を抽出すること

を特徴とする請求項 3 記載のプログラムコード自動生成システム。

[請求項5] 上記コード割当手段は、上記プログラム文法解析データベースを参照し、生成すべきプログラムの文法を解析することにより、その正誤をチェックすること

を特徴とする請求項 1 又は 2 記載のプログラムコード自動生成システム。

[請求項6] 上記コード効率化手段は、プログラムコードに対して、効率化されたプログラムが関連付けられたプログラム自動生成知識データベースを参照し、上記コード割当手段により割り当てられたプログラムコードに関連する上記効率化されたプログラムを抽出し、これに書き換えること

を特徴とする請求項 1 記載のプログラムコード自動生成システム。

[請求項7] 上記コード効率化手段は、プログラムコードにより実現すべき処理動作に対して、効率化前の上記プログラムコードから効率化されたプログラムへ変換するためのアルゴリズムが上記構造化パターンとして関連付けられたプログラム自動生成知識データベースを参照し、上記コード割当手段により割り当てられたプログラムコードにより実現すべき処理動作に関連付けられた上記構造化パターンとしてのアルゴリズムに基づいて、上記書き換えを行うこと

を特徴とする請求項1記載のプログラムコード自動生成システム。

[請求項8]

上記文脈抽出手段は、単語、指示動作、指示動作及びこれに係り受けする単語の何れかと、意味とを教師データとして機械学習されたモデルを用い、入力を単語、指示動作、指示動作及びこれに係り受けする単語の何れかとし、出力を意味とする意味変換データベースを参照すること

を特徴とする請求項3記載のプログラムコード自動生成システム。

[請求項9]

上記コード割当手段は、文脈とプログラムコードとを教師データとして機械学習されたモデルを用い、入力を文脈とし、出力をプログラムコードとするプログラムコードデータベースを参照すること

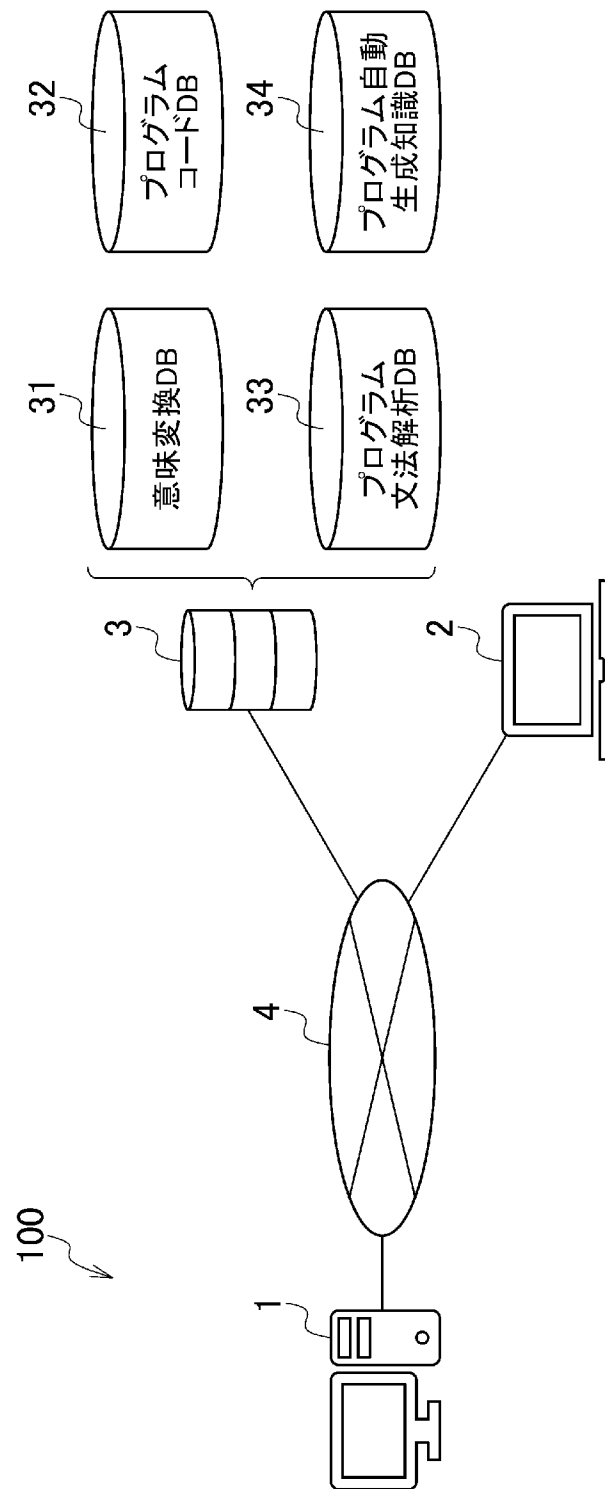
を特徴とする請求項1～8のうち何れか1項記載のプログラムコード自動生成システム。

[請求項10]

上記コード効率化手段は、プログラムコード又は当該プログラムコードにより実現すべき処理動作と、効率化されたプログラムの構造化パターンとを教師データとして機械学習されたモデルを用い、入力をプログラムコード又は当該プログラムコードにより実現すべき処理動作とし、出力を上記構造化パターンとするプログラム自動生成知識データベースを参照すること

を特徴とする請求項1記載のプログラムコード自動生成システム。

[図1]



[図2]

文脈		
指示動作	単語	意味
出力する	画面	データを見せる
出力する	ファイル	読み出す
登録する	ファイル	書き込む
照会する	画面、通信	データを見せる


 31

(a)

文脈		
指示動作	単語	意味
移す	データ	データを転送する
移す	データ	コピーする
移す	データ	移動する


 31

(b)

文脈		
指示動作	単語	意味
見せて	出力データ	プリンタに出力
見せて	出力データ	画面に出力
見せて	出力データ	通信回線に出力
見せて	出力データ	ファイルに出力


 31

(c)

[図3]

文脈		プログラムコード
指示動作又は単語		
送信		vary response=http.post
画面		text xxxx
ファイル		import xxxx file.write...

(a)

文脈			プログラムコード
指示動作	単語	意味	
移す	データ	データを転送する	send / post
移す	データ	コピーする	xcopy
移す	データ	移動する	copy / move

(b)

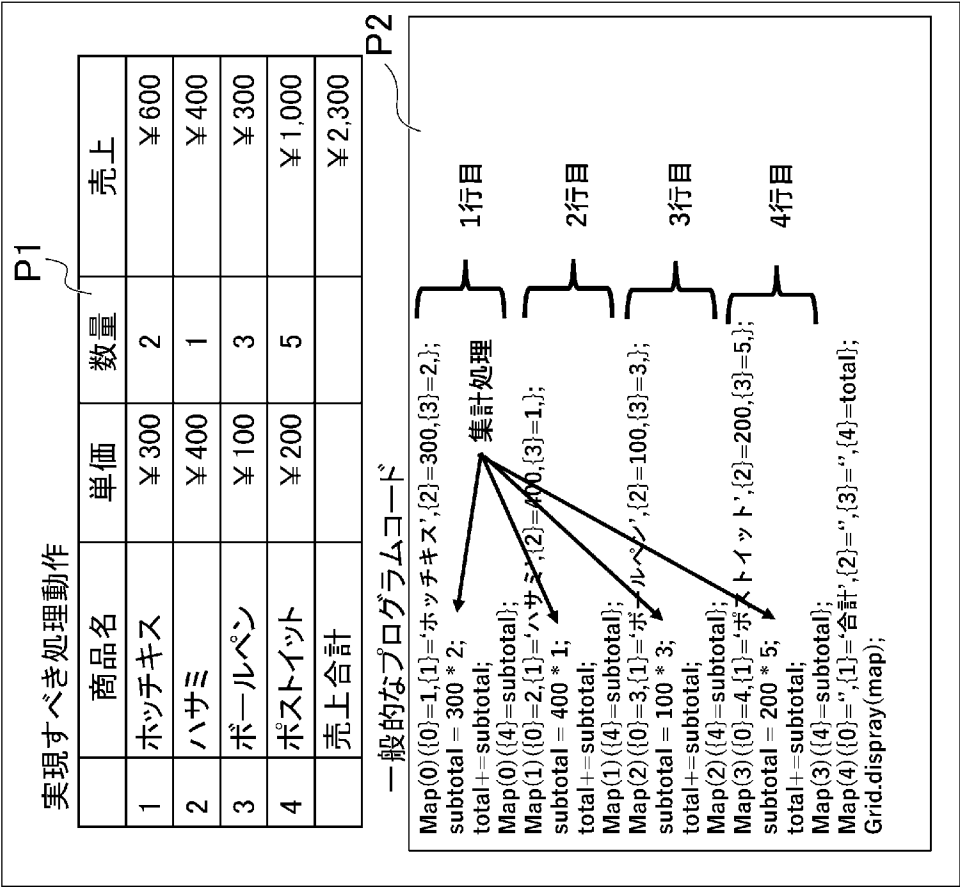
文脈			プログラムコード
指示動作	単語	意味	
見せて	出力データ	プリンタに出力	lp
見せて	出力データ	画面に出力	print xxxx
見せて	出力データ	通信回線に出力	request
見せて	出力データ	ファイルに出力	write

(c)

[図4]

34

画面に商品毎の売上データ等を表示する例



構造化パターン

P3

ここだけを出力データとして
表示すればよい

集計処理

```
for item in item_list
  i+=1;
  subtotal = item(1) * item(2);
  Map(i)({0}=i,{1}=item(1),{2}=item(2),{3}=item(3),{4}=subtotal);
  total += subtotal;

Map(i)({0}='',{1}='合計',{2}='',{3}='',{4}=total);
Grid.display(map);
```

共通ロジック
(構造化)

表示用
データ

```
item_list [{{'ホットキス','300','2'},{'ハサミ','400','1'}},
{'ボールペン','100','3'},{'ポストイット','200','5'}];
```

[図5]

34

エラー処理を複数箇所に行う例

P2

```
try
  write #1 '今日は体調が悪いです。';
  echo 'complete';
Catch exception e
  logger.output(e.code);
  logger.output('ファイルの書き込みに失敗しました。');
  logger.output(e.description);

try
  put #1 '今日の売上は¥101,098'です。';
  echo 'complete';
Catch exception e
  logger.output(e.code);
  logger.output('画面への出力に失敗しました。');
  logger.output(e.description);

try
  put #1 '今日の天気は晴れです。';
  echo 'complete';
Catch exception e
  logger.output(e.code);
  logger.output('音声出力に失敗しました。');
  logger.output(e.description);
```

Write #1 を実行した際に
エラーが発生した場合は、
logger.output(e.code)へ
処理が移動する。

プログラム中、
幾つもの箇所でエ
ラー処理のコード
を記述しなければ
ならない。

P3

```
err_func(e,msg,description) {
  logger.output(e.code);
  logger.output(msg);
  logger.output(e.description);
}

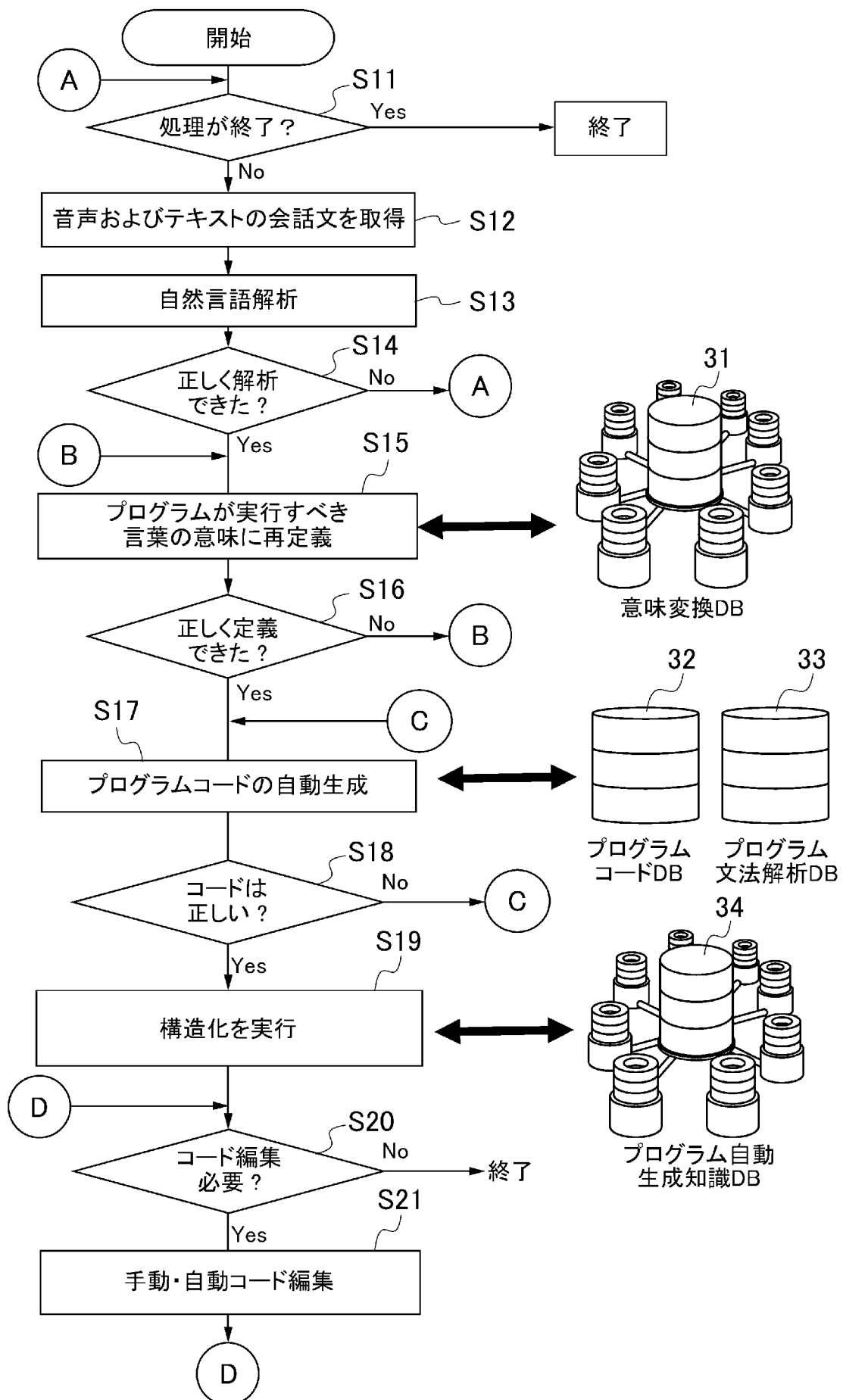
err_func(e, 'ファイルの書き込みに失敗しました');
err_func(e, '画面への出力に失敗しました。');
err_func(e, '音声出力に失敗しました。');
```

共通ロジック
(構造化)

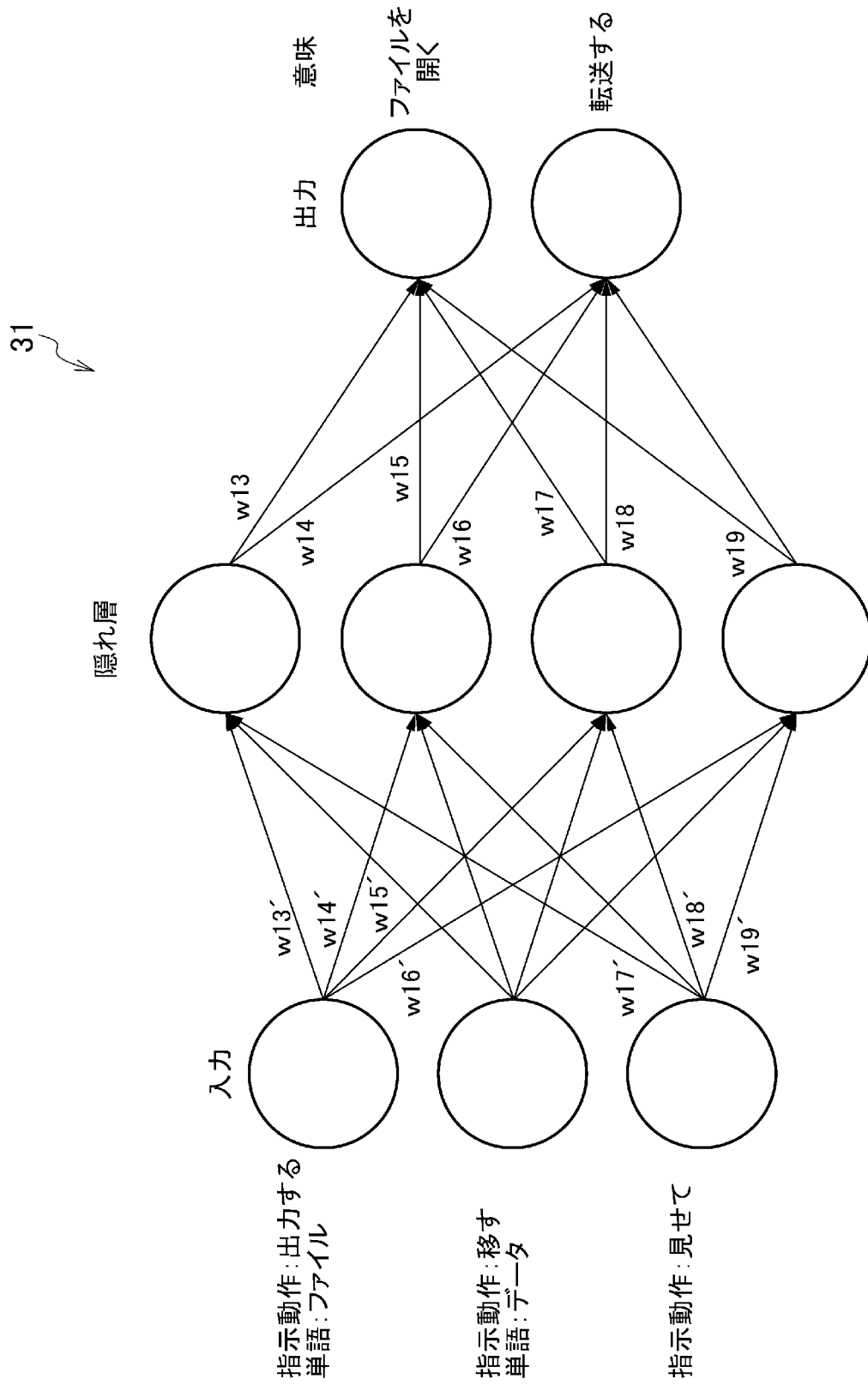
出力データ
だけを記述
すればよい



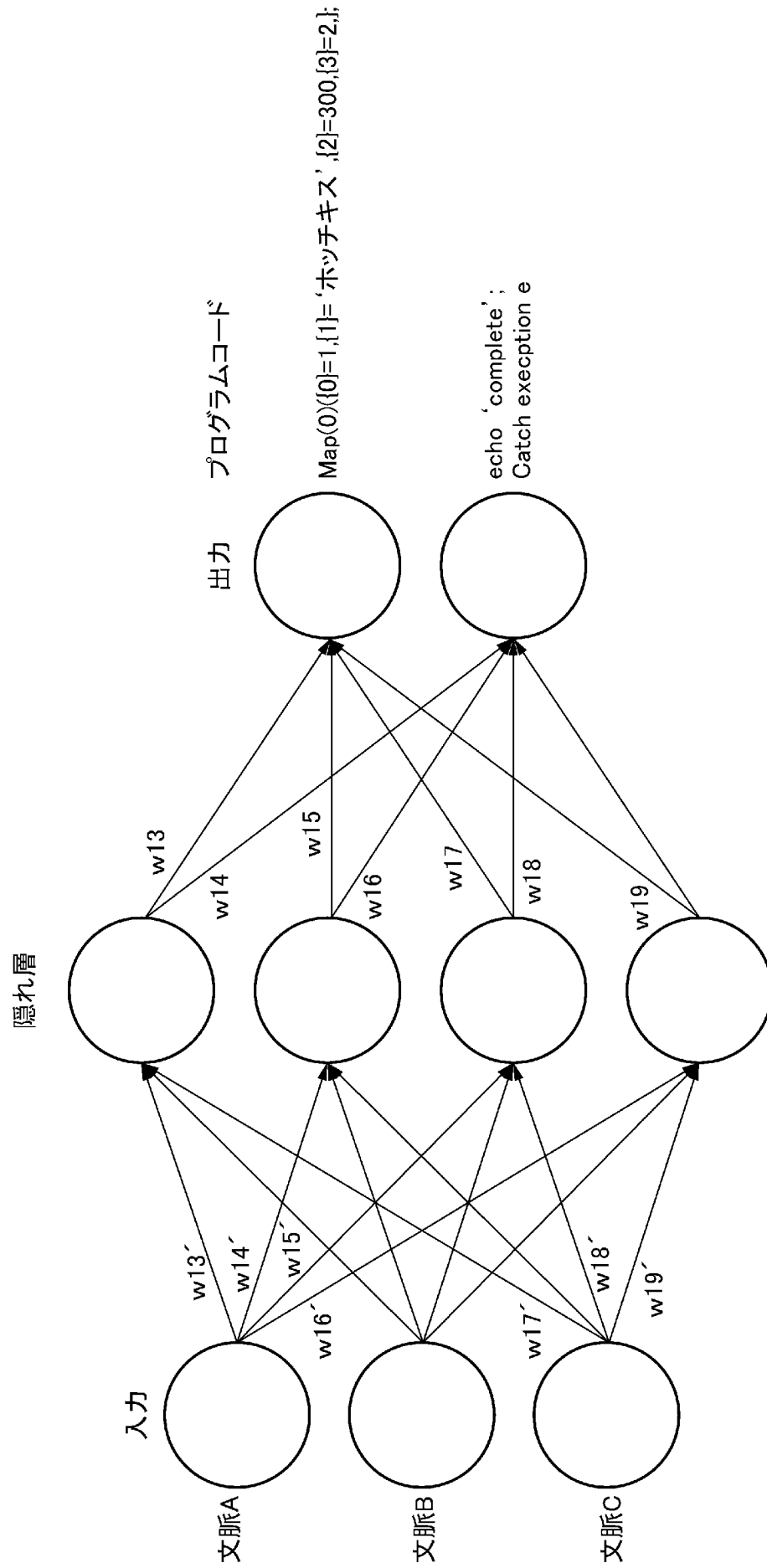
[図6]



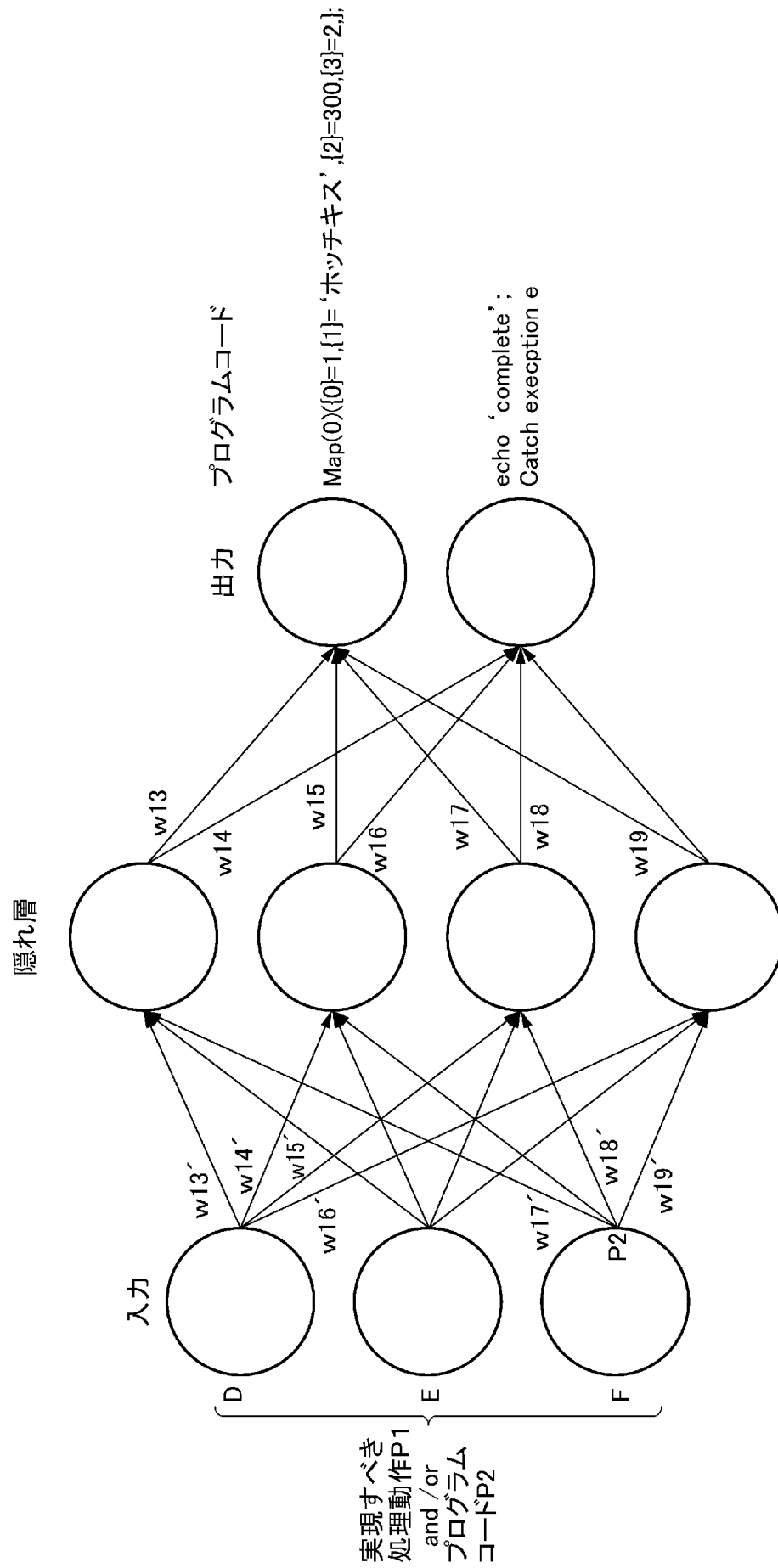
[図7]



[図8]



[図9]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2022/008237

A. CLASSIFICATION OF SUBJECT MATTER**G06F 8/35**(2018.01)i; **G06F 40/151**(2020.01)i; **G06F 40/211**(2020.01)i

FI: G06F8/35; G06F40/211; G06F40/151

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)

G06F8/35; G06F40/151; G06F40/211

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 1971-2022
 Registered utility model specifications of Japan 1996-2022
 Published registered utility model applications of Japan 1994-2022

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.
Y	WO 2021/106380 A1 (SOPPPA CORP) 03 June 2021 (2021-06-03)	1-9
A	paragraphs [0017]-[0073], [0075]-[0095]	10
Y	JP 10-254688 A (TOSHIBA CORP) 25 September 1998 (1998-09-25)	1-9
A	paragraphs [0043]-[0044]	10
Y	JP 4-88531 A (FUJI XEROX CO LTD) 23 March 1992 (1992-03-23)	1-9
A	p. 3, lower left column, lines 1-19	10
Y	JP 2008-243019 A (TOSHIBA CORP) 09 October 2008 (2008-10-09)	1-9
A	paragraphs [0022]-[0038]	10

☒ 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 March 2022

Date of mailing of the international search report

05 April 2022

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

International application No.

PCT/JP2022/008237

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	新野淳一, 「Amazon CodeGuru」が正式版に 機械学習によりコードの問題部分や遅い部分を指摘, ITmedia NEWS [オンライン], 02 July 2020, [retrieval date 24 March 2022], Internet: <URL:https://www.itmedia.co.jp/news/articles/2007/02/news090.html> entire text, (ITmedia NEWS [online]), non-official translation (NIINO, Junichi. "Amazon CodeGuru" becomes official version. Point out problematic or slow parts of code with machine learning.)	10

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/JP2022/008237

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
WO	2021/106380	A1	03 June 2021	JP 2021-86409 A paragraphs [0015]-[0072], [0074]-[0094]	
JP	10-254688	A	25 September 1998	(Family: none)	
JP	4-88531	A	23 March 1992	(Family: none)	
JP	2008-243019	A	09 October 2008	(Family: none)	

A. 発明の属する分野の分類（国際特許分類（IPC）） G06F 8/35(2018.01)i; G06F 40/151(2020.01)i; G06F 40/211(2020.01)i FI: G06F8/35; G06F40/211; G06F40/151		
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）） G06F8/35; G06F40/151; G06F40/211		
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 1971-2022年 日本国実用新案登録公報 1996-2022年 日本国登録実用新案公報 1994-2022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
Y	WO 2021/106380 A1（ソプラ株式会社）03.06.2021（2021-06-03） 段落[0017]-[0073], [0075]-[0095]	1-9
A		10
Y	JP 10-254688 A（株式会社東芝）25.09.1998（1998-09-25） 段落[0043]-[0044]	1-9
A		10
Y	JP 4-88531 A（富士ゼロックス株式会社）23.03.1992（1992-03-23） 第3頁左下欄1-19行	1-9
A		10
Y	JP 2008-243019 A（株式会社東芝）09.10.2008（2008-10-09） 段落[0022]-[0038]	1-9
A		10
☒ 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 24.03.2022		国際調査報告の発送日 05.04.2022
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） 北川 純次 5B 3650 電話番号 03-3581-1101 内線 3545

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	新野淳一, 「Amazon CodeGuru」が正式版に 機械学習によりコードの問題部分や遅い部分を指摘, ITmedia NEWS [オンライン], 2020.07.02, [検索日 2022.03.24], インターネット: <URL: https://www.itmedia.co.jp/news/articles/2007/02/news090.html > 全文	10

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/JP2022/008237

引用文献			公表日	パテントファミリー文献	公表日
WO	2021/106380	A1	03.06.2021	JP 2021-86409 A 段落[0015]-[0072], [0074]-[0094]	
JP	10-254688	A	25.09.1998	(ファミリーなし)	
JP	4-88531	A	23.03.1992	(ファミリーなし)	
JP	2008-243019	A	09.10.2008	(ファミリーなし)	