

(19)日本国特許庁(JP)

(12)特許公報(B1)

(11)特許番号

特許第7506957号  
(P7506957)

(45)発行日 令和6年6月27日(2024. 6. 27)

(24)登録日 令和6年6月19日(2024. 6. 19)

(51)Int. Cl. F I  
G 0 6 F 8/34 (2018. 01) G 0 6 F 8/34

請求項の数 9 (全 36 頁)

(21)出願番号 特願2024-43116(P2024-43116)  
(22)出願日 令和6年3月19日(2024. 3. 19)  
審査請求日 令和6年3月24日(2024. 3. 24)

早期審査対象出願

(73)特許権者 521079868  
株式会社c h i p p e r  
東京都品川区東五反田2丁目5番2号  
(74)代理人 100215027  
弁理士 留場 恒光  
(72)発明者 十時 悠径  
東京都品川区東五反田2丁目5番2号  
(72)発明者 遠藤 聡志  
東京都千代田区有楽町2丁目7番1号有楽  
町イトシア12階  
審査官 久々宇 篤志

最終頁に続く

(54)【発明の名称】 開発支援システムおよび開発支援プログラム

(57)【特許請求の範囲】

【請求項1】

ワークスペース作成部、大規模言語モデル入出力部、およびプロンプト実行部を備える開発支援システムであって、

前記ワークスペース作成部は、

ワークスペースを表示するワークスペース表示部と、

ユーザの操作により、前記ワークスペース上にプロンプトノードを作成するノード作成部と、

一のプロンプトノードと他のプロンプトノードとを接続するエッジを作成するエッジ作成部と、を備え、

前記プロンプトノードは、

大規模言語モデルの選択を受け付ける大規模言語モデル選択欄と、

プロンプトを受け付けるプロンプト受付欄と、

前記プロンプトに対する前記大規模言語モデルの出力結果を表示する出力結果表示欄と

、

実行ボタンと、

再実行ボタンと、を含み、

前記実行ボタンは、当該実行ボタンが押下されたプロンプトノードのみでプロンプトの送信と出力結果の受信を行うものであり、

また前記再実行ボタンは、最初のプロンプトノードから当該再実行ボタンが押下された

プロンプトノードまでの各プロンプトノードにおいてプロンプトの送信と出力結果の受信を行うものであり、

前記大規模言語モデル入出力部は、  
選択されている前記大規模言語モデルを取得する大規模言語モデル取得部と、  
前記プロンプトを前記大規模言語モデルに入力として提供するプロンプト提供部と、  
前記プロンプトに対する前記大規模言語モデルの出力結果を取得する出力結果取得部と、  
を備え、

前記プロンプト実行部は、  
ユーザが選択しているプロンプトノードを取得する選択ノード取得部と、  
最初のプロンプトノードから前記ユーザが選択しているプロンプトノードまでのルート  
を取得するルート取得部と、

前記ルート上の各プロンプトノードについて、最初のプロンプトノードから順に前記大規模言語モデル入出力部による大規模言語モデル入出力処理を実行する逐次入出力部と、  
を備えることを特徴とする、開発支援システム。

【請求項 2】

前記プロンプトノードはさらに、  
新たなノードを作成し、元のノードの親ノードと前記新たなノードとをエッジで結合する複製ボタンを含み、

前記新たなノードは、前記元のノードで選択されている大規模言語モデルとプロンプトとを引き継ぐものであり、

また、前記エッジごとに、エッジが結合するノードの情報を保存することを特徴とする、請求項 1 に記載の開発支援システム。

【請求項 3】

さらに、コメント作成部を備え、  
前記コメント作成部は、ユーザのコメントを取得するコメント取得部と、  
取得しているコメントを表示するコメント表示部と、  
を備えることを特徴とする、請求項 1 に記載の開発支援システム。

【請求項 4】

前記ワークスペース作成部が、さらに、  
前記プロンプト受付欄に含まれる特定の記述をもとに変数を取得して保存する変数取得部と、  
変数の変更を受け付けて当該変数を置換する変数置換部と、  
を備えることを特徴とする、請求項 1 に記載の開発支援システム。

【請求項 5】

前記ワークスペース上にプロンプトノードを作成するノード作成部が、  
前記ワークスペース上にプロンプトノードおよび情報提供ノードを作成するプロンプトノード作成部であり、  
前記情報提供ノードは、少なくとも文書データ、画像データ、動画データ、または音声データのいずれかを備えることを特徴とする、請求項 1 に記載の開発支援システム。

【請求項 6】

前記ワークスペース作成部が、さらに、ユーザごとにワークスペースへのアクセス権限を付与する権限付与部を備えることを特徴とする、請求項 1 に記載の開発支援システム。

【請求項 7】

前記ワークスペース作成部が、さらに、前記ワークスペースにタグを付して管理するタグ管理部を備えることを特徴とする、請求項 1 に記載の開発支援システム。

【請求項 8】

前記ワークスペース作成部が、さらに、プロンプトノードおよび/またはエッジにユーザが決定するコストを設定するコスト設定部を備えることを特徴とする、請求項 1 に記載の開発支援システム。

【請求項 9】

10

20

30

40

50

コンピュータを  
ワークスペース作成手段、大規模言語モデル入出力手段、およびプロンプト実行手段、  
として機能させる開発支援プログラムであって、

前記ワークスペース作成手段は、  
ワークスペースを表示するワークスペース表示手段、  
ユーザの操作により、前記ワークスペース上にプロンプトノードを作成するノード作成  
手段、および、

一のプロンプトノードと他のプロンプトノードとを接続するエッジを作成するエッジ作  
成手段、  
を備え、

前記プロンプトノードは、  
大規模言語モデルの選択を受け付ける大規模言語モデル選択欄と、  
プロンプトを受け付けるプロンプト受付欄と、  
前記プロンプトに対する前記大規模言語モデルの出力結果を表示する出力結果表示欄と

実行ボタンと、  
再実行ボタンと、を含み、  
前記実行ボタンは、当該実行ボタンが押下されたプロンプトノードのみでプロンプトの  
送信と出力結果の受信を行うものであり、

また前記再実行ボタンは、最初のプロンプトノードから当該再実行ボタンが押下された  
プロンプトノードまでの各プロンプトノードにおいてプロンプトの送信と出力結果の受信  
を行うものであり、

前記大規模言語モデル入出力手段は、  
選択されている前記大規模言語モデルを取得する大規模言語モデル取得手段、  
前記プロンプトを前記大規模言語モデルに入力として提供するプロンプト提供手段、お  
よび、

前記プロンプトに対する前記大規模言語モデルの出力結果を取得する出力結果取得手段  
、を備え、

前記プロンプト実行手段は、  
ユーザが選択しているプロンプトノードを取得する選択ノード取得手段と、  
最初のプロンプトノードから前記ユーザが選択しているプロンプトノードまでのルート  
を取得するルート取得手段、および、

前記ルート上の各プロンプトノードについて、最初のプロンプトノードから順に前記大  
規模言語モデル入出力手段による大規模言語モデル入出力処理を実行する逐次入出力手段  
、を備えることを特徴とする、開発支援プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、開発支援システムおよび開発支援プログラムに関するものである。

【背景技術】

【0002】

いわゆる生成AIと言われる技術が広く活用されるようになってきている。専門家による  
利用に限らず、例えばChatGPT（登録商標）のように、広く一般による利用が可能になっ  
ている。

【0003】

生成AIを利用するソフトウェアの開発も進んでいる。ただし、生成AIを用いたソフトウ  
ェア開発等において、プロンプトとそのプロンプトに対する出力結果が多数存在する場合  
、問題となるプロンプトや最適なプロンプト（またはそれらに対する出力結果）を見つける  
ことは容易ではない。よって、プロンプトとプロンプトとの関係を見やすく管理する開発  
支援ツールが求められている。

10

20

30

40

50

## 【 0 0 0 4 】

特許文献 1 には、処理のまとまりを示すノードをオブジェクトとして表示し、オブジェクトに対するユーザの操作に基づいてプログラムにおけるノード構成を示すフロー定義を作成するプログラム開発支援装置が開示されている。

また、非特許文献 1 および非特許文献 2 には、大規模言語モデル (LLM) によって動作する AI アプリケーションの開発サイクル全体を合理化するために設計された開発ツールが開示されている。

さらに、非特許文献 3 には、ユーザの作成したエレメントをもとにさまざまなアクションを実行する AI 補助機能を備えるアプリケーションが開示されている。

## 【 先行技術文献 】

10

## 【 特許文献 】

## 【 0 0 0 5 】

【 特許文献 1 】 特許第 6 8 0 1 0 8 6 号公報

## 【 非特許文献 】

## 【 0 0 0 6 】

【 非特許文献 1 】 共同著作、“プロンプト フローの概要”、フロー作成ページ、[online]、[2024年2月24日検索]、インターネット < URL : <https://learn.microsoft.com/ja-jp/azure/machine-learning/prompt-flow/get-started-prompt-flow> >

【 非特許文献 2 】 共同著作、“Azure Machine Learning プロンプト フローとは”、[online]、[2024年2月24日検索]、インターネット < URL : <https://learn.microsoft.com/ja-jp/azure/machine-learning/prompt-flow/overview-what-is-prompt-flow> >

20

【 非特許文献 3 】 Goodpatch Customer Success | Asako、“Strap AI (Beta) のご利用方法について”、[online]、[2024年2月24日検索]、インターネット < URL : <https://goodpatch.zendesk.com/hc/ja/articles/17210796403481-Strap-AI-Beta-%E3%81%AE%E3%81%94%E5%88%A9%E7%94%A8%E6%96%B9%E6%B3%95%E3%81%AB%E3%81%A4%E3%81%84%E3%81%A6> >

## 【 0 0 0 7 】

しかしながら特許文献 1 は生成 AI (大規模言語モデル) に関するものではなく、また、非特許文献 1 および非特許文献 2 は、プログラムを可視化するものであって、可視化されたビューにプロンプトや出力結果が表示されるものではない。

また、非特許文献 3 も、テキストが入力された付箋やエレメントにおいて、プロンプトや、プロンプトに対する出力結果を一覧表示するものではない。

30

さらに、いずれの文献においても、ビューや付箋ごとに大規模言語モデルを選択できるものではない。

## 【 発明の概要 】

## 【 発明が解決しようとする課題 】

## 【 0 0 0 8 】

解決しようとする問題点は、ワークスペース上に処理単位 (プロンプトノード) とそれらをつなぐエッジとを自由に作成できる開発支援システムであって、ユーザは各プロンプトノードで使用したい大規模言語モデルを選択したうえでプロンプトの提供と出力結果の取得が可能であり、かつそれらのプロンプトおよび出力結果を一覧表示できる開発支援システムがない点である。

40

## 【 課題を解決するための手段 】

## 【 0 0 0 9 】

本発明の開発支援システムは、ワークスペース作成部および大規模言語モデル入出力部を備えることを最も主要な特徴とする。

ここで、当該ワークスペース作成部によりワークスペースに作成されるプロンプトノードは、大規模言語モデルの選択を受け付ける大規模言語モデル選択欄、プロンプトを受け付けるプロンプト受付欄、および前記プロンプトに対する前記大規模言語モデルの出力結果を表示する出力結果表示欄を備える。また、大規模言語モデル入出力部は、大規模言語モデルにプロンプトを入力として提供し、得られた出力を取得する。

50

## 【 0 0 1 0 】

本発明は、上記課題を鑑みてなされたものであり、例えば以下の手段を採用している。

すなわち、ワークスペース作成部および大規模言語モデル入出力部を備える開発支援システムであって、前記ワークスペース作成部は、

ワークスペースを表示するワークスペース表示部と、

ユーザの操作により、前記ワークスペース上にプロンプトノードを作成するノード作成部と、

一のプロンプトノードと他のプロンプトノードとを接続するエッジを作成するエッジ作成部と、を備え、

前記プロンプトノードは、

大規模言語モデルの選択を受け付ける大規模言語モデル選択欄と、

プロンプトを受け付けるプロンプト受付欄と、

前記プロンプトに対する前記大規模言語モデルの出力結果を表示する出力結果表示欄と、を含み、

前記大規模言語モデル入出力部は、

選択されている前記大規模言語モデルを取得する大規模言語モデル取得部と、

前記プロンプトを前記大規模言語モデルに入力として提供するプロンプト提供部と、

前記プロンプトに対する前記大規模言語モデルの出力結果を取得する出力結果取得部と、を備えることを特徴とする、開発支援システムを提供する。

## 【 発明の効果 】

## 【 0 0 1 1 】

本発明の開発支援システム1は、ワークスペースを複数人で共有できるため、例えば生成 AI を用いる開発等について複数のユーザでコラボレーションできるほか、ユーザは最適なプロンプトを容易に発見することに資することができ、さらに、プロンプトに対して細かなチューニングが可能という利点がある。

## 【 図面の簡単な説明 】

## 【 0 0 1 2 】

【 図 1 】 開発支援システム1がユーザ端末20に表示する画面（ワークスペース）を示す図である。

【 図 2 】 プロンプトノードを示す図である。

【 図 3 】 プロンプト（子ノード）新規作成画面を示す図である。

【 図 4 】 プロンプト（孫ノード）新規作成画面を示す図である。

【 図 5 】 エッジ作成画面を示す図である。

【 図 6 】 プロンプト複製（第一子ノードの複製）画面を示す図である。

【 図 7 】 プロンプト複製（第二子ノードの複製）画面を示す図である。

【 図 8 】 プロンプト新規作成・複製後のワークスペースを示す図である。

【 図 9 】 コメント表示欄を含むワークスペース（その 1）を示す図である。

【 図 1 0 】 コメント表示欄を含むワークスペース（その 2）を示す図である。

【 図 1 1 】 コメント表示欄を含むワークスペース（その 3）を示す図である。

【 図 1 2 】 ワークスペースを示す図である。

【 図 1 3 】 ワークスペース作成処理を示すフローチャートである。

【 図 1 4 】 大規模言語モデル入出力処理を示すフローチャートである。

【 図 1 5 】 プロンプト実行処理を示すフローチャートである。

【 図 1 6 】 開発支援システム1のネットワーク構成図を示す図である。

【 図 1 7 】 サーバ10のハードウェア構成図である。

【 図 1 8 】 ユーザ端末20のハードウェア構成図である。

## 【 発明を実施するための形態 】

## 【 0 0 1 3 】

本発明の実施形態を図面に基づいて説明する。以下の各実施形態では、同一又は対応する部分については同一の符号を付して説明を適宜省略する場合がある。

また、以下に用いる図面は本実施形態を説明するために用いるものであり、実際の装置の構成やユーザーインターフェース（UI）、データ構造などとは異なる場合がある。

#### 【0014】

（実施形態の概要）

本実施形態の概要について、図1を用いて説明する。

図1は、本実施形態の開発支援プログラムP1による処理を行うシステム（以下「開発支援システム1」とする。）がユーザ端末20に表示する画面（ワークスペース）の一例である。

#### 【0015】

ユーザは、ワークスペース上にプロンプトノードを作成することができる。

プロンプトノードは、大規模言語モデルの選択を受け付ける大規模言語モデル選択欄、プロンプトを受け付けるプロンプト受付欄、および（プロンプトに対する）大規模言語モデルからの出力結果を表示する出力結果表示欄を備える。

#### 【0016】

ユーザが規模言語モデル受付部で大規模言語モデルを選択し、プロンプト受付欄にプロンプトを入力して実行ボタンを押下すると、プロセッサは選択されている大規模言語モデルを取得してプロンプトを前記大規模言語モデルに入力として提供し、また、前記プロンプトに対する前記大規模言語モデルの出力結果を取得する処理（大規模言語モデル入出力処理）を行う。またプロセッサは、この出力結果を出力結果表示欄に表示する。

#### 【0017】

各プロンプトノードはエッジにより連結され、プロセッサはプロンプトノードとプロンプトノードの前後関係（世代関係）を把握する。

ユーザが後の世代（例えば親ノードに対する子ノード）にあるプロンプトノードについて再実行ボタンを押下すると、プロセッサは（ユーザの任意により、あるいはあらかじめ定められた）最初のプロンプトノードからそのプロンプトノードまでのルート（経路）を取得し、当該ルート上の各プロンプトノードについて、最初から順に上述した大規模言語モデル入出力処理を行う（逐次入出力処理）。

#### 【0018】

このほか、ユーザは任意のプロンプトノードにコメントを残すことができる。

#### 【0019】

（実施形態の詳細）

以下、本実施形態に係る開発支援システム1について、詳細を説明する。開発支援システム1は、開発支援プログラムP1を備えるコンピュータを含み、ユーザに対し、生成AIを用いる開発の管理が可能なシステムをオンラインで提供する。

すなわち開発支援システム1は、開発支援プログラムP1による情報処理が、ハードウェア資源を用いて具体的に実現されるものである。

以下、開発支援システム1を構成する、1．ユーザーインターフェース、2．プログラム処理、3．データ、および4．ハードウェア構成、について順に説明する。

#### 【0020】

（用語の定義）

ここで、いくつか言葉の定義を行う。

「生成AI」は、与えられた情報をもとにテキストや画像などを生成することができる人工知能（AI・Artificial Intelligence）である。生成AIは大規模言語モデルを含む。

「大規模言語モデル」は、言語モデルのうち、「計算量」、「データ量」、「パラメータ数」を大規模にして構築する自然言語モデルである。簡単のため、大規模言語モデルをLLM（Large Language Model）と表記する場合がある。

本実施形態において、大規模言語モデルは、文書ファイル、画像ファイル、または音声ファイルといった各種ファイルなど、テキスト以外を入力または出力するものであってもよい。

以下に記載している課題を達成できるものであれば特に制限はないが、データ量として

10

20

30

40

50

10億語を超えるデータを扱う大規模言語モデルが知られている（例えばBERTなど）。また、1億以上のパラメータを備える大規模言語モデルが多く知られており、さらに、1000億以上のパラメータを持つものも存在する（例えばGPT-3など）。

大規模言語モデルの具体例として例えば、BERT、GPT (Generative Pretrained Transformer) -3 (登録商標)、GPT-4 (登録商標)、PaLM (Pathways Language Model) (登録商標)、LLaMA (Large Language Model Meta AI) またはNEMO LLMなどが挙げられる。

「ワークスペース」は、プロセッサが次項のノードやエッジを作成・描画するコンピュータ上のスペースである。ホワイトボードやキャンバスともいう。ワークスペースには、プロンプトノードやエッジを描画するスペースのほか、コメント欄も含む。

「ノード」は、ワークスペース上に表示される機能単位である。ブロックやオブジェクトと称される場合もある。本実施形態において、プロンプトノードはワークスペース上でボックス型の形状を有する。

「エッジ」は、ワークスペース上でプロンプトノードとプロンプトノードをつなぐ機能単位である。本実施形態において、エッジはワークスペース上で線形状を有する。

#### 【0021】

「プロンプトの提供（プロンプトを大規模言語モデルに入力として提供）」は、プロンプト受付欄に入力されているプロンプトを大規模言語モデルの入力として提供することであり、大規模言語モデルのAPIへの送信を含む。また、プロンプトの提供は、プロンプト受付欄に入力されているプロンプトを、そのまま提供する方法に限るものではなく、プロセッサ122などがプロンプト受付欄に入力されているプロンプトを加工等（例えば前処理など）をしたうえで提供するものであってもよい。

「出力結果」は、大規模言語モデルによる出力の全部または一部を含む結果である。大規模言語モデルによる出力そのものである必要はなく、サーバ10または生成AIサーバ30において加工等されていてもよい。

「大規模言語モデル入出力処理」は、選択されている大規模言語モデルを取得して、プロンプトを前記大規模言語モデルに入力として提供し、また、前記プロンプトに対する前記大規模言語モデルの出力結果を取得する処理である。

#### 【0022】

上記のほか、ノードの世代を表す用語として「親ノード」、「子ノード」、「孫ノード」といった用語を用いることがある。ある世代（第一世代）のノードを親ノードとすると、次の世代（第二世代）のノードを子ノードと、さらにその次の世代（第三世代）のノードを孫ノードと称する。

また、例えば第一世代、第二世代、第三世代のノードがある場合に、第一世代側のノードを古いノードあるいは上位ノードと、第三世代側のノードを新しいノードまたは下位ノードと表現する場合がある。

#### 【0023】

以下において、「」処理と記載している場合、コンピュータのプロセッサは、プログラム格納部に記憶されている「」プログラムに基づく処理を実行することを意味する。本段落において、「」の箇所には同じ語が入る。

すなわち、「」プログラムは、「」処理の実行により、コンピュータを「」手段として機能させるプログラムである。またこの際、当該プロセッサを備える制御部（コンピュータ）は、「」部（または「」装置）としても機能することを意味する。

この場合において、「」部は、「」プログラムに基づく「」処理を実行することを意味する。

#### 【0024】

例えば、開発支援プログラムP1は、開発支援処理の実行により、コンピュータを開発支援手段として機能させるプログラムである。またこの際、プロセッサ122を備えるコンピュータの制御部12は、開発支援部130（または開発支援装置）として機能する。

#### 【0025】

10

20

30

40

50

開発支援システム1において、ユーザ端末20などの各端末（コンピュータ）はそれぞれプロセッサを備えるが、単にプロセッサという場合は、開発支援プログラムP1により処理を行うプロセッサ、本実施形態ではサーバ10のプロセッサ122、を指すものとする。

【0026】

なお、サーバ10が開発支援処理を実行する場合、プロセッサとはサーバ10のプロセッサ122を指すが、開発支援システム1がネットワークを使用せず、ユーザ端末20で完結する場合、プロセッサとはユーザ端末20のプロセッサ222を指すものとする。

【0027】

1. ユーザーインターフェース（User Interface（UI））

まず、本実施形態の開発支援システム1がユーザ端末20に表示させるインターフェースについて、図を用いて説明する。

10

以降で説明するインターフェースは、プロセッサ122がユーザ端末20のブラウザに表示させるものを簡略化したものである。

【0028】

また、説明に必要な機能に関わるアイコン等のみ表示することとし、それ以外の公知のアイコンなどは省略する。

【0029】

なお以下において、簡単のため、「サーバ10のプロセッサ122が、ユーザ端末20からのリクエストを受けて、当該端末のブラウザに表示するためのデータを返す」ことを、「プロセッサ122が端末のブラウザに表示する（させる）」または「プロセッサ122が表示する（させる）」などと記載する場合がある。

20

また同様に、「サーバ10のプロセッサ122が、記憶部14のデータ記憶部14bにデータを保存させる」ことを、「プロセッサが（データを）保存する（させる）」などと記載する場合がある。

【0030】

図2は、プロンプトノードを示す図である。

本実施形態のプロンプトノードUI-10Pは、図2に示すようなプロンプト入力フォームを備える。

【0031】

図2に示すように、プロセッサ122は、プロンプトノードUI-10Pにプロンプト名入力欄UI-101、大規模言語モデル選択欄UI-102、プロンプト受付欄UI-103、処理ボタンUI-104、変数ボタンUI-105、変数追加ボタンUI-106、出力結果表示欄UI-107、新規ボタンUI-108、および複製ボタンUI-109、を表示する。

30

【0032】

プロンプト名入力欄UI-101は、ユーザが当該プロンプトノードのプロンプト名（ノード名）などを入力するためのテキストボックスである。管理に便利なプロンプト名として例えば、「プロンプト1」、「プロンプト2-1」、「プロンプト2-2」、などが挙げられる。

【0033】

大規模言語モデル選択欄UI-102は、ユーザがそのプロンプトノードで選択する大規模言語モデルを選択するためのものである。本実施形態において、プロセッサ122は大規模言語モデルとして例えばGPT-3（登録商標）、GPT-4（登録商標）、PaLM（登録商標）などをプルダウンで表示する。

40

【0034】

ここで、各プロンプトノードが大規模言語モデル選択欄UI-102を備えることにより、プロンプトノードごとに異なる大規模言語モデルを使用し、検討することができる。

つまり、プロンプトノードごとに最適な大規模言語モデルを選択することができ、フレキシブルなプロンプト作成が可能になることにより、各プロンプトノードにおける出力結果をブラッシュアップすることができる。その結果、一連のプロンプトノードによる処理で最終的に得られる結果が、改善しやすくなるという利点がある。

【0035】

50



プロンプト受付欄UI-103は、生成AIに対するプロンプトをユーザが入力するためのテキストボックスである。

【0036】

処理ボタンUI-104は、そのプロンプトノードにおける処理を実行するためのボタンである。

図2に示すように、プロセッサ122は処理ボタンUI-104として、(図2左から)コメントボタン、実行ボタン、再実行ボタン、および削除ボタンを表示する。

【0037】

コメントボタンは、ユーザがコメントを作成するためのボタンである。ユーザはコメントを後述するコメント表示欄UI-30に記述することができる。

10

【0038】

実行ボタンおよび再実行ボタンはどちらも、プロンプト受付欄UI-103に入力されているプロンプトを、(選択されている)大規模言語モデルに送信するためのボタンである。

つまり、本実施形態のプロセッサ122は、ユーザによる実行ボタンまたは再実行ボタンの押下を受け付けて、プロンプト受付欄UI-103に記載されているプロンプトを大規模言語モデル選択欄UI-102で選択されている大規模言語モデルのAPIに送信する。プロセッサ122は、当該プロンプトに対する大規模言語モデルからの出力結果を後述する出力結果表示欄UI-107に表示する。

【0039】

実行ボタンと再実行ボタンの違いは、プロンプトノードが複数連なっている場合において、上位プロンプトノードを変化させるか、変化させないかの違いである。

20

例えば、実行ボタンは、実行ボタンが押下されたプロンプトノードのみでプロンプトの送信と出力結果の受信を行う。

再実行ボタンは、最初のプロンプトノードから再実行ボタンが押下されたプロンプトノードまでの各プロンプトノードにおいてプロンプトの送信と出力結果の受信を行う。再実行ボタンは後述するプロンプト実行処理に関わる。

【0040】

削除ボタンは、その削除ボタンを備えるプロンプトノードを削除するボタンである。

【0041】

変数ボタンUI-105は、変数の設定を行うためのボタンである。

30

ここで、変数について説明する。

【0042】

プロンプト受付欄UI-103に中括弧(「{」および「}」)で囲まれている語句がある場合、プロセッサ122は、その語句をそのプロンプトノードにおける変数として取得し、保存する。

例えば、プロンプト受付欄UI-103に記載されているプロンプトが、「原材料として{原材料名1}と{原材料名2}を使用する西洋料理を教えてください。また、{原材料名1}の前処理方法を教えてください。」だったとする。また、ユーザは変数1に対応する語句として「ひよこ豆」を、変数2に対応する語句として「レンズ豆」を入力することができる(不図示)。

40

このとき、プロセッサ122は、最初の中括弧で囲まれた文字列を変数1とし、変数1(原材料名1) = 「ひよこ豆」として記憶する。同様に、2番目の中括弧で囲まれた文字列を変数2とし、変数2(原材料名2) = 「レンズ豆」として記憶する。

なおプロセッサ122は、中括弧で囲まれた文字列が同じものであれば同じ変数であるとして認識する。

【0043】

またこの例において、ユーザが図2に示す変数1ボタンを押下すると、プロセッサ122は「ひよこ豆」と入力されているテキストボックスを表示する。

ユーザは、このテキストボックスを編集することができる。テキストの内容を「ひよこ豆」から「米」に変えた場合、変数1の内容が「米」となる。

50

つまり、上記プロンプトは「原材料として{米}と{レンズ豆}を使用する西洋料理を教えてください。また、{米}の前処理方法を教えてください。」となる。

【0044】

変数追加ボタンUI-106は、所定個数（図2では4個）以上の変数を設定した場合に、不足している変数ボタンUI-105を追加するためのボタンである。

【0045】

出力結果表示欄UI-107は、プロンプト受付欄UI-103に入力されているプロンプトに対する大規模言語モデルからの出力結果を表示する表示部である。

【0046】

新規ボタンUI-108は、現在のプロンプトノードの（ワークスペース表示上の）下側に新たなプロンプトノードを追加するボタンである。

つまり、現在のプロンプトノードを親ノード、新たなプロンプトノードを子ノードとすると、ユーザによる新規ボタンUI-108の入力を受け付けて、プロセッサ122は、新しいプロンプトノード（子ノード）をワークスペースに表示し、現在のプロンプトノード（親ノード）と、作成した新しいプロンプトノード（子ノード）とをエッジUI-20で結ぶ。具体例は後述する。

ここでプロセッサ122は、新しいプロンプトノード（子ノード）として、大規模言語モデル選択欄UI-102で選択されている大規模言語モデルのみを引き継いだプロンプトノードを作成する。ただしこの大規模言語モデルは変更可能である（情報を引き継いだ場合に変更可能な点は以下も同様とする）。

【0047】

複製ボタンUI-109は、現在のプロンプトノードを複製するボタンである。

つまり、現在のプロンプトノードを第一の子ノード（第一子ノード）、新たなプロンプトノードを第二の子ノード（第二子ノード）とすると、ユーザによる複製ボタンUI-109の入力を受け付けて、プロセッサ122は、同じ親ノードにつながる新たなプロンプトノード（第二子ノード）をワークスペースに表示し、当該親ノードと、作成した新しいプロンプトノード（第二子ノード）とをエッジUI-20で結ぶ。具体例は後述する。

ここでプロセッサ122は、新しいプロンプトノードとして、大規模言語モデル選択欄UI-102で選択されている大規模言語モデルと、プロンプト受付欄UI-103に記載されているプロンプトを引き継いだプロンプトノードを作成する（出力結果表示欄UI-107は空欄になる）。

【0048】

なお、図2ではプロンプト受付欄UI-103および出力結果表示欄UI-107としてテキストボックスを例示したが、これに限られない。

例えば、プロンプト受付欄UI-103は、テキスト以外、例えば画像ファイルや音声ファイルなど、を入力とできるようにしてもよい。同様に、出力結果表示欄UI-107はテキスト以外、例えば画像ファイルや音声ファイルなど、を表示できるようにしてもよい。

【0049】

図3は、プロンプト（子ノード）新規作成画面を示す図である。

図3に示すように、ユーザが親ノードUI-11の新規ボタンを押下すると、プロセッサはワークスペース上において、親ノードUI-11の下側に子ノードUI-12を作成し、親ノードUI-11と子ノードUI-12をエッジUI-20で結ぶ。

【0050】

ここで、ワークスペース上のノードの位置関係について説明する。

原則として、エッジにより直接結合していない2つのノードがある場合、ワークスペース上で上側にあるノードが上位ノード、下側にあるノードが下位ノードとしてプロセッサ122は認識する。

【0051】

図4は、プロンプト（孫プロンプトノード）新規作成画面を示す図である。

図3の場合と同様に、ユーザが子ノードUI-12の新規ボタンを押下すると、プロセッサ1

10

20

30

40

50

22はワークスペース上において、子ノードUI-12の下側に孫ノードUI-13を作成し、子ノードUI-12と孫ノードUI-13をエッジUI-20で結ぶ。

【 0 0 5 2 】

なお、親ノード、子ノード、孫ノードの呼称は説明の便宜のための表現であり、例えばあるノードが常に親ノードになるというわけではない。

例えば、図3において、UI-12のノードはUI-11のノードの子ノードであるが、それと同時に、UI-12のノードはUI-13のノードの親ノードである。

【 0 0 5 3 】

図5は、エッジ作成画面を示す図である。

ユーザは、新規ボタンUI-108や後述する複製ボタンUI-109の押下時に限らず、ノードUI-10とノードUI-10とを結ぶエッジUI-20を作成することができる。

例えば図5において、ユーザはマウス操作またはタッチ操作などにより、孫ノードUI-13と親ノードUI-11をエッジUI-20で結合させている。

【 0 0 5 4 】

本実施形態において、より具体的には、ドラッグアンドドロップ様の操作において、最初に押下した場所のノードを親ノード、押下状態を維持してカーソルを動かし、押下状態を話した場所のノードを子ノードとする。

【 0 0 5 5 】

ここで、ノードUI-10とノードUI-10とをエッジUI-20で結合することにより、プロセッサ122はこれらのノードに関係があり、上位ノードや下位ノードを認識する。

例えば、後述の3.データの項で説明するエッジテーブルD20には、エッジが結合するノードの情報が保存されている。

【 0 0 5 6 】

図6は、プロンプト複製（第一子ノードの複製）画面を示す図である。

図6に示すように、ユーザが子ノードUI-12aの複製ボタンを押下すると、プロセッサはワークスペース上において、親ノードUI-11の下側に新しい子ノードUI-12bを作成し、親ノードUI-11と子ノードUI-12bをエッジUI-20で結ぶ。

【 0 0 5 7 】

図7は、プロンプト複製（第二子ノードの複製）画面を示す図である。

図6の場合と同様に、ユーザが子ノード（第二子ノード）UI-12bの複製ボタンを押下すると、プロセッサはワークスペース上において、親ノードUI-11の下側に新しい子ノード（第三子ノード）UI-12cを作成し、親ノードUI-11と子ノード（第三子ノード）UI-12cをエッジUI-20で結ぶ。

【 0 0 5 8 】

プロンプトノードを複製できることにより、例えば、親ノードのプロンプトに対して得られた出力結果に際し、さらに複数のプロンプトを検討したい場合において、複数のプロンプトとその結果を並べて表示することができるため、それらの複数のプロンプトのうち特にどれが有効であったかを比較検討することが容易になるという利点がある。

【 0 0 5 9 】

図8は、プロンプト新規作成・複製後のワークスペースを示す図である。

上述した新規ボタンUI-108や複製ボタンUI-109を操作することにより、ユーザはワークスペース上に複数のプロンプトとそのプロンプトに対する出力結果を一覧表示することができる。

ツリー状に形成されるこれらのプロンプトノードを比較検討することにより、ユーザはよりよいプロンプトを探索することができる。

特に、複数のユーザ間でひとつのワークスペースを共有できるため、複数の開発者目線で開発の方向性を確認することができるという利点がある。

【 0 0 6 0 】

図9は、コメント表示欄を含むワークスペース（その1）を示す図である。

図9に示すように、プロセッサ122はワークスペース上にコメント表示欄UI-30を表示す

10

20

30

40

50

ることができる。

ユーザはいずれかのプロンプトノード10Pのコメントボタンを押すことで、コメント表示欄UI-30にコメントを残すことができる。

【0061】

図9の例では、第一のユーザ（ユーザ名1）がプロンプト1のコメントボタンを押下し、コメントUI-31を残している。

コメントUI-31はコメントのほか、コメントがなされたノード名（プロンプト名）、コメントがなされた日付、およびコメントしたユーザ名を表示する。

【0062】

図10は、コメント表示欄を含むワークスペース（その2）を示す図である。

10

ここでは、第二のユーザ（ユーザ名2）がプロンプト2のコメントボタンを押下し、コメントを残している。

この例に示すように、コメント表示欄UI-30は複数のユーザがそれぞれの端末上で確認することで、またコメントを残すことができる。

【0063】

図11は、コメント表示欄を含むワークスペース（その3）を示す図である。

ここでは、図10の例に続き、第一のユーザ（ユーザ名1）がプロンプト2のコメントボタンを押下し、コメントを残している。

例えば、図10で第二のユーザがプロンプト2に対する疑問を呈し、図11で第一のユーザがその疑問に対する回答を行う、といった内容を示すものである。

20

【0064】

なお、ワークスペースへの書き込み（ノードやエッジの追加、プロンプトの実行、コメントの追加など）は各ユーザが自由に行うことができる。

他のユーザが書き込みをしている際のロック（共有ロックや占有ロック）やそれに関連する技術（更新など）については、公知の方法を好適に用いることができる。

【0065】

（フローの実行）

図12は、ワークスペースを示す図である。

ここでは、本実施形態の開発支援システム1の応用例を示す。

【0066】

30

ユーザが所定のプロンプトノードを選択し、そのプロンプトノードの再実行ボタンを押下すると、プロセッサ122は、最初のプロンプトノードから当該プロンプトノードまでのそれぞれのプロンプトノードにおいて処理（大規模言語モデル入出力処理）を実行する。

大規模言語モデル入出力処理において、プロセッサ122は、大規模言語モデル選択欄UI-102で選択されている大規模言語モデルに対してプロンプト受付欄UI-103に入力されているプロンプトを提供し、得られた出力結果を出力結果表示欄UI-107に表示する。

また、この最初のプロンプトノードから当該プロンプトノードまでのそれぞれのプロンプトノードにおいて大規模言語モデル入出力処理を実行する一連の処理（ひとまとまりの処理）を逐次入出力処理とする。

【0067】

40

図12の例において、ユーザはUI-14aの再実行ボタンを押下している。これにより、プロセッサ122は、最初のプロンプトノード（UI-11）から選択されているプロンプトノード（UI-14a）までのルートを取得する。

本実施形態において、ルート上のプロンプトノードはU-11、UI-12a、UI-13a、そしてUI-14aである。そして、プロセッサ122はそのルート上にある各プロンプトノードで順に大規模言語モデル入出力処理を実行する。

【0068】

なお、図12の例において、UI-11からUI-14bに至るルートは2通りある（UI-11 UI-12 a UI-13a UI-14 b と、UI-11 UI-12b UI-14 b ）。

ユーザが（UI-14aではなく）UI-14bの再実行ボタンを押下した場合、本実施形態のプロ

50

セッサ122はUI-14bに至るルートを計算し、2つのルートを見出す。そしてプロセッサ122は2つのルートについて、例えば色のついた線などで表示する。ユーザがそのうちの一方を選ぶことで、プロセッサ122は当該ルートの各プロンプトノードについて大規模言語モデル入出力処理を実行する。

【 0 0 6 9 】

以下、図 1 2 の各プロンプトノードにおけるプロンプトおよびそのプロンプトに対する出力結果の具体例を挙げて説明する。

なお以下のプロンプトに対する出力結果は例であるため、実際のものとは異なる場合がある。

また以下において、変数部分（中括弧）は変数に置換される文字列を記載している。例えば「原材料 1」ではなく「オリゴ糖」と記載している。

【 0 0 7 0 】

UI-11のプロンプトは以下である。

「 {オリゴ糖} を原材料に含む日本での通販専用美容食品を考えてください。STEP単位でその役割として指示通りに出力しそれ以上のことは行わないでください。

STEP1

- ・あなたはプロの研究開発者です。
- ・インターネット検索を行い、最新情報を取得しながら {オリゴ糖} の効果効能を洗い出してください。

# ルール

- ・日本で効果を謳える効果効能であること。
- ・消費者の悩みを解決できる効果効能であること。」

【 0 0 7 1 】

UI-11のプロンプトに対する出力結果は例えば以下のようなものになる。

「オリゴ糖の主な効果には、腸内環境の改善、便秘の改善、血糖値の上昇を抑制すること、免疫力の向上、虫歯予防などがあります。

具体的には、キシロビオースやキシロトリオースなどのオリゴ糖は、ビフィズス菌の増殖を活性化させることで腸内環境の改善に役立つとされています。また、ラクチュロースオリゴ糖、ガラクトオリゴ糖などは、特に砂糖に比べてカロリーが低く、味の変化を抑えつつ甘みを提供できるため、ダイエット中の方や健康を気にする方にも適しています。

オリゴ糖を利用した日本での通販専用商品として例えば、腸内環境を改善する健康食品、低カロリーなスイーツやドリンク、虫歯予防を特長とするキャンディーやガムなどが考えられます。」

【 0 0 7 2 】

UI-12aのプロンプトは以下である。

「STEP2-1

- ・あなたはプロのマーケターです。
- ・STEP1で導出した効果効能が訴求可能なターゲットを考えてください。
- ・商品コンセプトを3つ考えてください。

# ルール

- ・日本の市場規模で見ても、比較的ニーズが大きい市場であること
- ・競合と違うアプローチであることが望ましい

# 出力形式

- ・商品コンセプト：
- ・ターゲット：
- ・ターゲットの抱えるインサイト：」

【 0 0 7 3 】

UI-12aのプロンプトに対する出力結果は例えば以下のようなものになる。

「商品コンセプト 1：健康志向スイーツ - オリゴスマート

ターゲット：健康を意識する20代から40代の男女。甘いものを好むが、カロリーや健康へ

10

20

30

40

50

の影響を気にしている人々。

ターゲットの抱えるインサイト：甘いものを楽しみたいが、砂糖の摂取が気になる。また、腸内環境や免疫力向上に興味があり、健康と美味しさを兼ね備えたスイーツを求めている。

商品コンセプト2：腸活サポート飲料 - ビフィズブレンド

ターゲット：腸内環境に悩む全年代の男女。便秘や腸の健康に関心が高い人々。

ターゲットの抱えるインサイト：日常的に腸内環境の改善を図りたいが、忙しさから十分な食生活ができない。手軽に腸活ができる美味しい飲料を求めている。

商品コンセプト3：「免疫力アップのためのオリゴ糖サプリメント - イミュノオリゴ」

ターゲット：免疫力の強化に興味がある中高年の男女。特に風邪やインフルエンザなどを予防したいと考えている人々。

ターゲットの抱えるインサイト：年齢と共に低下する免疫力に不安を感じており、日常的に摂取できる健康食品を探している。特に自然由来の成分に興味がある。

これらのコンセプトは、日本の市場で健康志向が高まっている現状を踏まえ、比較的ニーズが大きいと考えられる市場に対応しています。

#### 【0074】

UI-12bのプロンプトは以下である。プロンプトに対する出力結果は省略する。

「STEP2-2

- ・あなたはプロのマーケターです。
- ・STEP1で導出した効果効能が訴求可能なターゲットの年齢層を教えてください。
- ・商品コンセプトを1つ考えてください。

# ルール

- ・日本の市場規模で見て、比較的ニーズが大きい市場であること
- ・ヨーロッパ地域の健康志向ユーザに人気のある商品を参照すること

# 出力形式

- ・商品コンセプト：
- ・ターゲット年齢層：
- ・ターゲットの抱えるインサイト：
- ・ヨーロッパ地域の健康志向ユーザに人気のある商品とその生産者の属する国名」

#### 【0075】

UI-13aのプロンプトは以下である。

「STEP3

- ・あなたはプロのリサーチャーです。
- ・インターネットを検索してコンセプトごとに競合分析をしてください。

# ルール

- ・関連キーワードで検索をした際に上位表示される商品は競合となる
- ・目的は競合に対してのバリュープロポジションを発見すること
- ・コンセプトごとに3つ以上の競合をリストアップすること

# 出力形式

- ・商品名
- ・価格

#### 【0076】

UI-13aのプロンプトに対する出力結果は例えば以下のようなものになる。

「1. 商品コンセプト1の商品（健康志向スイーツ - オリゴスマート）について

(1) 競合分析：

- ・競合1： カヌレ（3個入り）

価格： 円

- ・競合2： チョコレートケーキ

価格： 円

- ・競合3： ダイエットスイーツ

10

20

30

40

50

価格： 円

(2) 分析結果：

バリュープロポジション：オリゴスマートは、具体的な健康効果(例えば、オリゴ糖の腸内環境改善効果)を明確に打ち出し、他の低糖質スイーツと差別化することが可能です。

2. 商品コンセプト2の商品(腸活サポート飲料 - ビフィズブレンド)について(略)」

【0077】

UI-14aのプロンプトは以下である。プロンプトに対する出力結果は省略するが、大規模言語モデルは出力としてランディングページ(htmlファイルなど)を出力する。

「STEP4-1

10

- ・あなたはプロのWebマーケターです。
- ・STEP3までの情報を考慮し、ランディングページ(LP)を作成してください。

# ルール

- ・訪問者の関心を引きつけるデザインとコピーが必要
- ・ストーリー性やバリュープロポジションを強調するアプローチが望ましい

# 出力形式

1. 訪問者の興味を引き、下にスクロールしてもらうためのファーストビュー画像、キャッチコピー、購入ボタン、権威性を示す内容を必ず含めること

2. ターゲットの悩みと気持ちの代弁

ユーザの悩みに共感する形で言葉を作成する

20

3. ターゲットの具体的なメリット・ベネフィット

製品の特徴を何個かリストアップして可能な限り分かりやすく作成する

4. 購入ボタン(Call To Actionボタン)

消費者が今回購入しようと思う具体的なオファー(「今だけ 円引き」等)を含める」

【0078】

UI-14aのプロンプトに対する出力結果は例えば具体的なウェブページであり、上から順に上記1.ファーストビューから5.購入ボタンまでの内容作成されている。

【0079】

UI-14bのプロンプトは以下である。プロンプトに対する出力結果は省略する。

「STEP4-2

30

- ・あなたはプロのSEOライターです。
- ・STEP3までの情報を考慮し、ランディングページ(LP)を作成してください。

# ルール

- ・ユーザの関心を引きつけるデザインとコピーが必要
- ・ストーリー性やバリュープロポジションを強調するアプローチが望ましい

# 出力形式

訪問者の興味を引くためのファーストビューのほか、下記の内容を含むコンテンツを訪問者の興味を引くのに効果的と考える順序で表示すること

・ターゲットの悩みと気持ちの代弁

・ターゲットの具体的なメリット・ベネフィット

40

・STEP1で導出した効果効能

・お客さまの声

・購入ボタン(Call To Actionボタン)

【0080】

以上に示すように、ユーザはワークスペース上に自由にプロンプトノードとエッジを作成することができ、また、あるプロンプトノードについて複数のプロンプトノードを分岐させ、それぞれ検討することができる。

各プロンプトノードは、大規模言語モデルに対するプロンプトとその出力結果を保持することができ、表示することができる。これにより、各プロンプトノードの詳細について、ユーザは容易に確認することができる。

50

また、あるプロンプトノードは、エッジで繋がっている前の世代のプロンプトノードの情報を取得し、プロンプトに反映させることができる。

これにより、ユーザは一連の長大なプロンプトを作成する必要がなく、各プロンプトノードを検証し、変更が必要なプロンプトノードのプロンプトのみを変更・更新して、全体のプロンプトの流れを検証することができる。

これによりユーザは、どのプロンプトノードがカギとなるプロンプトを有しているのか、容易に把握することができ、所望の出力結果を得ることができる。

#### 【0081】

以上のような構成により、ユーザはGUI (Graphical User Interface) により直感的な操作でノードとエッジを作成でき、プロンプトノードの樹形図等を作成することができる。

10

ユーザは各プロンプトノードにおいて、使用している大規模言語モデル、プロンプト、およびその出力結果を確認することができる。また、すべてのプロンプトノードを一覧表示できるためプロンプトノードとプロンプトノードの関連性の確認や、比較等が可能である。

ユーザーインターフェースを複数のユーザで共有することができるため、開発を複数人でコラボレーションして実施できるほか、最適なプロンプトを容易に発見することができる。また、各開発者が、プロンプトに対して細かなチューニングを行うことができる。

#### 【0082】

### 2. プログラム処理

20

#### < 開発支援処理 >

本実施形態の開発支援システム1において行われるプログラム処理について説明する。

#### 【0083】

本実施形態において、プロセッサ122は、開発支援プログラムP1に基づき、開発支援処理を行う。

開発支援プログラムP1は、少なくともワークスペース作成プログラムP12、大規模言語モデル入出力プログラムP14、プロンプト実行プログラムP16、およびコメント作成プログラムP18を含み、プロセッサ122はこれらの各プログラムに基づいて、ワークスペース作成処理、大規模言語モデル入出力処理、プロンプト実行処理、およびコメント作成処理をそれぞれ実行する。

30

#### 【0084】

#### < 2 - 1 . ワークスペース作成処理 >

ワークスペース作成処理において、プロセッサ122は、ワークスペースを作成し、また、ユーザの操作により、ワークスペースにノードまたはエッジを作成 (描画) する。

#### 【0085】

プロセッサ122は、ワークスペース作成プログラムP12に基づき、ワークスペース作成処理を行う。

すなわち、ワークスペース作成プログラムP12は、プロセッサ122によるワークスペース作成処理の実行により、コンピュータをワークスペース作成手段として機能させる (制御部12はワークスペース作成部131として機能する)。

40

#### 【0086】

図13は、ワークスペース作成処理を示すフローチャートである。

プロセッサ122は、開発支援プログラムP1を含むアプリケーションソフトウェアの起動により、ワークスペース作成処理を開始する。

#### 【0087】

まずプロセッサ122は、ワークスペースを表示する (ステップ1)。

つづいてプロセッサ122は、ユーザによる画面要素の作成操作を待機する (ステップ2)。ここで画面要素とはノード、エッジまたはコメントである。

#### 【0088】

ステップ3の画面要素作成について説明する。

50



画面要素がプロンプトノードであれば、画面要素作成はプロンプトノードの新規作成かプロンプトノードの複製である。プロンプトノード作成操作を受け、プロセッサ122はワークスペース上にプロンプトノードを作成する（プロンプトノード作成処理）。

なお、プロンプトノードの作成や描画のための処理には、公知の方法を適宜用いてよい。

#### 【0089】

ここで、ユーザがワークスペース上で所定の操作をする（例えばダブルクリックや、右クリックで「プロンプトノードの新規作成」を選択する）と、プロセッサ122はワークスペース上に新たなプロンプトノードを作成する。

すでにプロンプトノードがある場合は、ユーザが新規ボタンUI-108や複製ボタンUI-109を押下することによっても、プロセッサ122は新たなプロンプトノードを作成する。

10

#### 【0090】

本実施形態のプロンプトノード作成処理において、プロセッサ122は、プロンプトノードUI-10Pにプロンプト名入力欄UI-101、大規模言語モデル選択欄UI-102、プロンプト受付欄UI-103、処理ボタンUI-104、変数ボタンUI-105、変数追加ボタンUI-106、出力結果表示欄UI-107、新規ボタンUI-108、および複製ボタンUI-109を作成して表示する。また、プロセッサ122は作成したプロンプトノードの情報を保存する。

#### 【0091】

つまり、プロンプトノード作成手段は、

- ・プロンプト名入力欄UI-101を作成して表示するプロンプト名入力欄作成手段、
  - ・大規模言語モデル選択欄UI-102を作成して表示する大規模言語モデル選択欄作成手段、
  - ・プロンプト受付欄UI-103を作成して表示するプロンプト受付欄作成手段、および、
  - ・出力結果表示欄UI-107を作成して表示する出力結果表示欄作成手段、
- を備える。

20

またこのとき、プロンプトノード作成処理を実行する制御部12は、上記に対応して、プロンプト名入力欄作成部、大規模言語モデル選択欄作成部、プロンプト受付欄作成部、出力結果表示欄作成部、として機能する。

#### 【0092】

図13に戻り、画面要素がエッジであれば、画面要素作成はエッジの作成である。エッジ作成操作を受け、プロセッサ122はワークスペース上にエッジを作成する（エッジ作成処理）。

30

なお、エッジの作成や描画のための処理には、公知の方法を適宜用いてよい。

#### 【0093】

ここで、ユーザがワークスペース上で所定の操作（例えば、右クリックで「エッジの新規作成」の選択）をし、親ノードと子ノードを選択することで、プロセッサ122はワークスペース上に新たなエッジを作成する。

例えば上述したように、ドラッグアンドドロップ操作で、選択操作（左クリックなど）をしたところが親ノード、選択状態を維持し、選択状態を解除（左クリックを離すなど）したところが子ノードとなる。

ユーザがあるプロンプトノードの新規ボタンUI-108や複製ボタンUI-109を押下することによっても、プロセッサ122は新たなエッジを作成する。

40

#### 【0094】

ユーザは、プロンプトノードが既にワークスペース上にある場合にのみエッジを作成することができる。

またプロセッサ122は、どちらのプロンプトノードが親ノードで、どちらのプロンプトノードが子ノードであるかの情報を保存する（3.データの項参照）。

#### 【0095】

画面要素がコメントであれば、画面要素作成はコメント表示欄UI-30へのコメントUI-31の投稿である（コメント作成処理）。詳細は後述する。

#### 【0096】

50

図13に戻り、ユーザがワークスペース作成処理の終了操作を行った場合（ステップ4Yes）、プロセッサ122は必要に応じてデータの保存等を行ってワークスペース作成処理を終了し、引き続きワークスペース上の操作を行う場合（ステップ4No）、プロセッサ122は画面要素の作成操作を待機する（ステップ2）

【0097】

以上小括すると、制御部12はワークスペース作成部131として機能し、当該ワークスペース作成部131は、ワークスペースを表示する（ステップ1）ワークスペース表示部と、ユーザの操作により、前記ワークスペース上にプロンプトノードを作成する（ステップ4）プロンプトノード作成部と、一のプロンプトノードと他のプロンプトノードとを接続するエッジを作成する（ステップ5）エッジ作成部と、を備える。

10

また、プロンプトノードUI-10は、大規模言語モデルの選択を受け付ける大規模言語モデル選択欄（UI-102）と、プロンプトを受け付けるプロンプト受付欄（UI-103）と、プロンプトに対する大規模言語モデルの出力結果を表示する出力結果表示欄（UI-107）と、を含む。

【0098】

ここで、プロンプトノード作成部は、ワークスペース上に少なくとも2以上のプロンプトノードUI-10Pを作成する。プロセッサ122が、エッジにより互いに連結している複数のプロンプトノードUI-10Pをワークスペース上に一覧表示することで、各プロンプトがどのようなものか、結果はどうだったかなどについてユーザが容易に確認できるという利点がある。

20

【0099】

特に、プロンプトノード作成部がワークスペース上に（少なくとも3以上のプロンプトノードUI-10Pを作成し、かつ、）一つのプロンプトノードから異なる二つのプロンプトノードに至る少なくとも一つの分岐を含む場合（一番小さい例では親ノードの下に2つの子ノードがある場合（図6参照））、ユーザが2つの子ノードでプロンプトやその出力結果を並べて検討できるという利点がある。

【0100】

ワークスペース作成部131（プロンプトノード作成部）はさらに、プロンプト受付欄に含まれる特定の記述をもとに変数（本実施形態では中括弧内に記述されている文字列）を取得して保存する変数取得部と、変数の変更を受け付けて当該変数を置換する変数置換部を備える（ユーザーインターフェースの項（図2）参照）。

30

【0101】

< 2 - 2 . 大規模言語モデル入出力処理 >

大規模言語モデル入出力処理は、上述したように、選択されている大規模言語モデルを取得してプロンプトを前記大規模言語モデルに入力として提供し、また、前記プロンプトに対する前記大規模言語モデルの出力結果を取得する処理である。

大規模言語モデル入出力処理において、プロセッサ122は、大規模言語モデル選択欄UI-102で（ユーザに）選択されている大規模言語モデルに対してプロンプト受付欄UI-103に入力されているプロンプトを提供し、得られた出力結果を出力結果表示欄UI-107に表示する。

40

例えば、大規模言語モデルのAPIにプロンプトを送信し、その出力結果を取得する。

【0102】

プロセッサ122は、大規模言語モデル入出力プログラムP14に基づき、大規模言語モデル入出力処理を行う。

すなわち、大規模言語モデル入出力プログラムP14は、プロセッサ122による大規模言語モデル入出力処理の実行により、コンピュータを大規模言語モデル入出力手段として機能させる（制御部12は大規模言語モデル入出力部132として機能する）。

【0103】

図14は、大規模言語モデル入出力処理を示すフローチャートである。

プロセッサ122は、プロンプトノードUI-10の実行ボタンまたは再実行ボタンのユーザに

50

よる押下を受け付けて、大規模言語モデル入出力処理を開始する。

【0104】

プロセッサ122は、大規模言語モデル選択欄UI-102で選択されている大規模言語モデルを取得する（ステップ11）。またプロセッサ122は、プロンプト受付欄に入力されているプロンプトを取得する（ステップ12）。

【0105】

ここで、プロンプトに中括弧が含まれる場合、プロセッサ122はその中括弧に含まれる文字列（数値、記号含む）を変数として保存する。1つのプロンプトに中括弧が複数含まれている場合、プロセッサ122は各変数に一意的番号を付して保存する。

ユーザは、変数ボタンUI-105を押下することで、格納されている

10

【0106】

プロセッサ122は、ステップ12のプロンプトを、ステップ11の大規模言語モデルの入力として提供する（ステップ13）。

本実施形態において、プロセッサ122はプロンプトを大規模言語モデルのAPIに送信する。

【0107】

つづいてプロセッサ122は、大規模言語モデルによる出力結果を取得し（ステップ14）、出力結果表示欄UI-107に表示する（ステップ15）。またプロセッサ122は、出力結果などの各種データを保存する。

なお、大規模言語モデルによる出力結果を取得したあとプロセッサ122はその出力結果を加工等してから出力結果表示欄UI-107に表示してもよい。

20

【0108】

以上小括すると、制御部12は大規模言語モデル入出力部132として機能し、当該大規模言語モデル入出力部132は、選択されている大規模言語モデルを取得する（ステップ11）大規模言語モデル取得部と、プロンプトを大規模言語モデルに入力として提供する（ステップ13）プロンプト提供部と、前記プロンプトに対する大規模言語モデルの出力結果を取得する（ステップ14）出力結果取得部と、を備える。

【0109】

< 2 - 3 . プロンプト実行処理 >

プロンプト実行処理において、プロセッサ122は、ユーザが選択しているプロンプトノードの情報を取得し、最初のプロンプトノードから当該プロンプトノードまでのルートを取得したうえで、逐次入出力処理を行う。

30

逐次入出力処理では、最初のプロンプトノードからユーザに選択されているプロンプトノードに至る各プロンプトノードにおいて、大規模言語モデル入出力処理を行う。

【0110】

プロセッサ122は、プロンプト実行プログラムP16に基づき、プロンプト実行処理を行う。

すなわち、プロンプト実行プログラムP16は、プロセッサ122によるプロンプト実行処理の実行により、コンピュータをプロンプト実行手段として機能させる（制御部12はプロンプト実行部133として機能する）。

40

【0111】

図15は、プロンプト実行処理を示すフローチャートである。

図15に示すように、プロンプト実行処理は逐次入出力処理を含み、逐次入出力処理は各プロンプトノードにおける大規模言語モデル入出力処理の繰り返しである。

プロセッサ122は、ユーザによる再実行ボタンの押下を受け付け、プロンプト実行処理を開始する。

【0112】

プロセッサ122は、ユーザが選択したプロンプトノード、つまり再実行ボタンが押下されたプロンプトノードを取得し、最初のプロンプトノードから当該プロンプトノードまでのルートを探索する（ステップ21）。

50

本実施形態において、最初のプロンプトノードは最上位のプロンプトノード（あらかじめ定められるプロンプトノード）であり、例えば図 1 2 であればUI-11で確定する。これ以外の例については変形例で説明する。

【 0 1 1 3 】

ルートの探索に際し、プロセッサ122はエッジテーブルD20を参照し、上位ノード、下位ノードなどの世代情報などを参照する。

またプロセッサ122は、すべての末端のプロンプトノードに至るまでの全ルートを取得する。例えば図 1 2 であれば、UI-14aまでのルートと、UI-14bまでのルートを取得している。

【 0 1 1 4 】

そのうえで、ユーザが選択しているプロンプトノードまでのルートが複数ある場合、プロセッサは候補が複数あることを表示する。例えば、2つルートがある場合、それぞれのルートを色で分けるなどである。

複数ルートがある場合、ユーザはそのうちの一つを選択することができる。本実施形態では、選択可能なルートについてマウスオーバーされたときに、プロセッサが当該ルートを強調表示する。そして、ユーザは強調されている当該ルートを選択することができる。ユーザによるルートの選択を受け付けて、プロセッサ122はルートを確定する。

【 0 1 1 5 】

ルートが確定したら、プロセッサ122は逐次入出力処理を行う（ステップ22）。すなわち、最初のプロンプトノードから順に、大規模言語モデル入出力処理を実行し（ステップ23）。ユーザが実行（再実行）ボタンを押下したプロンプトノードにおける大規模言語モデル入出力処理が終わるまで繰り返す。ループ終了後、プロセッサ122はデータの保存等を行ってプロンプト実行処理を終了する。

【 0 1 1 6 】

以上小括すると、制御部12はプロンプト実行部133として機能し、当該プロンプト実行部133は、ユーザが選択するプロンプトノード（の情報）を取得する（ステップ21）選択ノード取得部と、最初のプロンプトノードからユーザが選択しているプロンプトノードまでのルートを取得する（ステップ21）ルート取得部と、前記ルート上の各プロンプトノードについて、最初のプロンプトノードから順に大規模言語モデル入出力処理を行う（ステップ22、ステップ23）逐次入出力部と、を備える。

< 2 - 4 . コメント作成処理 >

コメント作成処理において、プロセッサ122は、ユーザのコメントを取得し、そのコメントをコメント表示欄UI-30に表示する。開発支援システム1を使用する各ユーザは、コメントのやり取りをすることができる。

【 0 1 1 7 】

プロセッサ122は、コメント作成プログラムP18に基づき、コメント作成処理を行う。

すなわち、コメント作成プログラムP18は、プロセッサ122によるコメント作成処理の実行により、コンピュータをコメント作成手段として機能させる（制御部12はコメント作成部134として機能する）。

【 0 1 1 8 】

プロセッサ122は、ユーザによるコメントボタンの押下を受け付けることにより、コメント作成処理を開始する。

【 0 1 1 9 】

プロセッサ122は、コメントボタンが押下されたノード、コメントの内容、コメントボタンが押された日時、およびコメントボタンを押したユーザ名、の各情報（以下「コメント関連情報」とする。）を取得する。

そしてプロセッサ122は、コメント関連情報をコメント表示欄UI-30に表示する。

【 0 1 2 0 】

以上小括すると、制御部12はコメント作成部134として機能し、当該コメント作成部134は、ユーザのコメントを取得するコメント取得部と、取得しているコメントを（ワークス

10

20

30

40

50

ページのコメント表示欄に)表示するコメント表示部と、を備える。

3. データ

以下、本実施形態の開発支援システム1が扱うデータについて、図を用いて説明する。

本実施形態の開発支援システム1は、サーバ10の記憶部14(データ格納部14b)にノードテーブルD10、エッジテーブルD20、およびコメントテーブルD30を備える。

いずれのテーブルも作成日時(分、秒まで保存してもよい)をデータとして保有するが、下記では記載を省略する。

【0121】

ノードテーブルD10は、例えばプロンプトノードに関するデータ(プロンプトノードデータ)を備えるテーブルである。

10

【0122】

【表1】

ノードID	作成日時	プロンプト名	LLMモデル	プロンプト	出力結果	変数
N00001	xxxxxx xx:xx	プロンプト1	○○○	.....	.....	.....
N00002	xxxxxx xx:xx	プロンプト2-1	○○○	.....	.....	
N00003	xxxxxx xx:xx	プロンプト2-2	xxx	.....	.....	
N00004	xxxxxx xx:xx	プロンプト3	○○○	.....	.....	.....

【0123】

表1は、ノードテーブルD10が備えるデータを例示する。これはノードがプロンプトノードの場合の例である。

20

表1に示すように、本実施形態のノードテーブルD10は、一意のノードIDのほか、作成日時(タイムスタンプ)、プロンプト名入力欄UI-101に入力されているプロンプト名(ノード名)、大規模言語モデル選択欄UI-102でユーザが選択している大規模言語モデル(LLMモデル)、プロンプト受付欄UI-103に記載されているプロンプト、出力結果表示欄UI-107に表示される出力結果、および変数、の情報を備える。

【0124】

作成日時はデータが作成された日時であり、タイムスタンプである。データには日、時に限らず、分や秒のデータを含んでいてもよい。また、作成日時については以下のテーブルでも同様であるため、説明を省略する。

30

【0125】

エッジテーブルD20は、エッジに関するデータ(エッジデータ)を備えるテーブルである。

【0126】

【表2】

エッジID	作成日時	ノードID(上位)	ノードID(下位)
E00001	xxxxxx xx:xx	N00001	N00002
E00002	xxxxxx xx:xx	N00001	N00003
E00003	xxxxxx xx:xx	N00002	N00004
E00004	xxxxxx xx:xx	N00003	N00004

【0127】

表2は、エッジテーブルD20が備えるデータを例示する。

表2に示すように、エッジテーブルD20は、作成日時と、結合しているノード(上位ノードおよび下位ノード)のIDを備える。

【0128】

コメントテーブルD30は、コメントに関するデータ(コメントデータ)を備えるテーブルである。

【0129】

【表 3】

コメントID	作成日時	ノードID	ユーザID	コメント
C00001	xxxxxx xx:xx	N00001	USER0001	………
C00002	xxxxxx xx:xx	N00002	USER0002	………
C00003	xxxxxx xx:xx	N00002	USER0001	………
C00004	xxxxxx xx:xx	N00004	USER0004	………

## 【 0 1 3 0 】

表 3 は、コメントテーブルD30が備えるデータを例示する。

表 3 に示すように、コメントテーブルD30は、作成日時と、コメントされている（コメントボタンが押下されている）ノードのノードID、コメントしたユーザのユーザID、およびコメントの内容（テキストなど）を備える。

10

## 【 0 1 3 1 】

以上のように、ノードテーブルD10とエッジテーブルD20により、プロセッサ122は複数のノードのなかで、どれが上位ノードでどれが下位ノードかを判断することができ、ルート探索を行うことができる。

また、これらのテーブルの情報が管理・保存されることにより、プロセッサ122はワークスペース上のすべてのプロンプトノード上に、ユーザの判断に資する情報（大規模言語モデル、プロンプト、プロンプトに対する出力結果など）を表示することができる。

このほか、コメントテーブルD30により、プロセッサ122はコメント表示欄UI-30に表示するコメントの内容を管理することができる。

20

## 【 0 1 3 2 】

## 4．ハードウェア構成

図 1 6 は、開発支援システム1のネットワーク構成図を示す図である。

図 1 6 に示すように、本実施形態における開発支援システム1は、サーバ10、ユーザ端末20、および生成AIサーバ30を備える。また、これらの各装置は、ネットワークNを介して接続されている。ネットワークNは例えばインターネットなどである。

サーバ10には、開発支援プログラムP1を含み、本実施形態に係る開発支援システム1を動作させるためのソフトウェア（アプリケーションソフトウェア）がインストールされており、当該ソフトウェアの機能により、各種処理が実行される。

30

以下、各ハードウェアについて説明する。

## 【 0 1 3 3 】

なお図 1 6 は、サーバ10が開発支援プログラムP1を備え、開発支援システム1がウェブアプリケーションの形で提供される場合の構成図である。

なおこれに対し、ユーザ端末20が開発支援プログラムP1を備え、開発支援システム1がユーザ端末20で完結し、ネットワークNに接続しない場合もあり得る。

ただしこの場合、ネットワークNに接続していることで得られる特有の機能、例えば外部のデータベースや情報源からリアルタイムでデータを取得する技術のRAG（Retrieval-based Language Model）などは使用できない。

40

## 【 0 1 3 4 】

## &lt;サーバ10&gt;

サーバ10は、開発支援プログラムP1を実行するためのコンピュータである。

図 1 6 においてサーバ10は1台のみ図示しているが、数は1台に限られるものではなく、複数のサーバにより実現してもよい。

例えば、負荷分散や可用性の観点から、複数のサーバを用いることも考えられる。

サーバ10はクラウドサービス事業者のコンピュータを利用するものであってもよいし、ユーザがコンピュータを用意してもよい。

## 【 0 1 3 5 】

図 1 7 は、サーバ10のハードウェア構成図である。

図 1 7 に示すように、サーバ10は、制御部12、記憶部14、および通信制御部16を備える

50

。また制御部12は、プロセッサ122、ROM124、RAM126、計時部128を備える。それぞれの基本的な機能については後でまとめて説明する。

【0136】

プロセッサ122は、本実施形態においてCPU (Central Processing Unit) であり、プロセッサ122を備える制御部12は、開発支援部130としても機能する (不図示)。開発支援部130は、開発支援プログラムP1を実行して開発支援処理を行う。

開発支援部130は、ワークスペース作成部131、大規模言語モデル入出力部132、プロンプト実行部133、およびコメント作成部134を含む。

【0137】

また、一のプログラムは、別のプログラムを含んでいてもよい。例えば本実施形態において、開発支援プログラムP1は、ワークスペース作成プログラムP12や大規模言語モデル入出力プログラムP14などを含む。

10

【0138】

図17に示すように、記憶部14は、プログラム格納部14aとデータ格納部14bを備え、各種処理に必要なプログラムやデータを備える。

例えば、プログラム格納部14aには、本実施形態に係る開発支援プログラムP1のほか、サーバ10に接続されている機器を制御するための制御プログラム、例えば通信制御部16を制御する通信制御プログラムなどが格納 (インストール) されている。

【0139】

通信制御部16は、サーバ10と、外部にある端末、例えば、後述するユーザ端末20などとの間で通信を行う装置である。通信制御部16は、図16に示すように、サーバ10をネットワークNに接続する。

20

【0140】

上記のほか、サーバ10は、命令やデータの入力を行うための入力部 (例えばキーボード) や、情報を何らかの形で出力するための出力部 (例えば音声出力装置) などを備えていてもよい (不図示)。また、本実施形態の用途のために追加的に必要な装置や、本実施形態の用途について利便性を向上させるための装置を備えていてもよい。

【0141】

<ユーザ端末20>

ユーザ端末20は、ユーザが開発支援システム1を利用するための情報処理装置である。ユーザは、ユーザ端末20を用いてサーバ10にアクセスすることにより、開発支援システム1を利用する。

30

なお、図16では簡単のためユーザ端末20について1台のコンピュータのみ描画しているが、開発支援システム1において開発は複数人できるため、ユーザ端末20は、複数台が存在し得る。

【0142】

図18は、ユーザ端末20のハードウェア構成図である。図18に示すように、ユーザ端末20は、制御部22、記憶部24、通信制御部26、および入出力部 (入力部および出力部) 28を備える。

また制御部22は、プロセッサ222、ROM、RAM、および計時部を備える。入出力部28は、出力部として表示部を備える。

40

説明済みの内容と重複するものや、後述する基本的な機能に関するものは説明を省略する。

【0143】

ユーザは、表示部に表示されるワークスペースを見て、あるいは操作して、本実施形態の開発支援システム1を利用する。

【0144】

本実施形態において、ユーザ端末20はデスクトップPCである。ただし、ユーザ端末20はこれに限られるものではなく、スマートフォンやタブレットなどの携帯型端末であってもよい。

50

## 【0145】

ユーザ端末20のプログラム格納部24aには、本実施形態に係るユーザ端末用アプリケーションプログラム（ユーザ用アプリ）が格納されており、当該ソフトウェアの機能により、プロセッサ222が各種処理を実行する。

ただしこれはあくまで一例であり、例えばユーザ端末20ではブラウザ操作のみで操作が可能であり、ユーザ用アプリを要しないものであってもよい。

## 【0146】

各種処理とは、サーバ10から取得した情報などに基づく出力（画面表示、音声出力）や、ユーザの入力受付、または各種通信を含む。

例えば、ユーザがユーザ用アプリを起動すると、ユーザ端末20のプロセッサ222はネットワークNを通じてサーバ10と通信を行い、各種処理を実行する。

## 【0147】

なお本実施形態において、ユーザ用アプリは、ネットワークNを通じて、またはユーザ用アプリを格納している記憶媒体を通じてユーザ端末20にインストールされる。

## 【0148】

<生成AIサーバ30>

生成AIサーバ30は、大規模言語モデルを備えるコンピュータである。大規模言語モデルに対するプロンプトの入力を受け付けて、大規模言語モデルの出力を返すことができる。

図16では便宜上コンピュータを1台のみ描画しているが、生成AIサーバ30を構成するコンピュータは1つとは限らない。例えば、プロンプトの入力を受け付けるコンピュータと、大規模言語モデルによる処理を行うコンピュータは別であってもよい。

## 【0149】

本実施形態において、生成AIサーバ30は大規模言語モデル（例えばGPT-4（登録商標））の利用に供するコンピュータである。

つまり、本実施形態の開発支援システム1は、ユーザから取得したプロンプトを大規模言語モデル（GPT-4）のAPIに送信し、そのプロンプトに対する出力結果を取得する。

## 【0150】

（コンピュータの基本的機能に係る説明）

以下、制御部（プロセッサ、ROM、RAM、計時部）、記憶部、通信制御部、入力部、および出力部について説明する。

なお、本実施形態のいずれの端末においても、機能部間の接続態様（ネットワークトポロジ）は特に限定されない。例えばバス型であってもよいし、スター型、メッシュ型などであってもよい。

## 【0151】

プロセッサは、ROMや記憶部などに記憶されたプログラムに従って、情報処理や各種装置の制御を行う。本実施形態において、プロセッサはCPUである。

## 【0152】

ただし、プロセッサはこれに限られない。CPU、DSP（Digital Signal Unit）、GPU（Graphics Processing Unit）、GPGPU（General Purpose computing on GPU）、ASIC（Application Specific Integrated Circuit）、またはFPGA（Field Programmable Gate Array）など、各種プロセッサを単独で、あるいは組み合わせて用いてもよい。

例えば、CPUとGPUを統合したプロセッサはAPU（Accelerated Processing Unit）などと呼ばれるが、このようなプロセッサを用いてもよい。

## 【0153】

ROMは、プロセッサが各種制御や演算を行うための各種プログラムやデータがあらかじめ格納された、リードオンリーメモリである。

## 【0154】

RAMは、プロセッサにワーキングメモリとして使用されるランダムアクセスメモリである。このRAMには、本実施形態の各種処理を行うための各種エリアが確保可能になっている

10

20

30

40

50



る。

【0155】

計時部は、時間情報の取得などに係る計時処理を行う。コンピュータが通信制御部を備える場合は、NTP(ネットワーク・タイム・プロトコル)により外部から時間情報を取得してもよい。

【0156】

記憶部は、プログラムやデータなどの情報を記憶するための装置である。記憶部はストレージとも称する。記憶部は内蔵型か、外付型かを問わない。

【0157】

記憶部は、データの読み書きが可能な記憶媒体と、当該記憶媒体に読み書きするドライブとを含む。

記憶媒体は例えば、内蔵型や外付型があり、HD(ハードディスク)、CD-ROM、フラッシュメモリなどが挙げられる。

ドライブは例えば、HDD(ハードディスクドライブ)、SSD(ソリッドステートドライブ)などが挙げられる。

【0158】

記憶部は、機能部としてプログラム格納部とデータ格納部を備える。

プログラム格納部には、各種機器を制御するための制御プログラム、例えば通信を制御する通信制御プログラムなどが格納されている。

【0159】

通信制御部は、端末などの間で通信を行うための装置である。通信制御部は、当該通信制御部を備える端末をネットワークNに接続する。

【0160】

通信制御部の通信方式は公知の方式であり、機器に応じて有線による方式や無線による方式が適用される。

例えば、端末がデスクトップPCであれば有線、無線の両方の場合が考えられ、また、端末がスマートフォンであれば、無線による通信方式が考えられる。

【0161】

有線であれば、例えばIEEE802.3(例えばバス型やスター型の有線LAN)で規定される通信方式を好適に用いることができるが、それ以外にも、IEEE802.5(例えばリング型の有線LAN)で規定される通信方式などを用いてもよい。

【0162】

無線であれば、例えばIEEE802.11(例えばWi-Fi)で規定される通信方式を好適に用いることができるが、それ以外にも、IEEE802.15(例えばBluetooth(登録商標)、BLE(Bluetooth(登録商標)ローエナジー)など)、IEEE802.16(例えばWiMAX)、または赤外線通信などの光通信で規定される通信方式などを用いてもよい。

【0163】

入力部および出力部は、それぞれ端末に対する入力と出力を担う装置である。入力部および出力部をあわせて入出力部と称する場合がある。

入力部はユーザからの入力を受け付ける装置である。このような入力部として例えば、キーボード、ポインティングデバイスとしてのマウス、トラックパッド、タブレット、またはタッチパネルなどが挙げられる。

【0164】

端末がタブレットやスマートフォンなどであって、入力部がタッチパネルの場合、入力部はタッチスクリーンなど、画像などを表示する表示部の表面に配置される。この場合、入力部は、表示部に表示される各種操作アイコンに対応したユーザのタッチ位置を特定し、ユーザによる入力を受け付ける。

【0165】

出力部は例えば、画像や音声、帳票などを出力するための装置である。

出力部として例えば、タッチスクリーンやディスプレイ(液晶ディスプレイや有機EL

10

20

30

40

50

ディスプレイ)などの表示装置(表示部)や、スピーカなどの音声出力装置、プリンタなどの帳票出力装置が挙げられる。

【0166】

(他の実施形態)

【0167】

開発支援システム1は、さらに、RAG(Retrieval-Augmented Generation)機能(データインポート機能)を備える。

例えば、ノードがさらに各種ファイルを格納している情報提供ノードを備えており、プロセッサ122はあるプロンプトノード10Pとエッジ20で結合している情報提供ノード10Rから情報を取得してもよい。

10

本変形例における情報提供ノードは、大規模言語モデル選択欄UI-102、プロンプト受付欄UI-103、および出力結果表示欄UI-107を備えないものの、文書(テキスト)データ、画像データ、動画データ、または音声データを提供することができる。例えば、文書ファイル、画像ファイル、動画ファイル、または音声ファイルなどのファイルを添付することができる。

具体例を挙げると、情報提供ノード10Rにpdfファイルが添付され、当該情報提供ノードに結合しているプロンプトノード10Pのプロンプト受付欄UI-103には「情報提供ノードが備える情報を参照すること。」といったプロンプトが記載される。

これにより、そのプロンプトノード10Pが大規模言語モデルにプロンプトを送信する際、そのpdfの情報も共に送信することができる。

20

つまり、本実施形態のプロンプトノードは、「大規模言語モデルの選択を受け付ける大規模言語モデル選択欄と、プロンプトを受け付けるプロンプト受付欄と、前記プロンプトに対する前記大規模言語モデルの出力結果を表示する出力結果表示欄と」を含むが、情報提供ノードについてはこれに限るものではない。

情報提供ノードは、プロンプトノードに情報を提供するノードであり、少なくとも文書データ、画像データ、動画データ、または音声データのいずれかを備えるものであればよい。

また、上記は一例であり、データインポートの方法として公知の方法を適宜用いることができる。

小括すると、ノード作成部はワークスペース上にプロンプトノードおよび情報提供ノードを作成し、前記情報提供ノードは、少なくとも文書データ、画像データ、動画データ、または音声データのいずれかを備えることを特徴とする。

30

【0168】

本実施形態の開発支援システム1は、利用ユーザ権限を定義し、権限ごとにワークスペースの公開範囲を制限することができる。

例えば、プロセッサ122は、ユーザごとに「Owner」、「Admin」、「Member」または「Member外」の利用ユーザ権限を設定できる。

この場合、「Owner」のユーザは全権限の所有者であり、すべてのワークスペースの操作および他のユーザへの権限付与が可能である。「Admin」のユーザは他のユーザへの権限付与が可能である。「Member」のユーザは許可されているワークスペースについて操作が可能であり、「Member外」のユーザは許可されているワークスペースの一部操作が可能である。

40

ワークスペースの操作は例えば、ワークスペースの新規作成、複製、削除、閲覧、編集、コメント投稿などである。

上記の例で考えると、「Member」のユーザはどの操作もできるが、「Member外」のユーザはワークスペースの閲覧のみできる、という内容が考えられるが、この操作権原付与範囲も適宜変更可能である。

また、ワークスペースが複数ある場合、あるユーザはワークスペース1には「Member」権限を、ワークスペース2には「Member外」権限を付与することができる。

このような管理のため、データ格納部14bはユーザごとのデータベースであるユーザデ

50

ータベースなどを備えていてもよい。

このような機能により、例えば、プロジェクトごとにユーザの権限を変更し、プロジェクトメンバーを明確にすることができる。また、誤ってワークスペースを削除してしまうなどのユーザの誤操作を防ぐこと等ができる。

このように、ユーザごとにワークスペースへのアクセス権限を付与する処理を権原付与処理とし、権限付与処理を実行する制御部12は権限付与部として機能する。

小括すると、ワークスペース作成部は、さらに、ユーザごとにワークスペースへのアクセス権限を付与する権限付与部を備える。

【0169】

本実施形態の開発支援システム1は、ワークスペースごとにタグを設定することができる。

例えば、セールスやマーケティングといったタグを割り振ることができる。具体的には複数のワークスペースがある場合（例えばワークスペース1から10がある場合）において、ワークスペース1から6がセールス、ワークスペース7-9がマーケティング、ワークスペース10が広報、といった割り振りができる。

これにより、現場の生成AIを活用した業務課題吸い上げにおいて、管理しやすくなるという利点がある。

このように、ワークスペースにタグを付して管理する処理をタグ管理処理とし、タグ管理処理を実行する制御部12はタグ管理部として機能する。

小括すると、ワークスペース作成部は、さらに、ワークスペースにタグを付して管理するタグ管理部を備える。

【0170】

（変形例）

本発明は上述の実施形態に限られるものではなく、本発明の趣旨を逸脱しない範囲において、上述の実施形態に種々の変更を加えたものを含む。

【0171】

例えば上述したように、開発支援システム1がウェブアプリケーションの形式で提供されるほか、開発支援システム1がネットワークを要しないアプリケーションの形式で提供されてもよい。

【0172】

上述の実施形態では、ワークスペース上、上側が上位ノード、下側が下位ノードであった。ただし、ワークスペース上の位置にもとづく上位ノード、下位ノードの（プロセッサ122による）判断はこれに限られない。

例えば、ワークスペースの相対的に左側にあるノードを上位ノード、相対的に右側にあるノードを下位ノードとしてもよい。

【0173】

上述の実施形態では、大規模言語モデルからの出力結果として主にテキストデータが得られる例を述べたが、これに限られない。例えば、画像データ、動画データ、音声データなどのデータなどが含まれていてもよい。

また、出力結果がプログラムコードの場合、開発支援システム1のプロセッサは、プロンプト実行処理により、一連のプログラムを実行できるようにしてもよい。

大規模言語モデルへの入力も同様であり、テキストに限らず、画像データ、動画データ、音声データなどのデータなどを入力としてもよい。また、プログラムコードを入力としてもよい。

【0174】

上述の実施形態では、大規模言語モデル選択欄UI-102は、ユーザがそのプロンプトノードで選択する大規模言語モデルをプルダウンで選択するものであったが、選択を受け付ける方法はこれに限られない。プロンプトノードUI-10Pごとにユーザが大規模言語モデルを選ぶことができる点がポイントであるためである。

例えば、ユーザが大規模言語モデル名をテキスト等で入力し、プロセッサ122がその大

10

20

30

40

50

規模言語モデルの情報を取得するようにしてもよい。

またこの場合、入力されている大規模言語モデルのAPIが存在しない、利用できない場合などにおいて、プロセッサ122はエラーを返すようにしてもよい。

【0175】

上述の実施形態では、実行ボタンが押下されたプロンプトノードに至るルートが複数ある場合、ユーザがルートを選択するようにしていた。しかし、これに限られるものではなく、プロセッサ122がコストの小さいルートを自動で計算し、最適ルートを表示するようにしてもよい。

例えば、各プロンプトノードおよび/または各エッジにユーザがコストを決定することができ、ユーザが選択する所定のプロンプトノードへのルートを探査する際、プロセッサ122は最初のプロンプトノードから目的のプロンプトノードまでの各ルートでコスト計算を行い、最もコストの低いルートを表示する。

すなわちこの場合、開発支援システム1は、プロンプトノードおよび/またはエッジにユーザが決定するコストを設定するコスト設定部を備える。

この場合、プロセッサ122は、ノードテーブルD10またはエッジテーブルD20にコストを保存する。

また、プロンプト実行処理のルート探査（ステップ21）において、プロセッサは上述したコスト計算を行う。

これにより、ワークスペース上のプロンプトノードとエッジが多くなったときでも、ユーザは最適のルートを見つけやすいという利点がある。

【0176】

上述のプロンプト実行処理において、最初のプロンプトノードは最上位プロンプトノードであった。しかしながら、最初のプロンプトノードはユーザの任意によるもの、あるいはあらかじめ定められているものであるため、これに限られない。つまり、ユーザが、（プロンプト実行処理を行う最後のプロンプトノードだけではなく、）最初のプロンプトノードを指定できるようにしてもよい。

また、最初のプロンプトノードと最後のプロンプトノードとの間（ルート上）にあるプロンプトノードの中で任意のプロンプトノードを通過ノードとしてユーザが指定できるようにしてもよい。

前半の処理は第一ノード指定処理、とし、この処理を行う制御部12は第一ノード指定部として機能する。また後半の処理は通過ノード指定処理とし、この処理を行う制御部12は通過ノード指定部として機能する。

【0177】

本実施形態を含む本発明の態様は、換言すると以下の特徴を備える。下記は本願出願時における特許請求の範囲と対応する。ただし、出願後における特許請求の範囲の補正により、当該補正後の特許請求の範囲の記載とは異なる場合がある。

（1）第1の態様では、ワークスペース作成部および大規模言語モデル入出力部を備える開発支援システムであって、前記ワークスペース作成部は、ワークスペースを表示するワークスペース表示部と、ユーザの操作により、前記ワークスペース上にプロンプトノードを作成するノード作成部と、一のプロンプトノードと他のプロンプトノードとを接続するエッジを作成するエッジ作成部と、を備え、

前記プロンプトノードは、大規模言語モデルの選択を受け付ける大規模言語モデル選択欄と、プロンプトを受け付けるプロンプト受付欄と、前記プロンプトに対する前記大規模言語モデルの出力結果を表示する出力結果表示欄と、を含み、

前記大規模言語モデル入出力部は、選択されている前記大規模言語モデルを取得する大規模言語モデル取得部と、前記プロンプトを前記大規模言語モデルに入力として提供するプロンプト提供部と、前記プロンプトに対する前記大規模言語モデルの出力結果を取得する出力結果取得部と、を備えることを特徴とする、開発支援システムを提供する。

（2）第2の態様では、さらに、プロンプト実行部を備え、前記プロンプト実行部は、ユーザが選択しているプロンプトノードを取得する選択ノード取得部と、最初のプロンプ

10

20

30

40

50

トノードから前記ユーザが選択しているプロンプトノードまでのルートを取得するルート取得部と、前記ルート上の各プロンプトノードについて、最初のプロンプトノードから順に前記大規模言語モデル入出力部による大規模言語モデル入出力処理を実行する逐次入出力部と、を備えることを特徴とする、第1の態様に記載の開発支援システムを提供する。

この場合、複数のプロンプトノードが繋がり、関連しあっている中で、一部のプロンプトノードについて変更を加えた時に、ユーザが選択するプロンプトノードにおいて、大規模言語モデルからの出力結果がどのように変化するか、などを確認することができ、検討することができる。

(3) 第3の態様では、さらに、コメント作成部を備え、前記コメント作成部は、ユーザのコメントを取得するコメント取得部と、取得しているコメントを表示するコメント表示部と、を備えることを特徴とする、第1の態様に記載の開発支援システムを提供する。

この場合、プロンプト変更時などのメモ等をコメントとして記しておくことで、コメントした本人や、そのコメントを見た他のユーザがそのプロンプト変更の意図を確認することができ、開発効率が向上する。

(4) 第4の態様では、前記ワークスペース作成部が、さらに、前記プロンプト受付欄に含まれる特定の記述をもとに変数を取得して保存する変数取得部と、変数の変更を受け付けて当該変数を置換する変数置換部と、を備えることを特徴とする、第1の態様に記載の開発支援システムを提供する。

この場合、変数の中身を変化させ、検討することができるため、開発効率が向上する。

(5) 第5の態様では、前記ワークスペース上にプロンプトノードを作成するノード作成部が、前記ワークスペース上にプロンプトノードおよび情報提供ノードを作成するプロンプトノード作成部であり、前記情報提供ノードは、少なくとも文書データ、画像データ、動画データ、または音声データのいずれかを備えることを特徴とする、第1の態様に記載の開発支援システムを提供する。

この場合、RAG (Retrieval-Augmented Generation) 機能 (データインポート機能) により、良質な情報を大規模言語モデルへの入力として提供することができる。

(6) 第6の態様では、前記ワークスペース作成部が、さらに、ユーザごとにワークスペースへのアクセス権限を付与する権限付与部を備えることを特徴とする、第1の態様に記載の開発支援システムを提供する。

この場合、例えばユーザが参加するプロジェクトに関連するワークスペースにのみ編集権限等を与えることで、自分がどのプロジェクトに参加しているか、あるプロジェクトに参加しているユーザは誰かなどの管理が容易になる。また、自身のプロジェクトに関係のないワークスペースを編集してしまうなどの誤操作などを防ぐことができる。

(7) 第7の態様では、前記ワークスペース作成部が、さらに、前記ワークスペースにタグを付して管理するタグ管理部を備えることを特徴とする、第1の態様に記載の開発支援システムを提供する。

この場合、複数のワークスペースが併存する場合であっても、ユーザが関連するタグが付いたワークスペースのみを抽出することができ、利便性が向上する。

(8) 第8の態様では、前記ワークスペース作成部が、さらに、プロンプトノードおよび/またはエッジにユーザが決定するコストを設定するコスト設定部を備えることを特徴とする、第1の態様に記載の開発支援システムを提供する。

この場合、プロンプトノードやエッジが多くなり、ルートを選択することに困難が生じた際に、ユーザによるルート決定を支援することができるという利点がある。

(9) 第9の態様では、コンピュータをワークスペース作成手段および大規模言語モデル入出力手段として機能させる開発支援プログラムであって、

前記ワークスペース作成手段は、ワークスペースを表示するワークスペース表示手段、ユーザの操作により、前記ワークスペース上にプロンプトノードを作成するノード作成手段、および一のプロンプトノードと他のプロンプトノードとを接続するエッジを作成するエッジ作成手段、を備え、

前記プロンプトノードは、大規模言語モデルの選択を受け付ける大規模言語モデル選択

10

20

30

40

50

欄と、プロンプトを受け付けるプロンプト受付欄と、前記プロンプトに対する前記大規模言語モデルの出力結果を表示する出力結果表示欄と、を含み、

前記大規模言語モデル入出力手段は、選択されている前記大規模言語モデルを取得する大規模言語モデル取得手段、前記プロンプトを前記大規模言語モデルに入力として提供するプロンプト提供手段、および、前記プロンプトに対する前記大規模言語モデルの出力結果を取得する出力結果取得手段、を備えることを特徴とする、開発支援プログラムを提供する。

【産業上の利用可能性】

【0178】

生成AIを利用する開発全般に応用することができる。例えば、プログラムで出力することをプロンプトに含めることにより、生成AIを用いるプログラム開発用途にも適用できる。

10

【符号の説明】

【0179】

## 1 開発支援システム

10 サーバ

12 制御部

122 プロセッサ

124 ROM

126 RAM

20

128 計時部

130 開発支援部

131 ワークスペース作成部

132 大規模言語モデル入出力部

133 プロンプト実行部

134 コメント作成部

14 記憶部

14a プログラム格納部

14b データ格納部

16 通信制御部

30

20 ユーザ端末

22 制御部

222 プロセッサ

24 記憶部

26 通信制御部

28 入出力部

30 生成AIサーバ

UI-1 ワークスペース

UI-10 ノード

UI-10P プロンプトノード

40

UI-10R 情報提供ノード

UI-101 プロンプト名入力欄

UI-102 大規模言語モデル選択欄

UI-103 プロンプト受付欄

UI-104 処理ボタン

UI-105 変数ボタン

UI-106 変数追加ボタン

UI-107 出力結果表示欄

UI-108 新規ボタン

UI-109 複製ボタン

50

- UI-11 親ノード（第一世代ノード）
- UI-12 子ノード（第二世代ノード）
- UI-12a 第一子ノード（第一の第二世代ノード）
- UI-12b 第二子ノード（第二の第二世代ノード）
- UI-12c 第三子ノード（第三の第二世代ノード）
- UI-13 孫ノード（第三世代ノード）
- UI-14 ひ孫ノード（第四世代ノード）
- UI-20 エッジ
- UI-30 コメント表示欄
- UI-31 コメント
- P1 開発支援プログラム
- P12 ワークスペース作成プログラム
- P14 大規模言語モデル入出力プログラム
- P16 プロンプト実行プログラム
- P18 コメント作成プログラム
- D10 ノードテーブル
- D20 エッジテーブル
- D30 コメントテーブル

10

**【要約】**

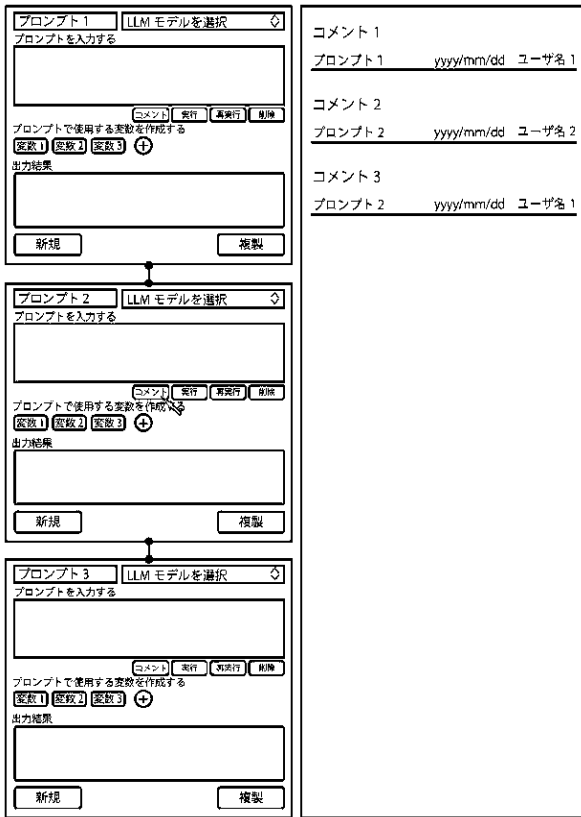
**【課題】**ワークスペース上にノードとエッジとを自由に作成できる開発支援システムであって、プロンプトノードごとに大規模言語モデルを選択でき、また、ノード単位でプロンプト提供（送信）と、出力結果の取得が可能な開発支援システムを提供する。

20

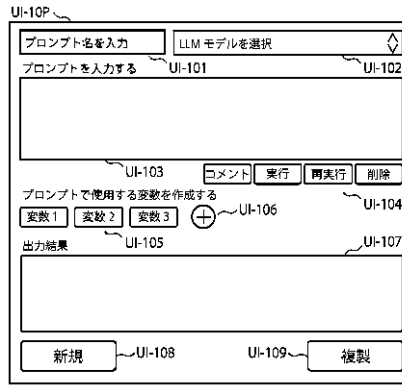
**【解決手段】**開発支援システム1は、ワークスペース作成部および大規模言語モデル入出力部を備え、ワークスペースに作成されるプロンプトノードは、大規模言語モデル選択欄、プロンプト受付欄、およびプロンプトに対する出力を表示する出力結果表示欄を含む。またプロンプトノードごとに、大規模言語モデルへのプロンプト提供と、結果取得を可能とし、プロンプトノードは出力結果を表示する。

**【選択図】**図1

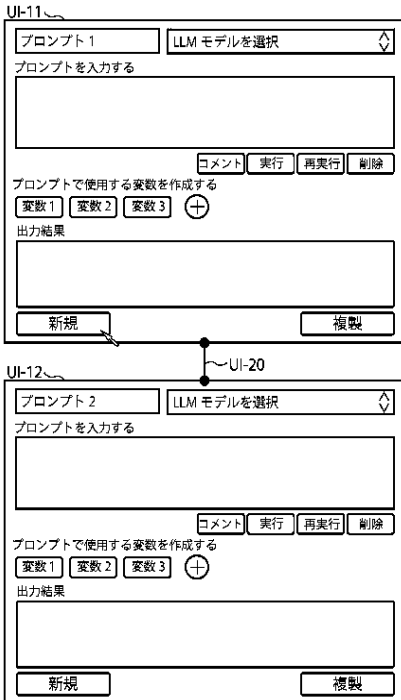
【 図 1 】



【 図 2 】



【 図 3 】

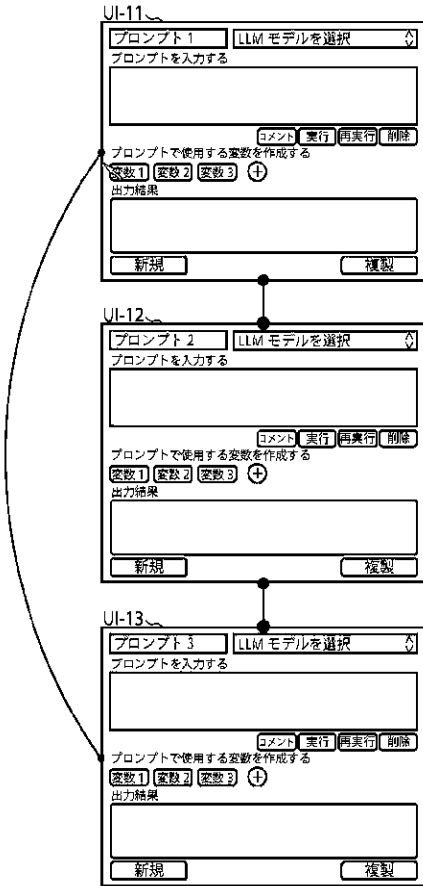


【 図 4 】

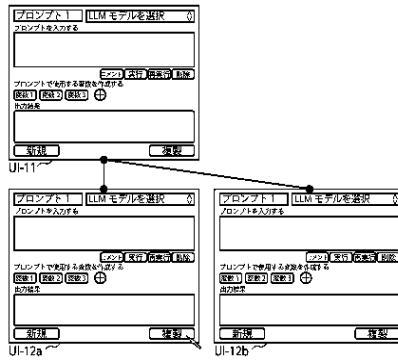




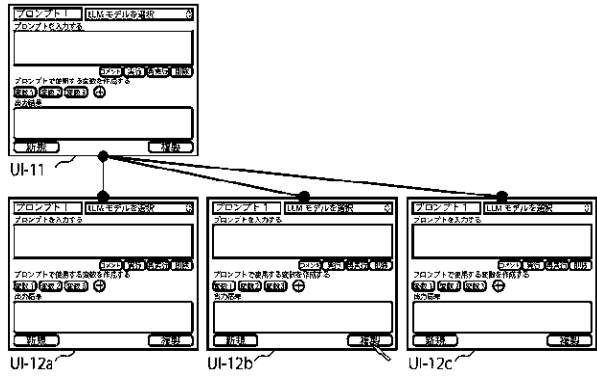
【図5】



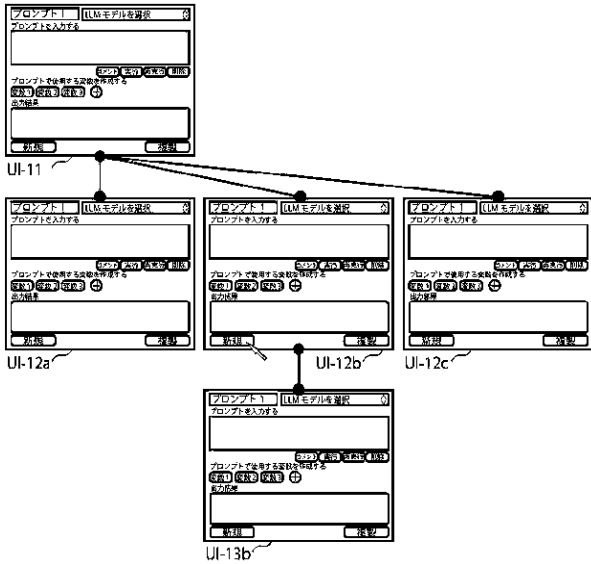
【図6】



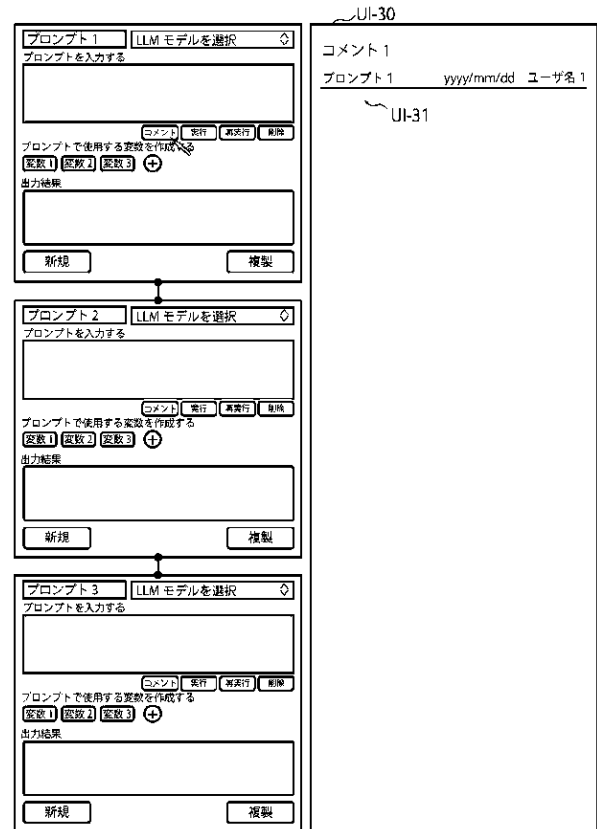
【図7】



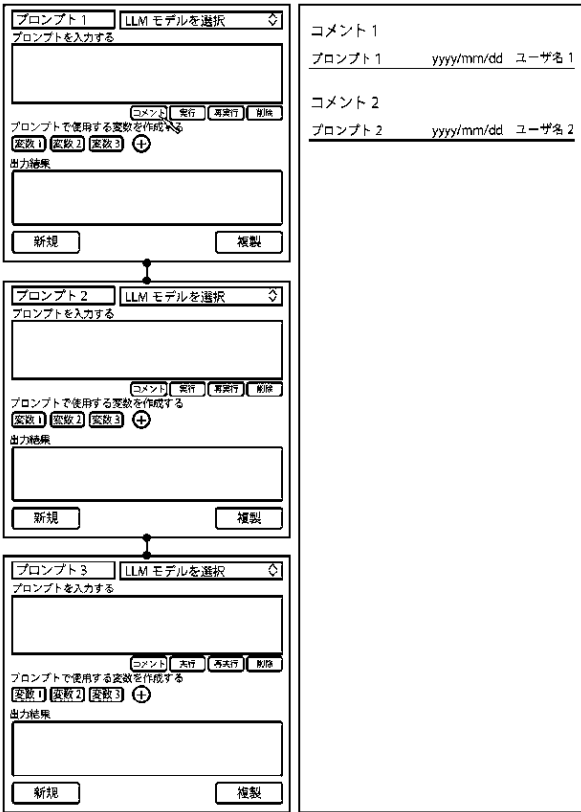
【図8】



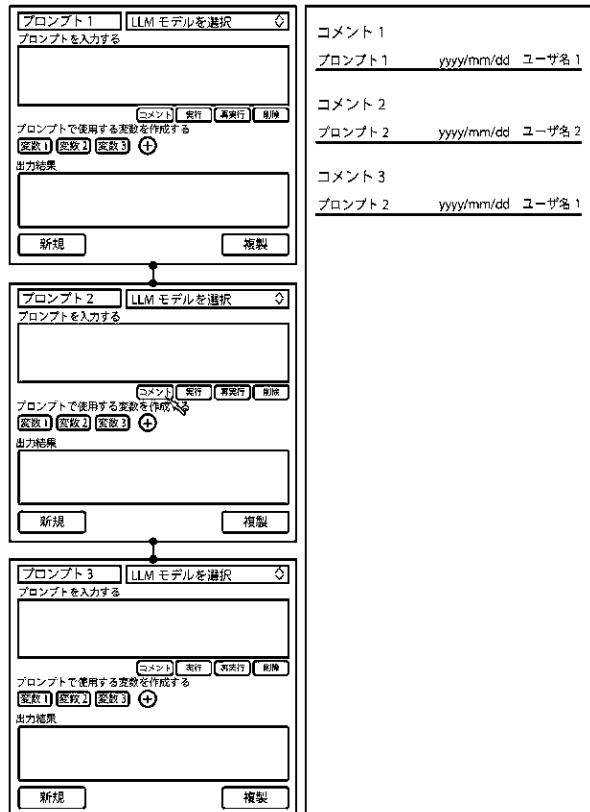
【図9】



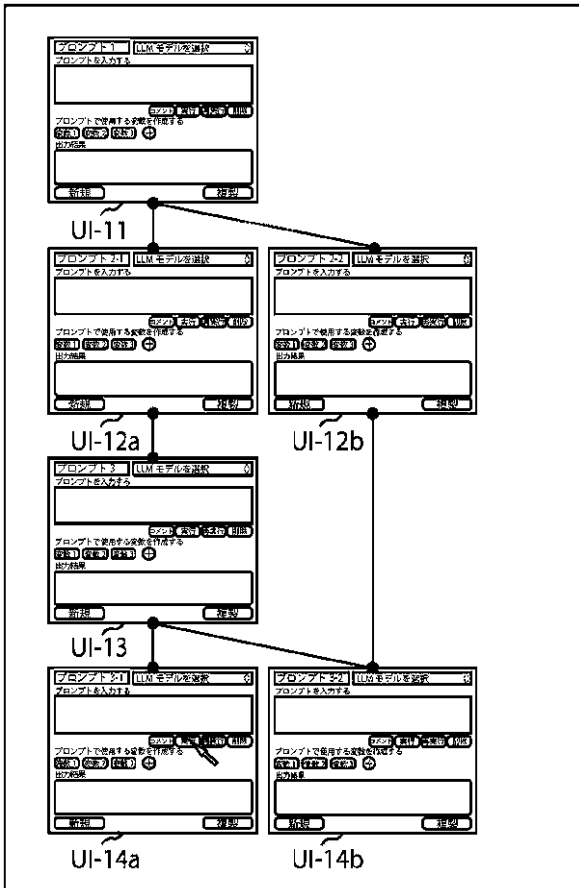
【図 10】



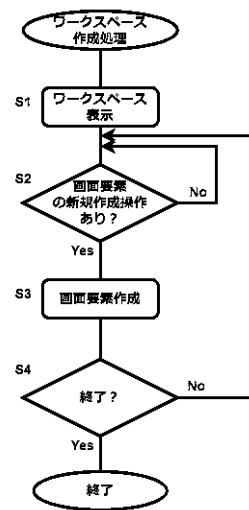
【図 11】



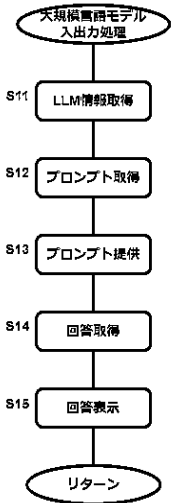
【図 12】



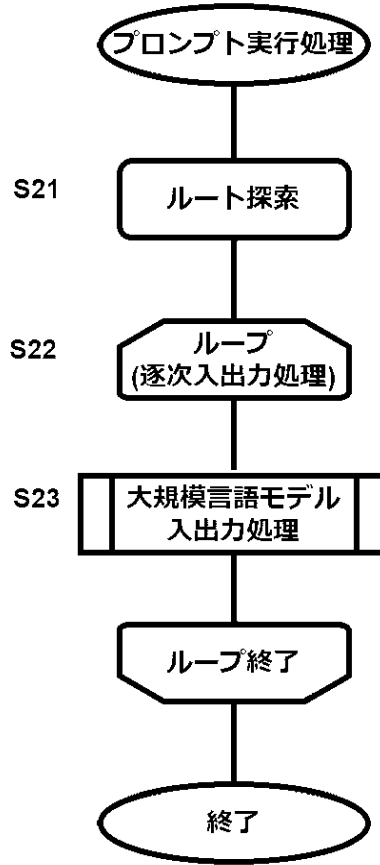
【図 13】



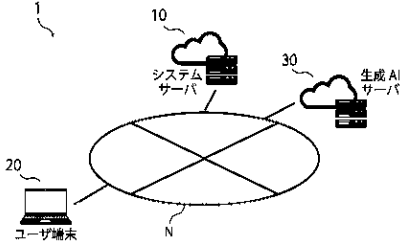
【図14】



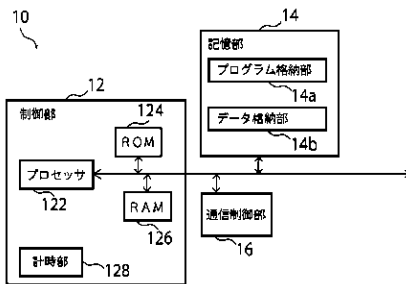
【図15】



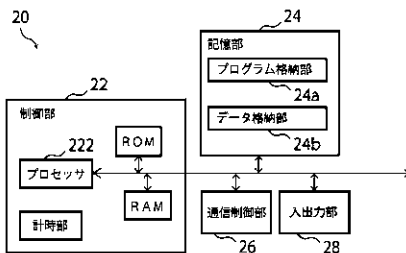
【図16】



【図17】



【図18】



---

フロントページの続き

(56)参考文献 特開平 0 8 - 1 8 5 3 3 8 ( J P , A )

特開 2 0 2 2 - 0 1 1 4 4 2 ( J P , A )

特開 2 0 1 9 - 1 9 2 1 5 8 ( J P , A )

Ian Arawjo ほか 4 名 , ChainForge:A visual Toolkit for Prompt Engineering and LLM Hypothesis Testing , arXiv , 2023年12月20日 , pages 1-25 , <https://arxiv.org/pdf/2309.09128.pdf>

Tongshuang Wu ほか 6 名 , PromptChainer:Chaining Large Language Model Prompts through Visual Programming , arXiv , 2022年03月13日 , <https://arxiv.org/pdf/2203.06566.pdf>

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F 8 / 3 4