

AI 特許紹介(73)
AI 特許を学ぶ！究める！
～Semantic Interpreter 特許～

2025年2月10日
河野特許事務所
所長弁理士 河野英仁

「AI 特許紹介」シリーズは、注目すべき AI 特許のポイントを紹介します。熾烈な競争となっている第4次産業革命下では AI 技術がキーとなり、この AI 技術・ソリューションを特許として適切に権利化しておくことが重要であることは言うまでもありません。

AI 技術は Google, Microsoft, Amazon を始めとした IT プラットフォーマ、研究機関及び大学から毎週のように新たな手法が提案されており、また AI 技術を活用した新たなソリューションも次々とリリースされています。

本稿では米国先進 IT 企業を中心に、これらの企業から出願された AI 特許に記載された AI テクノロジー・ソリューションのポイントをわかりやすく解説致します。

1.概要

特許権者 Microsoft

出願日 2024年5月19日

登録日 2024年11月14日

登録番号 WO2024233714

発明の名称 プログラム合成によるアプリケーションでの自然言語コマンド用セマンティックインタープリター

714 特許は、パワーポイント等のオフィスアプリケーション上で実行させたいクエリをユーザから受け付け、受け付けたクエリに基づき LLM に、アプリケーションに固有のドメイン固有言語 (DSL: Domain Specific Language) プログラムコードを生成させることにより、自然言語によりオフィスアプリケーションを動作させる Semantic Interpreter 技術に関する。

2.特許内容の説明

図 1B-1C は、自然言語コマンド技法を実装するサンプルアプリケーションのサンプルユーザインターフェース 105 を示す図である。

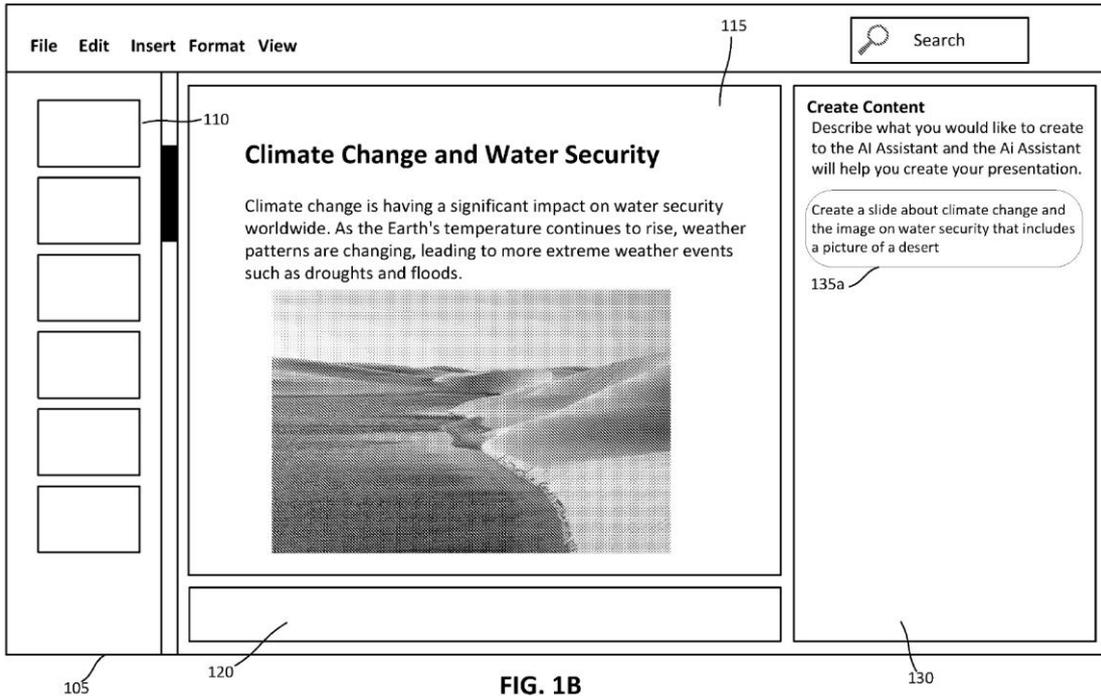


FIG. 1B

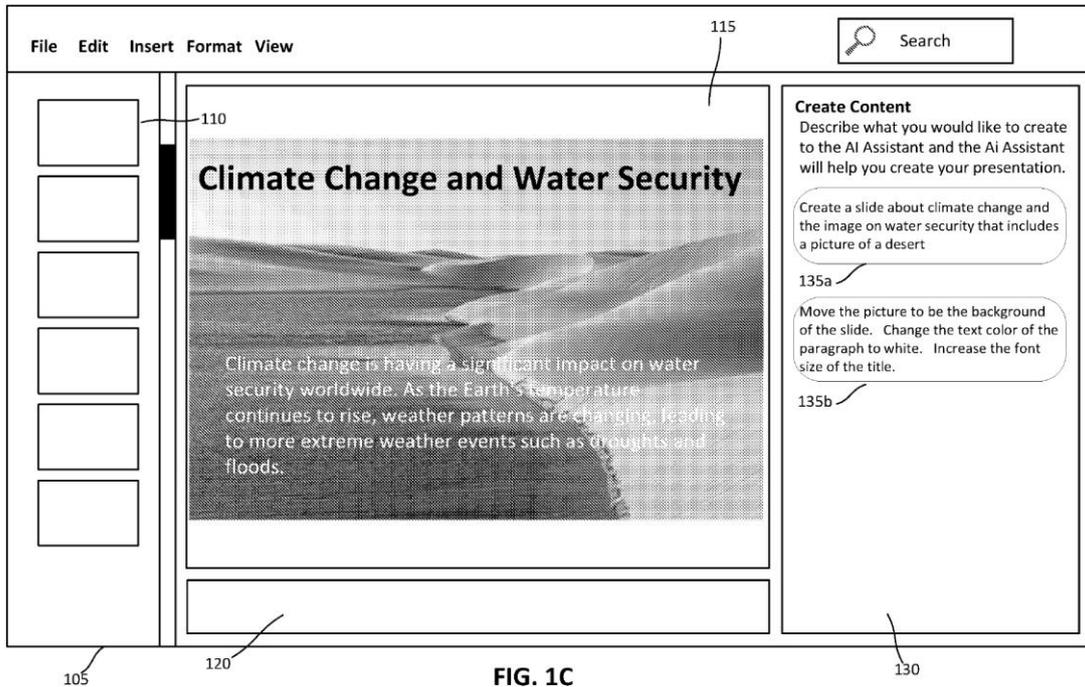


FIG. 1C

図 1B は、初期プロンプトがユーザによって入力され、以下に述べる技術に従って生成されたスライドがユーザインターフェース 105 のコンテンツペイン 115 に表示される例を示している。図 1C は、ユーザがテキスト入力フィールド 135b に第 2 の自然言

語コマンドを入力し、アプリケーションが第1の自然言語コマンドに回答して作成されたスライドを変更する例を示している。図2は、アプリケーションで自然言語コマンドを提供する技術を実装するコンピューティング環境200を示す図である。

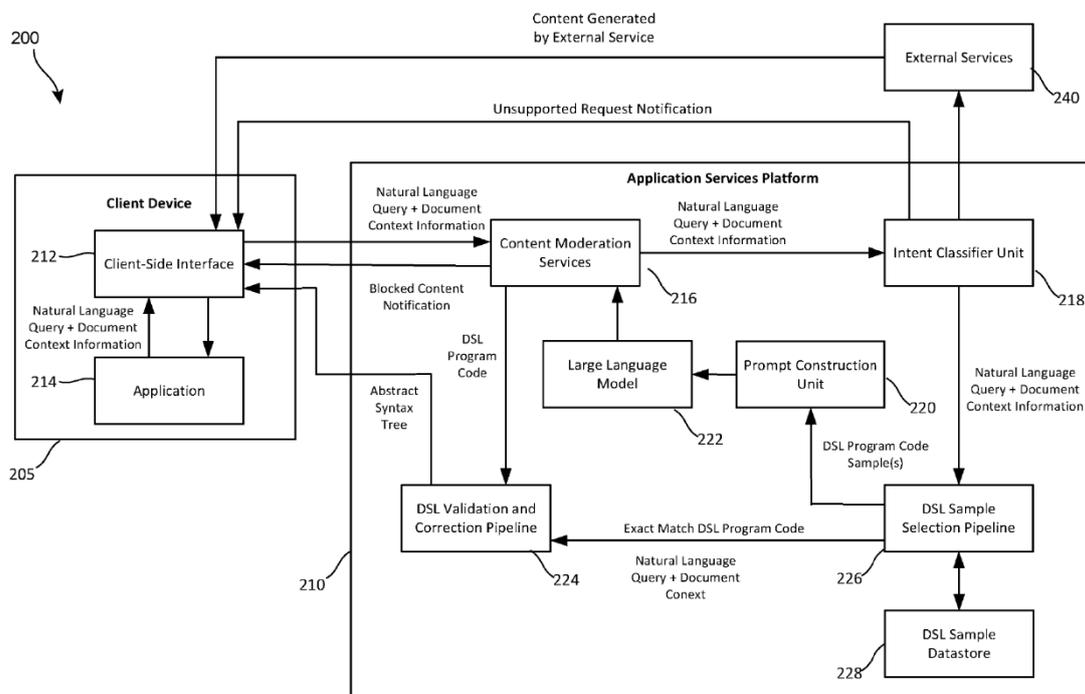


FIG. 2

図2に示す実装例では、コンピューティング環境には、クライアントデバイス205とアプリケーション サービスプラットフォーム210が含まれる。クライアント側インターフェース212は、アプリケーション214から自然言語クエリを受信し、その自然言語クエリを処理のためにアプリケーションサービスプラットフォーム210に送信する。

クライアント側インターフェース212は、自然言語クエリとともにアプリケーション214からドキュメントコンテキスト情報も受信する。自然言語クエリは、図1Cに示す例のように、新しいコンテンツの作成と既存のコンテンツの変更を目的としている。クライアント側インターフェース212は、LLM222またはアプリケーションサービスプラットフォーム210の他の生成モデルによって生成された実行可能コードの中間表現を受信する。以下の例で説明するように、中間コードは、一部の实装では抽象構文ツリー (AST : abstract syntax tree) として表される。

アプリケーションサービスプラットフォーム210には、コンテンツモデレーションサービス216が含まれている。コンテンツモデレーションサービス216は、自然言語クエリと生成モデルによって生成されたコンテンツを分析して、自然言語クエリと生成モデルによって生成されたコンテンツに、不快または不快な可能性のあるコンテンツが

含まれていないことを確認する。

アプリケーションサービスプラットフォーム 210 は、クライアントデバイス 205 から受信した自然言語クエリとドキュメントコンテキストをコンテンツモデレーションサービス 216 に提供する。ドキュメントコンテキストは、自然言語クエリが既存の電子コンテンツを変更する可能性がある場合に、クライアント側インターフェース 212 によって提供される。このようなドキュメントコンテンツ情報の例は以下の通りである。

```
Document Context Information
{
  "slides": [
    {
      "index": 0,
      "shapes": [
        {
          "index": 0,
          "textRange": {
            "text": "Video Games",
            "color": "#000000",
            "bold": false,
            "italic": false,
            "underline": "None",
            "fontName": "Segoe UI Semibold",
            "size": 36,
            "bulleted": false
          },
          "type": "GeometricShape",
          "name": "Title 1",
          "left": 52.1099212598425,
          "top": 36,
          "height": 43.6218897637795,
          "width": 976.05,
          "lineColor": "",
          "lineDashStyle": "Solid",
          "lineWeight": -1,
          "lineTransparency": -1,
          "fillColor": ""
        },
        {
          "index": 1,
          "textRange": {
            "text": "Video games [a] also known as computer games, are electronic games that involve interaction with a user interface or input device – such as a joystick, controller, keyboard, or motion sensing device – to generate audiovisual feedback. })",
            "color": "",
            "bold": false,
            "italic": false,
            "underline": "None",
            "fontName": "",
            "size": 20,
            "bulleted": false
          },
          "type": "GeometricShape",
          "name": "body",
          "left": 95.8344881889764,
          "top": 103.78125984252,
          "height": 411.984409448819,
          "width": 838.551732283465,
          "lineColor": "",
          "lineDashStyle": "Solid",
          "lineWeight": -1,
          "lineTransparency": -1,
          "fillColor": ""
        }
      ]
    }
  ]
}
```

FIG. 12

ドキュメントコンテキスト情報は、ここではコンテキストツリーと呼ばれる軽量ツリー構造に格納される。ドキュメントコンテンツ情報には、プレゼンテーションスライドまたはその他のドキュメントのレイアウト、スライドまたはドキュメント内の画像の位置とサイズ、スライドまたはドキュメントに含まれるテキストとそのテキストの属性、ドキュメント内のテキストの位置、スライドまたはドキュメントに含まれる画像を取得するためのソースファイルまたは位置、およびスライドまたはドキュメントの要素とそのコンテンツを説明するその他の情報が含まれる。LLM 222 は、既存のコンテンツを

変更するときはこの情報を使用してコンテキストを提供する。

LLM 222 は、実行されるとアプリケーション 214 にコンテンツを変更するアクションを実行させる命令を含む DSL プログラムコードを生成する。DSL プログラムの例を以下に示す。

DSL Program Code

```
# Add cube to the slide and set the color to teal
slides = select_slides()
shapes = insert_shapes(shapeType="cube", slides=slides)
format_shapes(shapes=shapes, fillColor="teal", fillTransparency=0.3, height=200, width=200)
text = select_text(scope=shapes)
text = insert_text(precededBy=title, text="Box")
format_text(textRanges=text, horizontalAlignment="Center")
```

FIG. 10

LLM222 は、既存のスライドまたはドキュメントに組み込むためにクライアントデバイス 205 に提供される可能性のある、グラウンドテキストコンテンツを含むテキストコンテンツを生成する。LLM によって生成されたテキストコンテンツは、LLM222 によって生成された DSL プログラムコード、DSL プログラムコードから生成された AST、およびトランスパイラによって AST から生成されたプラットフォーム固有のプログラムコードに組み込まれる。以下に AST を示す。

Abstract Syntax Tree

```
{
  "statements": [
    {
      "name": "select_slides",
      "returnVariable": "slides",
      "arguments": []
    },
    {
      "name": "insert_shapes",
      "returnVariable": "shapes",
      "arguments": [
        {
          "name": "shapeType", "value": "\"cube\""
        },
        {
          "name": "slides", "value": "slides"
        }
      ]
    },
    {
      "name": "format_shapes",
      "returnVariable": null,
      "arguments": [
        {
          "name": "shapes", "value": "shapes"
        },
        {
          "name": "fillColor", "value": "\"teal\""
        },
        {
          "name": "fillTransparency", "value": "0.3"
        },
        {
          "name": "height", "value": "200"
        },
        {
          "name": "width", "value": "200"
        }
      ]
    },
    {
      "name": "select_text",
      "returnVariable": "text",
      "arguments": [
        {
          "name": "scope", "value": "shapes"
        }
      ]
    },
    {
      "name": "insert_text",
      "returnVariable": "text",
      "arguments": [
        {
          "name": "precededBy", "value": "text"
        },
        {
          "name": "text", "value": "\"Box\""
        }
      ]
    },
    {
      "name": "format_text",
      "returnVariable": null,
      "arguments": [
        {
          "name": "textRanges", "value": "text"
        },
        {
          "name": "horizontalAlignment", "value": "\"Center\""
        }
      ]
    }
  ]
}
```

FIG. 11

コンテンツモデレーションサービス 216 は、自然言語クエリとドキュメントコンテンツを分析し、不快または不愉快なコンテンツを含む自然言語クエリまたはドキュメントコンテンツに応答して、ブロックされたコンテンツ通知をクライアントデバイス 205 のクライアント側インターフェース 212 に返す。自然言語クエリとドキュメントコンテンツ（自然言語クエリと共に受信された場合）は、コンテンツモデレーションサービス 216 が不快または不愉快なコンテンツが検出されなかったと判断すると、インテント分類ユニット 218 への入力として提供される。

意図分類ユニット 218 は、ユーザが入力した自然言語クエリが、アプリケーションサービスプラットフォーム 210 によって提供されるサービスの範囲内にあるかどうかを判定する。意図分類ユニット 218 が、自然言語クエリとドキュメントコンテンツがサポートされているアクションカテゴリに関連付けられていると判断すると、自然言語クエリとドキュメントコンテンツが DSL 選択パイプライン 226 に提供され、処理される。

DSL 選択パイプライン 226 は、自然言語クエリとドキュメントコンテキストを分析し、DSL サンプルデータストア 228 からサンプル DSL コード例を選択する。サンプル DSL コード例は、生成モデルが有効な DSL プログラムを生成するのに役立つように、LLM 222 に提供されるプロンプトに含まれている。LLM 222 に提供されるプロンプトのサイズは限られているため、DSL サンプル選択パイプラインは、プロンプトに含める最も関連性の高いサンプルを DSL サンプルデータストア 228 から選択する。

DSL 選択パイプライン 226 は、自然言語クエリで表されたユーザインテントを実装するために必要な DSL と完全に一致するものを見つけた場合、サンプル DSL を DSL 検証および修正パイプライン 224 に提供し、サンプル DSL 内のエラーを特定して修正する。それ以外の場合、DSL 選択パイプライン 226 が、自然言語クエリで表現された意図を実装するために必要な DSL の正確な一致を識別できない場合、DSL 選択パイプライン 226 は、サンプル DSL サンプルのセットをプロンプト構築ユニット 220 に提供し、LLM 222 にプロンプトを生成して新しい DSL プログラムを生成する。DSL 選択パイプライン 226 は、新しい DSL を生成するためのプロンプトの一部として、サンプル DSL プログラムのセットを LLM 222 に提供する。DSL 選択パイプライン 226 は、自然言語クエリとドキュメントコンテキストもプロンプト構築ユニット 220 に渡す。

プロンプト構築ユニット 220 は、自然言語クエリで表現されたユーザの意図を実装する出力として DSL プログラムコードを生成する生成モデルである LLM222 へのプロンプトを構築する。プロンプト構築ユニット 220 には、自然言語クエリと、DSL 選択パイプライン 226 によって提供される DSL サンプルが含まれる。プロンプト構築ユニット 220 には、自然言語プロンプトを使用してアプリケーション 214 内の既存のコンテンツを変更するドキュメントコンテキストも含まれる。

プロンプト構築ユニット 220 は、DSL サンプル選択パイプライン 226 のエンティティ分類ユニット 502 によって識別されたエンティティに基づいて、プロンプトに追加のコンテンツを含める。DSL サンプル選択パイプライン 226 は、このエンティティ情報をプロンプト構築ユニット 220 に提供する。プロンプト構築ユニット 220 は、エンティティ情報を使用して、DSL 構文記述から各エンティティの構文記述を取得する。

LLM222 は、プロンプト構築ユニット 220 からプロンプトを受信し、プロンプトを分析し、自然言語クエリで表現されたユーザインテントを実装するための DSL プログラムコードを出力する。一部の実装では、LLM 222 は自然言語クエリに基づいて追加

コンテンツを生成する。追加コンテンツには、テキストコンテンツ、画像、ビデオ、およびモデルでサポートされているその他の種類のコンテンツが含まれる。インテント分類ユニット 218 によって検出されたサポートされていない要求は、外部サービス 240 にリダイレクトされるか、インテント分類ユニット 218 によって、ユーザがサポートされていない操作を要求したという通知がクライアントデバイス 205 に送信される。

LLM 222 は、DSL プログラムコードとその他の生成されたコンテンツをコンテンツモデレーションサービス 216 に提供し、コンテンツモデレーションサービス 216 は DSL を分析して、DSL プログラムコードと生成モデルによって生成されたコンテンツに、潜在的に不快または不愉快な内容が含まれていないことを確認する。コンテンツモデレーションサービス 216 は、LLM 222 が不注意で不快または不愉快と思われる DSL プログラムコードまたはその他のコンテンツを生成したと判断した場合、クライアントデバイス 205 のクライアント側インターフェース 212 に、自然言語クエリを処理できなかったことを示すブロックされたコンテンツ通知を送信する。ユーザは、クエリを修正して再送信を試みることがでる。

コンテンツモデレーションサービス 216 は、LLM 222 によって生成された DSL コードおよびその他のコンテンツを DSL 検証および修正パイプライン 224 に提供する。自然言語プロンプトおよびドキュメントコンテキストも、DSL 検証および修正パイプライン 224 に提供できる。DSL 検証および修正パイプライン 224 は、LLM222 によって生成された DSL コードを分析して、プログラムコードの構文が有効であることを確認し、プログラムコードで構文エラーが検出された場合に DSL プログラムコードを修正する。抽象構文ツリーは、ステートメント、各ステートメントに提供されるパラメータ、および DSL プログラムコードからの関連メタデータを、簡単に解析して構文を検証できるツリー構造で表す。抽象構文ツリーはクライアントデバイス 205 に提供され、クライアント側インターフェース 212 は、抽象構文ツリーの中間表現を、クライアントデバイス 205 で実行できるプラットフォーム固有の実装コードに変換する。

3.クレーム

714 特許のクレーム 1 は以下の通りである。

1.データ処理装置において、

プロセッサと、

実行可能命令を記憶した機械可読媒体とを備え、実行されると、プロセッサが単独で、または他のプロセッサと組み合わせて、以下の操作を実行する：

第1のクライアントデバイスの第1のユーザインターフェースを介して第1のユーザから第1の自然言語クエリを受信し、第1の自然言語クエリは、第1のクライアント

デバイス上の第1のアプリケーションを使用して実行される1つ以上のアクションのテキスト記述を含み、

第1の自然言語クエリの属性に基づいて、サンプル DSL プログラムコードデータストアから1つ以上のドメイン固有言語 (DSL: Domain Specific Language) プログラムコードサンプルを、以下の処理により選択し、

第1の大規模言語モデルに関連付けられた LLM ベースのエンコーダを使用して第1の自然言語クエリのクエリ埋め込みを決定し、

該クエリ埋め込みを、DSL プログラムコードデータストアの各 DSL プログラムコードサンプルに関連付けられたサンプル埋め込みと比較し、サンプル埋め込みは、第1の LLM に関連付けられた LLM ベースのエンコーダを使用して決定されており、

クエリ埋め込みと、1つ以上のサンプル DSL プログラムコードサンプルに関連付けられたサンプル埋め込みとの類似性に基づいて、1つ以上のサンプル DSL プログラムコードサンプルを選択し、

第1のユーザからの第1の自然言語クエリと1つ以上の DSL プログラムコードサンプルに基づいて、第1の LLM の第1のプロンプトを生成し、

第1の LLM に第1のプロンプトを提供して第1のプログラムコードを取得し、このコードが実行されると、第1のアプリケーションが1つ以上のアクションを実行し、第1のプログラムコードはドメイン固有言語 (DSL) で実装されており、

第1プログラムコード内の構文エラーを識別するために、第1プログラムコードの構文を分析し、

修正されたプログラムコードを生成するために、第1プログラムコード内の構文エラーを修正し、

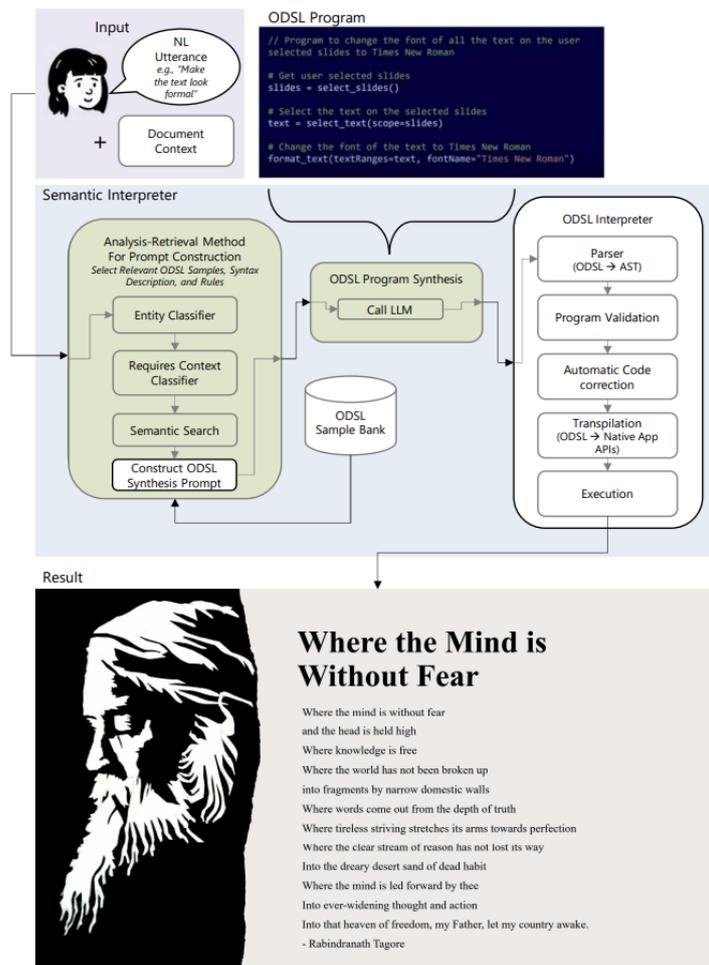
第一クライアントデバイスに第一プログラムコードを実行させて、第一クライアントデバイス上の第一アプリケーションにおける1つ以上のアクションを実行させる。

4. 本特許に関連する論文

本特許に関する論文 “Natural Language Commanding via Program Synthesis”¹ が、Microsoft の Apurva Gandhi 氏らにより公表されている。

下記図は、セマンティックインタープリターのアーキテクチャと全体的なアプローチを図示したものである。ユーザが PPT スライドに関し、「文章をフォーマルに見せて下さい」と発話する。

¹ William Fedus, et al. “Natural Language Commanding via Program Synthesis” arXiv:2101.03961v3 [cs.LG] 16 Jun 2022



セマンティックインタープリターは、プログラム合成用の LLM を使用した分析検索の few-shot プロンプトアプローチを活用して、ユーザの発話を ODSL プログラムに変換する。その後、ODSL プログラムは検証され、実行のためにアプリのネイティブ API にトランスパイルされる。最終的に PPT 文書が「恐れのない心がある場所・・・(Where the Mind is Without Fear)」とフォーマルな内容に変更される。

以上

著者紹介

河野英仁

河野特許事務所、所長弁理士。立命館大学情報システム学博士前期課程修了、米国フランクリンピアースローレンター知的財産権法修士修了、中国清華大学法学院知的財産夏季セミナー修了、MIT(マサチューセッツ工科大学)コンピュータ科学・AI 研究所 AI コース、生成 AI ビジネスコース修了。

[AI 特許コンサルティング](#)、[医療 AI 特許コンサルティング](#)の他、米国・中国特許の権

利化・侵害訴訟を専門としている。著書に「世界のソフトウェア特許(共著)」、「FinTech 特許入門」、「[AI/IoT 特許入門 3](#)」、「[ブロックチェーン 3.0](#)(共著)」がある。