

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7422767号
(P7422767)

(45)発行日 令和6年1月26日(2024.1.26)

(24)登録日 令和6年1月18日(2024.1.18)

(51)国際特許分類 F I
G 0 6 F 8/00 (2018.01) G 0 6 F 8/00
G 0 6 F 40/56 (2020.01) G 0 6 F 40/56

請求項の数 64 (全137頁)

(21)出願番号	特願2021-531877(P2021-531877)	(73)特許権者	521241786
(86)(22)出願日	令和2年1月3日(2020.1.3)		ブレーン コグニティブズ ピーティーイー
(65)公表番号	特表2022-516227(P2022-516227 A)		ー、リミテッド
(43)公表日	令和4年2月25日(2022.2.25)		シンガポール国, シンガポール 0 4 8
(86)国際出願番号	PCT/SG2020/050004		6 1 6, ワン ラッフルズ プレイス, ラ
(87)国際公開番号	WO2020/145892		ッフルズ プレイス 1, タワー 2, ナ
(87)国際公開日	令和2年7月16日(2020.7.16)	(74)代理人	100079108
審査請求日	令和4年12月28日(2022.12.28)		弁理士 稲葉 良幸
(31)優先権主張番号	201941001135	(74)代理人	100109346
(32)優先日	平成31年1月10日(2019.1.10)		弁理士 大貫 敏史
(33)優先権主張国・地域又は機関	インド(IN)	(74)代理人	100117189
(31)優先権主張番号	201941028675		弁理士 江口 昭彦
(32)優先日	令和1年7月16日(2019.7.16)	(74)代理人	100134120
	最終頁に続く		弁理士 内藤 和彦
			最終頁に続く

(54)【発明の名称】 自然言語ソリューション

(57)【特許請求の範囲】

【請求項1】

ユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリューションを構築する方法であって、

ユーザから計算デバイスのプロセッサにより、前記自然言語を使用して構築される前記ソリューションを示すグローバル意図言明を受信することであって、前記グローバル意図言明は、前記自然言語の形態で受信され、潜在状態に設定される、ことと、

前記ユーザから前記プロセッサにより、前記グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明並びに各ローカル意図言明に関連付けられたN個の実体及び1つのエージェントの詳細を受信することであって、各ローカル意図言明並びに各実体及び前記エージェントの前記詳細は、前記自然言語の形態で受信され、それぞれ潜在状態に設定され、各ローカル意図言明は、前記ソリューションを実行するための要件を履行するサブステップを示す文章であり、各実体は、名詞句を含み、且つ対応する前記ローカル意図言明により示される前記サブステップの前記要件の履行に参加し、前記エージェントは、人間エージェント及び機械エージェントの少なくとも一方である、ことと、

実体毎に、前記ユーザから前記プロセッサにより前記自然言語の形態で、前記実体のそれぞれの特性を定義し、且つ前記対応するローカル意図言明の他の実体から前記実体のそれぞれを差別化する、潜在状態に設定された1つ又は複数の属性を受信することであって、各属性は、形容詞句及び副詞句の少なくとも1つを含み、

前記エージェントは、少なくとも、各属性、各実体、各ローカル意図言明及び前記グ

ローカル意図言明の前記潜在状態を現実状態に変える、ことと、

前記プロセッサにより、ローカル意図言明毎に1組の実体-状態組合せ(CES)を形成することであって、前記1組のCESは、前記ローカル意図言明それぞれの前記N個の実体の 2^N 個の可能な組み合わせを含み、前記それぞれのローカル意図言明の全て(N個)の実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、前記1組中の各CESは、潜在状態に設定され、且つ関連付けられた前記実体を現実状態に変えることに応答して現実状態に変わる、ことと、

前記グローバル意図言明に関連付けられた、受信したローカル意図言明が1つのみであるとの判断に応答して、前記受信したローカル意図言明の前記トリガーCESを前記ソリューションの構築の終了として識別することと、

10

前記グローバル意図言明に関連付けられた、受信したローカル意図言明が2つ以上であるとの判断に応答して、前記ユーザから前記プロセッサにより前記自然言語の形態で、前記ローカル意図言明間の予め定義されたルール、制約及び公式のうちの1つ又は複数に基づく複数の別個の関係を受信することであって、別個の各関係は、前記ソリューションを実行するための前記要件を履行する別個の経路であり、前記関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明の前記1組のCESに影響しているか、それとも前記ソリューションの構築の終了であることを示す、ことと、
を含み、

前記潜在状態は、空のバイナリ状態であり、前記現実状態は、空ではないバイナリ状態であり、

20

前記自然言語の形態の情報は、手書きベースインターフェース、タッチセンシティブインターフェース、音声ベースインターフェース、又はそれらの組合せを通して受信される、方法。

【請求項2】

各ローカル意図言明の実体毎に、

前記関連するエージェントから前記プロセッサにより前記自然言語の形態で、前記実体のそれぞれに対する値を受信することであって、前記実体のそれぞれに対する値を受信することは、前記受信される値に基づいて、前記実体のそれぞれについての前記潜在状態を現実状態に変えるイベントの記録であり、各ローカル意図言明に関連する全ての実体に対する値を受信することは、前記ローカル意図言明のそれぞれについての前記潜在状態を現実状態に変えるイベントの記録であり、全てのローカル意図言明に関連する全ての実体に対する値を受信することは、前記グローバル意図言明についての前記潜在状態を現実状態に変えるイベントの記録である、こと

30

を更に含む、請求項1に記載の方法。

【請求項3】

各ローカル意図言明の属性毎に、

前記関連するエージェントから前記プロセッサにより前記自然言語の形態で、前記属性のそれぞれに対する値を受信することであって、前記属性のそれぞれに対する値を受信することは、前記受信される値に基づいて、前記属性のそれぞれについての前記潜在状態を現実状態に変えるイベントの記録である、こと

40

を更に含む、請求項1に記載の方法。

【請求項4】

各実体の属性について、

前記関連するエージェントから前記プロセッサにより前記自然言語の形態で、前記属性のそれぞれに対する値を受信することであって、前記属性のそれぞれに対する値を受信することは、前記受信される値に基づいて、前記属性のそれぞれについての前記潜在状態を現実状態に変えるイベントの記録である、こと

を更に含む、請求項1に記載の方法。

【請求項5】

前記プロセッサにより、ログイン詳細、バイOMETリック詳細、顔認識技法、及び網膜

50

検出技法の1つに基づいて前記関連するエージェントを認証することを更に含む、請求項2に記載の方法。

【請求項6】

前記機械エージェントは、有線通信チャンネル又は無線通信チャンネルを介して前記計算デバイスに通信可能に結合される、請求項1に記載の方法。

【請求項7】

前記機械エージェントは、前記計算デバイスの機能モジュールであり、前記プロセッサに結合される、請求項1に記載の方法。

【請求項8】

前記グローバル意図言明に関連するローカル意図言明は、
前記別のコンピュータ実施ソリューションのローカル意図言明、又は
前記別のコンピュータ実施ソリューションのグローバル意図言明
である、請求項1に記載の方法。

10

【請求項9】

前記グローバル意図言明のローカル意図言明に関連する実体は、
前記別のコンピュータ実施ソリューションのローカル意図言明、又は
前記別のコンピュータ実施ソリューションの実体
である、請求項1に記載の方法。

【請求項10】

前記ローカル意図言明のそれぞれの特性を定義する属性は、
前記別のコンピュータ実施ソリューションの実体、又は
前記別のコンピュータ実施ソリューションの属性
である、請求項1に記載の方法。

20

【請求項11】

前記実体のそれぞれの前記特性を定義する属性は、
前記別のコンピュータ実施ソリューションの実体、又は
前記別のコンピュータ実施ソリューションの属性
である、請求項1に記載の方法。

【請求項12】

前記ソリューションは、
ローカル意図言明を2つ以上の他のローカル意図言明に分割し、前記対応する実体及び
エージェントを各ローカル意図言明に添付すること、又は
2つ以上のローカル意図言明を結合して1つのローカル意図言明にし、前記対応する実
体及びエージェントをそのローカル意図言明に添付すること
により構築される、請求項1に記載の方法。

30

【請求項13】

前記ソリューションは、自然言語を使用して構築されるため、1つ又は複数のサーチエ
ンジンによりサーチ可能である、請求項1に記載の方法。

【請求項14】

各実体にわたる1つ又は複数の情報権又は決定権を前記実体のそれぞれの前記ローカル
意図言明に関連する前記エージェントに割り当てることと、前記割り当てられた情報権又
は決定権に基づいて実体を前記エージェントに表示することとを更に含む、請求項1に記
載の方法。

40

【請求項15】

ユーザインターフェースを通して実体を表示して、前記関連するエージェントにより前
記実体に対する値を受信することを更に含む、請求項1に記載の方法。

【請求項16】

ローカル意図言明の1つ若しくは複数の前記実体又は1つ若しくは複数のCESは、他
のコンピュータ実施ソリューションの1つ又は複数のローカル意図言明と共有され、前記
共有実体又はCESは、所与の時点で1つのみのトリガーCESに参加して、前記1つの

50

トリガー C E S に対応する前記ローカル意図言明について前記潜在状態から前記現実状態に変更し、前記共有実体又は C E S の可用性は、複数のローカル意図言明についての前記潜在状態から前記現実状態への変更の順序に影響する、請求項 1 に記載の方法。

【請求項 17】

前記実体は本質的に物理的であり、前記実体は時間及び空間に存在する、請求項 1 に記載の方法。

【請求項 18】

前記ローカル意図言明を受信すること及び前記実体の前記詳細を受信することは、他のコンピュータ実施ソリューションのローカル意図言明及び実体を記憶したデータベース中のライブラリから 1 つ又は複数のローカル意図言明候補及び 1 つ又は複数の実体候補を提供することを含む、請求項 1 に記載の方法。

10

【請求項 19】

前記プロセッサにより、前記別個の関係の 1 つを選択して、前記実体に対する値を受信し、他の別個の関係をなくすことを更に含む、請求項 1 に記載の方法。

【請求項 20】

前記自然言語は、1 つ若しくは複数の母語、1 つ若しくは複数の手話、1 つ若しくは複数の記号、1 つ若しくは複数の数字、又はそれらの組合せに基づく、請求項 1 に記載の方法。

【請求項 21】

前記自然言語の形態で受信される前記情報は、物体認識技術、文字認識技術、画像認識技術、又はそれらの組合せを使用して解読される、請求項 1 に記載の方法。

20

【請求項 22】

前記プロセッサにより、記録すべき残りのイベント数を特定することと、前記プロセッサにより前記残りのイベント数を表示することと、を更に含む、請求項 2 に記載の方法。

【請求項 23】

各実体及び各ローカル意図言明のユーザインターフェース上の場所は、ドラッグアンドドロップ機能であって、有線入力デバイス、無線入力デバイス、及びタッチセンシティブインターフェースの少なくとも 1 つにより実行される、ドラッグアンドドロップ機能、又は

30

前記実体のそれぞれ若しくは前記ローカル意図言明のそれぞれの場所座標に対応する属性の値の変更により変更可能である、請求項 1 に記載の方法。

【請求項 24】

公開ブロックチェーン技術、秘密ブロックチェーン技術、及びハイブリッドブロックチェーン技術の 1 つを適用することを更に含む、請求項 1 に記載の方法。

【請求項 25】

前記公開ブロックチェーン技術、前記秘密ブロックチェーン技術、及び前記ハイブリッドブロックチェーン技術のそれぞれは、対称鍵暗号化技法、非対称鍵暗号化技法、又はそれらの組合せに基づく、請求項 24 に記載の方法。

40

【請求項 26】

1 つ又は複数の予め定義された規範に基づいて測定フレームワークを前記実体に割り当てることを更に含み、前記測定フレームワークは、実体が前記潜在状態から現実状態に変更される時間期間を示す、請求項 1 に記載の方法。

【請求項 27】

データベース中のライブラリの形態で、別のコンピュータ実施ソリューションの構築で使用するために、前記グローバル意図言明、前記ローカル意図言明、前記実体、及び前記エージェントの 1 つ又は複数に関連するデータを記憶することを更に含む、請求項 1 に記載の方法。

【請求項 28】

50

前記データベース内の前記ライブラリへのクエリベースのアクセスを提供することを更に含み、前記クエリベースのアクセスは、再使用に適切な実体の識別のためのサーチ及びクエリベースの技術に依存し、前記クエリベースのアクセスは、構造化クエリ言語 (SQL) 及びノットオンリー構造化クエリ言語 (NoSQL) の少なくとも一方を含む、請求項 2.7 に記載の方法。

【請求項 29】

ドラッグアンドドロップアクセスを前記データベース中の前記ライブラリに提供することを更に含み、請求項 2.7 に記載の方法。

【請求項 30】

前記 CES の 1 つ又は複数の過去の挙動及び前記イベントの 1 つ又は複数の発生に基づいて、前記実体の 1 つ又は複数及び前記イベントの 1 つ又は複数に対応する 1 つ又は複数の不確実性及び確率を提供することを更に含み、請求項 2.2 に記載の方法。

10

【請求項 31】

会計システム及び金融システムに関する 1 つ又は複数の実体を前記コンピュータ実施ソリューションの各ローカル意図言明に添付することにより、会計システム及び金融システムを統合することを更に含み、請求項 1 に記載の方法。

【請求項 32】

1 つ又は複数のアドバンスプランニングアンドオプティマイゼーション (APO) 機能を実行して、前記実体の 1 つ又は複数及び前記イベントの 1 つ又は複数に対応する 1 つ又は複数の不確実性及び確率を提供し、前記イベントの記録についての前記値の受信を最適化することを更に含み、請求項 3.0 に記載の方法。

20

【請求項 33】

機械学習技法及び人工知能技法、
実体中心手法であって、前記実体中心手法は 1 つ又は複数のイベントの発生を提供する、実体中心手法、並びに
各イベントのデータ及び挙動パターンの記憶
の少なくとも 1 つに基づいて、前記 CES の 1 つ又は複数の過去の挙動及び前記イベントの 1 つ又は複数の発生を査定することを更に含み、

前記機械学習技法は、教師あり学習技法、及び / 又は教師なし学習技法、及び / 又は半教師あり学習技法を含み、前記教師あり学習技法は、サポートベクターマシン、線形回帰、ロジスティック回帰、ナイーブベイズ、線形判別分析、決定木、k 最近傍アルゴリズム、及びニューラルネットワークの少なくとも 1 つを含み、前記教師なし学習技法は、階層クラスタリング、K 平均クラスタリング、K-NN (k 最近傍)、及び相関ルールの少なくとも 1 つを含む、請求項 3.0 に記載の方法。

30

【請求項 34】

前記グローバル意図言明、各ローカル意図言明、及び前記実体の前記詳細の 1 つ又は複数を受信することは、対話式質問表に回答してのものであり、前記対話式質問表は、前記コンピュータ実施ソリューションを構築するための構造化フォーマットでの質問を含む、請求項 1 に記載の方法。

【請求項 35】

自然言語文法に基づいて前記受信したグローバル意図言明、各ローカル意図言明、及び前記実体の前記詳細の 1 つ又は複数を変更することを更に含み、前記自然言語文法は、自然言語ライブラリを使用して、修正に適切な 1 つ又は複数の動詞及び前置詞を選ぶ、請求項 1 に記載の方法。

40

【請求項 36】

前記プロセッサにより、前記残りの数のイベントに必要な時間及びリソースの量を特定することと、

前記プロセッサにより、前記必要とされる時間及びリソースの量を表示することと、
を更に含み、請求項 2.2 に記載の方法。

【請求項 37】

50

前記プロセッサにより、前記潜在状態から前記現実状態に変化するとき、記録される2つの連続イベント間の最適アイドル時間をデータベースに記憶することと、

前記プロセッサにより、前記潜在状態から前記現実状態に変化するとき、記録される2つの連続イベントのそれぞれの間のアイドル時間をリアルタイムで特定することと、

前記プロセッサにより、前記特定されたアイドル時間と前記最適アイドル時間との比較に基づいてレポートを準備することと、
を更に含む、請求項2に記載の方法。

【請求項38】

前記プロセッサにより、前記受信した値をデータベースに記憶された最適値と比較することと、

前記プロセッサにより、前記比較に基づいて、前記受信した値が、前記コンピュータ実施ソリューションの構築に良好、不良、それとも曖昧であるかを判断することと、
を更に含む、請求項2に記載の方法。

【請求項39】

一意の識別子(ID)を各ローカル意図言明、各実体、及び各エージェントに割り当てることを更に含む、請求項1に記載の方法。

【請求項40】

前記ユーザが、リソースアイドル時間を最適化するように、全ての利用可能な実体の最適な使用を定義できるようにすることと、

リアルタイムで、前記トリガーCES内の各実体が他のローカル意図言明の前記1組のCESに影響を与えるのにかかった時間を監視するための、各実体に関連付けられた属性としての時間を追跡することと、

を更に含む、請求項1に記載の方法。

【請求項41】

1つ又は複数の結果可能性及び1つ又は複数の機会又は1つ又は複数のリスクの1つ又は複数の価値判断を提供し、前記1つ又は複数の結果可能性に対する1つ又は複数の確率を前記エージェントに割り当てることと、又は

1つ又は複数の行動を実行することであって、それにより、前記1つ又は複数の実体を改変することにより1つ又は複数の経路に介入して改変することであって、前記行動は、関連付けられたエージェントにより望まれる1つ又は複数のリソース最適化原理を駆動する、ことと、

を更に含む、請求項30に記載の方法。

【請求項42】

前記1つ又は複数の別個の経路を選び抜くための1つ又は複数の可能な経路を提供することを更に含み、前記1つ又は複数の可能な経路は、データベース中のライブラリからサーチされる、前記コンピュータ実施ソリューションと同様のソリューションのものである、請求項1に記載の方法。

【請求項43】

1つ又は複数の教師あり及び/又は1つ又は複数の教師なし機械学習法を前記データベース中の前記ライブラリに対して実施することを更に含み、前記1つ又は複数の教師あり及び/又は1つ又は複数の教師なし機械学習法の実施は、ソリューション内容を処理するNLPコンポーネント、ANNコンポーネント、及び最近傍コンポーネントを含むDLDEンジンによるものであり、前記DLDEンジンは、前記ライブラリの一部であるリッチソリューション内容をマイニングし、トランザクションクラスからトランザクションデータを抽出し、NSLソリューションライブラリの一部である前記リッチソリューション内容及び前記トランザクションクラスから抽出された前記トランザクションデータは、前記DLDEンジンに供給され、前記DLDEンジンは、種々のレベルでの距離の計算を可能にし、前記距離計算は、機械学習アルゴリズムを使用してNSL実体値、時間、及び空間を使用している、請求項42に記載の方法。

【請求項44】

10

20

30

40

50

動画及びユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリューションを構築する方法であって、

計算デバイスのプロセッサにより、前記動画の画像フレームを処理することであって、それにより、複数の差別化された画像フレームを識別し、差別化された各画像フレームは、その内容の違いに基づいて前記動画の1つ又は複数の前の画像フレームと異なるものとして識別され、前記内容は、前記自然言語の形態である、ことと、

前記プロセッサにより、前記動画の直前の画像フレームの内容に関する差別化された各画像フレームの内容の差に基づいて、それぞれ1つずつ複数のローカル意図言明を特定し、前記ローカル意図言明のそれぞれを潜在状態に設定することであって、各ローカル意図言明は、前記ソリューションを実行するための要件を履行するサブステップを示す文章であり、各ローカル意図言明は、データベースに記憶された自然言語データに関して前記画像フレームの内容を分析することにより特定される、ことと、

前記プロセッサにより、前記データベースに記憶された前記自然言語データに関して差別化された画像フレームの連続対間の前記画像フレームの内容を分析することにより、前記複数のローカル意図言明のそれぞれに関連付けられたN個の実体及び前記実体のそれぞれに関連付けられた属性を特定し、前記実体及び前記属性のそれぞれを潜在状態に設定することであって、各実体は、名詞句を含み、且つ対応するローカル意図言明により示される前記サブステップの前記要件の履行に参加し、前記属性は、前記実体のそれぞれの特性を定義し、且つ前記対応するローカル意図言明の他の実体から前記実体のそれぞれを差別化し、各属性は、形容詞句及び副詞句の少なくとも1つを含む、ことと、

前記プロセッサにより、最初の差別化された画像フレームと最後の差別化された画像フレームの内容の変化に基づいて、前記コンピュータ実施ソリューションのグローバル意図言明を特定し、前記グローバル意図言明を潜在状態に設定することであって、前記グローバル意図言明は、前記自然言語を使用して構築される前記ソリューションを示し、前記グローバル意図言明は、前記データベースに記憶された前記自然言語データに関して、前記最初の差別化された画像フレーム及び前記最後の差別化された画像フレームの前記内容を分析することにより特定される、ことと、

前記プロセッサにより、ローカル意図言明毎に1組の実体状態組合せ(CES)を形成することであって、前記1組のCESは、前記ローカル意図言明それぞれの前記N個の実体の 2^N 個の可能な組み合わせを含み、前記それぞれのローカル意図言明の全て(N個)の前記実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、前記1組中の各CESは潜在状態であり、且つ関連付けられた前記実体を現実状態に変えることに応答して現実状態に変わる、ことと、

ユーザから前記プロセッサにより前記自然言語の形態で、前記複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、前記エージェントの前記詳細を潜在状態に設定することであって、前記エージェントは人間エージェント及び機械エージェントの少なくとも一方であり、前記エージェントは、少なくとも、各属性、各実体、各ローカル意図言明及び前記グローバル意図言明の前記潜在状態を現実状態に変える、ことと、

前記ユーザから前記プロセッサにより前記自然言語の形態で、前記ローカル意図言明間の予め定義されたルール、制約及び公式のうち1つ又は複数に基づく複数の別個の関係を受信することであって、別個の各関係は、前記ソリューションを実行するための前記要件を履行する別個の経路であり、前記関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明の前記1組のCESに影響しているか、それとも前記コンピュータ実施ソリューションの前記構築の終わりであるかを示す、ことと、

前記潜在状態は、空のバイナリ状態であり、前記現実状態は、空ではないバイナリ状態である、方法。

【請求項45】

前記画像フレームを処理することは、前記画像フレームの内容及び前記画像フレームに

10

20

30

40

50

わたる前記内容の変化に基づいて情報を抽出することを含み、前記情報を抽出することは、物体認識技術、文字認識技術、音声認識技術、又はそれらの組合せに基づく、請求項 4.4 に記載の方法。

【請求項 4.6】

前記グローバル意図言明及び前記複数のローカル意図言明のそれぞれを特定することは、前記内容の前記変化に 1 つ又は複数の動詞をタグ付けすることを含む、請求項 4.4 に記載の方法。

【請求項 4.7】

各ローカル意図言明の実体毎に、

前記関連付けられたエージェントから前記プロセッサにより前記自然言語の形態で、前記実体のそれぞれに対する値を受信することを更に含み、前記実体のそれぞれに対する前記値を受信することは、前記受信した値に基づいて前記実体のそれぞれを前記潜在状態から現実状態に変えるイベントの記録である、請求項 4.4 に記載の方法。

10

【請求項 4.8】

各実体の属性毎に、

前記関連付けられたエージェントから前記プロセッサにより前記自然言語の形態で、前記属性のそれぞれに対する値を受信することを更に含み、前記属性のそれぞれに対する前記値を受信することは、前記受信した値に基づいて前記属性のそれぞれを前記潜在状態から現実状態に変えるイベントの記録である、請求項 4.4 に記載の方法。

【請求項 4.9】

前記プロセッサにより、ログイン詳細、バイOMETリック詳細、顔認識技法、及び網膜検出技法の 1 つに基づいて前記関連するエージェントを認証することを更に含む、請求項 4.7 に記載の方法。

20

【請求項 5.0】

各ローカル意図言明に関連付けられた実体のそれぞれについての前記イベントの記録時に、前記プロセッサにより、前記ローカル意図言明のそれぞれを前記潜在状態から現実状態に変えるイベントを記録することを更に含む、請求項 4.7 に記載の方法。

【請求項 5.1】

前記ローカル意図言明のそれぞれを前記潜在状態から前記現実状態に変える前記イベントを記録する際に、前記プロセッサにより、前記グローバル意図言明を前記潜在状態から現実状態に変えるイベントを記録することを更に含む、請求項 5.0 に記載の方法。

30

【請求項 5.2】

前記自然言語は、1 つ若しくは複数の母国語、1 つ若しくは複数の手話、1 つ若しくは複数の記号、1 つ若しくは複数の数字、又はそれらの組合せに基づく、請求項 4.4 に記載の方法。

【請求項 5.3】

標準作業手順書 (SOP) ファイル及びユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリューションを構築する方法であって、前記 SOP ファイルは規範的言明、記述的言明、及びフローチャートを含み、前記方法は、

計算デバイスのプロセッサにより、前記 SOP ファイルをパースすることであって、それにより、前記規範的言明、前記記述的言明、及び前記フローチャートを前記自然言語の形態で識別することと、

40

前記プロセッサにより、前記規範的言明のそれぞれについてそれぞれ 1 つずつ複数のローカル意図言明を特定し、前記ローカル意図言明のそれぞれを潜在状態に設定することであって、各ローカル意図言明は、前記ソリューションを実行するための要件を履行するサブステップを示す文章であり、各ローカル意図言明は、データベースに記憶された自然言語データに関して対応する規範的言明を分析することにより特定される、ことと、

前記プロセッサにより、前記データベースに記憶された自然言語データに関して前記対応する規範的言明に関連付けられた前記記述的言明を分析することにより、前記複数のローカル意図言明のそれぞれに関連付けられた N 個の実体及び前記実体のそれぞれに関連付

50

けられた属性を特定し、前記実体及び前記属性のそれぞれを潜在状態に設定することであって、各実体は、名詞句を含み、且つ前記対応するローカル意図言明により示される前記サブステップの前記要件の履行に参加し、前記属性は、前記実体のそれぞれの特性を定義し、且つ前記対応するローカル意図言明の他の実体から前記実体のそれぞれを差別化し、各属性は、形容詞句及び副詞句の少なくとも1つを含む、ことと、

前記プロセッサにより、ローカル意図言明毎に、1組の実体 - 状態組合せ (CES) を形成することであって、前記1組のCESは、前記ローカル意図言明それぞれの前記N個の実体の 2^N 個の可能な組み合わせを含み、前記それぞれのローカル意図言明の全て (N個) の実体に基づいて形成されるCESは、トリガー実体状態組合せ (トリガーCES) であり、前記1組中の各CESは、潜在状態に設定され、且つ関連付けられた前記実体を現実状態に変えることに応答して現実状態に変わる、ことと、

前記プロセッサにより、前記フローチャートに基づいて前記ローカル意図言明間の予め定義されたルール、制約及び公式のうちの1つ又は複数に基づく複数の別個の関係を特定することであって、別個の各関係は、前記ソリューションを実行するための前記要件を履行する別個の経路であり、前記関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明の前記1組のCESに影響しているか、それとも前記コンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを示し、前記グローバル意図言明は、前記自然言語を使用して構築される前記ソリューションを示し、前記関係は、前記データベースに記憶された前記自然言語データに関して前記フローチャートを分析することにより特定される、ことと、

ユーザから前記プロセッサにより前記自然言語の形態で、前記複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、前記エージェントの前記詳細を潜在状態に設定することであって、前記エージェントは人間エージェント及び機械エージェントの少なくとも一方であり、前記エージェントは、少なくとも、各属性、各実体、及び各ローカル意図言明の前記潜在状態を現実状態に変える、ことと、

を含み、

前記潜在状態は、空のバイナリ状態であり、前記現実状態は、空ではないバイナリ状態である、方法。

【請求項54】

前記複数のローカル意図言明を特定すること、前記実体及び前記属性を特定すること、並びに前記関係を特定することのそれぞれは、

前記規範的言明、前記記述的言明、前記フローチャートの内容における発話部分を識別することと、

前記内容中の同一指示、特に先行詞による代名詞を解決することと、

継承関係をトラバースして、前記内容中の曖昧性をなくすことと、

前記分析において曖昧性がある場合、前記内容の1つ又は複数の部分をマークすることと、

前記1つ又は複数の曖昧な部分の曖昧さをなくすユーザ入力を受信することと、

を含む、請求項53に記載の方法。

【請求項55】

前記実体を特定することは、前記記述的言明内の名詞句に基づく、請求項53に記載の方法。

【請求項56】

前記SOPファイルをパースすることは、分散ドキュメントパースシステムを使用しており、

前記分散ドキュメントパースシステムは、コアパス、参照パス、関係パス、理解パス、プロセスパス、エンリッチメントパス、及びコンプライアンスパスの少なくとも1つを含む、請求項53に記載の方法。

【請求項57】

前記自然言語は、1つ若しくは複数の母国語、1つ若しくは複数の手話、1つ若しくは

10

20

30

40

50

複数の記号、1つ若しくは複数の数字、又はそれらの組合せに基づく、請求項 5.3 に記載の方法。

【請求項 5.8】

計算デバイスのプロセッサにより、自然言語ベースの前記コンピュータ実施ソリューションをパースすることであって、前記コンピュータ実施ソリューションは、前記グローバル意図言明と、前記複数のローカル意図言明と、前記複数のローカル意図言明のそれぞれに関連付けられた前記N個の実体及び前記エージェントと、前記実体のそれぞれに関連付けられた前記属性と、前記複数のローカル意図言明間の前記複数の別個の関係と、各ローカル意図言明に関する前記1組の実体状態組合せ(CES)と、前記1つ又は複数のトリガーCESを含む、ことと、

10

計算デバイスのプロセッサにより、前記自然言語ベースのコンピュータ実施ソリューションをパースすることに基づいて、複数の自然言語ソリューションコンポーネントを抽出することであって、前記自然言語ソリューションコンポーネントは、前記グローバル意図言明と、前記複数のローカル意図言明と、前記実体のそれぞれに関連付けられた前記属性と、前記複数のローカル意図言明間の前記複数の別個の関係と、前記実体状態組合せ(CES)と、前記1つ又は複数のトリガーCESを含む、ことと、

データベースに記憶されたプログラミング言語の記号、キーワード、演算子、及び関数への前記自然言語ソリューションコンポーネントのマッピングに基づいて、前記プログラミング言語のコードを自動生成することと、

を含む、請求項 1 に記載の方法。

20

【請求項 5.9】

ユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスであって、

プロセッサと、

前記プロセッサに結合されたメモリと、

を含み、前記メモリは、請求項 1 ~ 4.3 の何れか1項に記載の方法を実行するように前記プロセッサにより実行可能な命令を含む、計算デバイス。

【請求項 6.0】

動画及びユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスであって、

30

プロセッサと、

前記プロセッサに結合されたメモリと、

を含み、前記メモリは、請求項 4.4 ~ 5.2 の何れか1項に記載の方法を実行するように前記プロセッサにより実行可能な命令を含む、計算デバイス。

【請求項 6.1】

標準作業手順書(SOP)ファイル及びユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスであって、前記SOPファイルは規範的言明、記述的言明、及びフローチャートを含み、前記計算デバイスは、

プロセッサと、

前記プロセッサに結合されたメモリと、

を含み、前記メモリは、請求項 5.3 ~ 5.7 の何れか1項に記載の方法を実行するように前記プロセッサにより実行可能な命令を含む、計算デバイス。

40

【請求項 6.2】

ユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリューションを構築する命令を記憶した非一時的コンピュータ可読媒体であって、前記命令は機械実行可能コードを含み、前記機械実行可能コードは、プロセッサにより実行されると、前記プロセッサに、請求項 1 ~ 4.3 の何れか1項に記載の方法を実行させる、非一時的コンピュータ可読媒体。

【請求項 6.3】

動画及びユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリ

50

ューションを構築する命令を記憶した非一時的コンピュータ可読媒体であって、前記命令は機械実行可能コードを含み、前記機械実行可能コードは、プロセッサにより実行されると、前記プロセッサに、請求項44～52の何れか1項に記載の方法を実行させる、非一時的コンピュータ可読媒体。

【請求項64】

標準作業手順書(SOP)ファイル及びユーザにより理解され、入力される自然言語を使用してコンピュータ実施ソリューションを構築する命令を記憶した非一時的コンピュータ可読媒体であって、前記SOPファイルは規範的言明、記述的言明、及びフローチャートを含み、前記命令は機械実行可能コードを含み、前記機械実行可能コードは、プロセッサにより実行されると、前記プロセッサに、請求項53～57の何れか1項に記載の方法

10

【発明の詳細な説明】

【技術分野】

【0001】

【0001】本願は、2019年1月10日付けで出願された米国特許出願第201941001135号及び2019年7月16日付けで出願された米国特許出願第201941028675号と同種であり、これらは全体的に、参照により本明細書に援用される。

【0002】

技術分野

【0002】本開示は情報技術の分野に関する。より詳細には、本開示は、ソリューションを設計し展開するコンピュータ実施方法、システム、及びコンピュータ可読媒体に関する。

20

【背景技術】

【0003】

背景

【0003】ソフトウェアのエンジニアリング、設計、及びアーキテクチャ実施は、ここ60年にわたりかなり大きく変わり進化した。簡潔にするために、コンピュータへの動作論理の伝達における多くのレベルの抽象化は2つのグループにグループ化することができる：(1)プログラミング言語を通して付与される高水準アプリケーション論理及び(2)コンピュータ及びそれらの動作を仲介又はより直接的に対応するオペレーティングシステム以下。

30

【0004】

【0004】主に3つの普及しているオペレーティングシステムがある：Microsoft Windows、Apple Mac OS X、及びLinux。しかしながら、数百ものプログラミング言語がある。一般に、全てのプログラミング言語は専らコード(自然言語及び算術から離れた特定の意味及び機能を有するシンボル)により駆動される。ソフトウェア専門家及び技術者が何れかのプログラミング言語を習得し、ソリューションの作成又は保守に参加するには数週間から数ヶ月かかる。結果として、ソリューションのユーザは一般に、技術専門家の介入なしではソリューションを作成又は状況により変更することができない。

【発明の概要】

【発明が解決しようとする課題】

40

【0005】

【0005】ここ何年も情報技術の背後の計算力及び通信力は非常に成長した。一般に、ソフトウェア設計及び管理技法は、コンポーネントベースの設計、サービス指向アーキテクチャ、ウェブサービス、ビジネスプロセス管理、及びアジャイルプロジェクト管理方法論に向かう動きに伴って改良された。同時に、可動部品の数が増加するにつれて、技術の大勢は複雑になった。一般的なユーザは、彼らができることと比較して、ソリューション設計に直接参加したり、ソリューション設計に継続的に影響を及ぼす立場にないため、コンピュータから依然として遠ざけられている。このようにユーザが技術エキスパートや仲介者に過度に依存しているのは、プログラミング言語が自然言語と大きく異なることに起因している。現行水準の技術システムは、コンピュータ言語が自然言語と略同じである技術

50

的、創作的、及び革新的な機会を実現することができていない。

【課題を解決するための手段】

【0006】

図面の簡単な説明

[0006] 本開示の特徴、態様、及び利点は、添付図面を参照して以下の詳細な説明が読まれた場合、よりよく理解されよう。

【図面の簡単な説明】

【0007】

【図1】[0007]状態の潜在性及び現実性の一例である。

【図2】[0008]何か異なるものが実体である一例を表す。

10

【図3】[0009]ペン及び紙が別個の異なる実体である分化実体の一例である。

【図4】[0010]2本のペンは別個であるが、それでもなお他と同じ実体である未分化実体の一例である。

【図5】[0011]ソリューションがユーザにより構築される潜在実体状態の一例である。

【図6】[0012]ソリューションエコシステムの状況での適格実体の一例である。

【図7】[0013]イベントが到着した現実実体状態の一例である。

【図8】[0014]最高一般化レベルにおける全能実体の一例である。

【図9】[0015]属性を一緒にドラッグする独立実体の一例である。

【図10】[0016]黙示実体の一例である。

【図11】[0017]ソリューション設計者が、潜在又は現実のバイナリ状態でのみ考慮する凍結実体の一例である。

20

【図12】[0018]実体の作成及び削除の一例である。

【図13】[0019]実体の変更の一例である。

【図14】[0020]存在についての何らかの他の実体への属性の依存性を表す。

【図15】[0021]実体に関して常に存在する必須属性の一例である。

【図16】[0022]ソリューション環境に基づく属性の標準化の一例である。

【図17】[0023]実体がソリューション環境内で置かれた状況に基づいて使用に選択される一般属性の一例である。

【図18】[0024]属性のレベルの一例である。

【図19】[0025]自然界に物理的に存在する現実実体の一例である。

30

【図20】[0026]他の現実実体又は他の具象実体を表す実体の一例である。

【図21】[0027]人間エージェントにより彼ら自身の脳で自然に所与のニューロン状態の形態で作成される実体の一例である。

【図22】[0028]情報実体の一例である。

【図23】[0029]真理値についての理解の一例である。

【図24】[0030]物理世界で物理的に存在している全ての実体の一例である。

【図25】[0031]自然言語の形態のローカル意図言明及びグローバル意図言明の一例である。

【図26】[0032]変化単位のサイズの一例である。

【図27】[0033]実体のバイナリ状態の一例である。

40

【図28】[0034]変異性の一例である。

【図29】[0035]状態が常に同じままである実体の一例である。

【図30】[0036]入力及び出力の一例である。

【図31】[0037]自然言語及び自然言語ソリューションが機能する方法の一例である。

【図32】[0038]意図を表現するだけで、現実に変形する能力を有さない静的意図言明の一例である。

【図33】[0039]実体を生じさせる基礎変形である動的意図言明の一例である。

【図34】[0040]分化原理の一例である。

【図35】[0041]視座の一例である。

【図36】[0042]分化ツリーを上又は下に移動する状況での方向性の一例である。

50

【図37】[0043]非トリガー及びトリガー実体状態組合せ（CES）の一例である。

【図38】[0044]多くの可能性間で行われる選択に課される制限にすぎない制約の一例である。

【図39】[0045]殆どの場合、システムに課される一連の制約であるルールの一例である。

【図40】[0046]ルールと同じであるが、多くの場合、情報システムの状況で使用されるアルゴリズムの一例である。

【図41】[0047]人間エージェントの一例である。

【図42】[0048]値クリエータ及びコンシューマの一例である。

【図43】[0049]エージェントに関する実体の所有権の一例である。

10

【図44】[0050]マインド機能、身体機能、及び情報機能の一例である。

【図45】[0051]人間エージェントの情報権の一例である。

【図46】[0052]人間エージェントの決定権の一例である。

【図47】[0053]機械エージェントの一例である。

【図48】[0054]変形に影響を及ぼすことが可能な第3の種類のエージェントであるマザーネイチャーの一例である。

【図49】[0055]判決システムの一例である。

【図50】[0056]測定フレームワークの一例である。

【図51】[0057]エージェントに関する実体の一例である。

【図52】[0058]自然言語と自然言語ソリューション（NSL）との間の共通性の一例である。

20

【図53】[0059]コンポーネントの再使用可能性の一例である。

【図54】[0060]リエンジニアリングの一例である。

【図55】[0061]NSLライブラリの一例である。

【図56】[0062]共有実体の一例である。

【図57】[0063]ステークホルダエンゲージメントセンタ（SEC）の一例である。

【図58】[0064]コードと自然言語との間の差の一例である。

【図59】[0065]意味と価値との間の差の一例である。

【図60】[0066]経過時間の一例である。

【図61】[0067]価値の定量化の一例である。

30

【図62】[0068]価値の相対性の一例である。

【図63】[0069]論理の段落の一例である。

【図64】[0070]人工ニューラルネットワークの一例である。

【図65】[0071]APIとプログラムとの間の差の一例である。

【図66】[0072]ITにおける抽象化のレベルについての理解の一例である。

【図67】[0073]ソリューションの全ライフサイクルの一例である。

【図68】[0074]NSLが言語に依存しないことの一例である。

【図69】[0075]NSL技術フレームワークの一例である。

【図70】[0076]分散ミッションコントロールセンタの概念の一例である。

【図71】[0077]動的ミッションコントロールセンタの一例である。

40

【図72】[0078]NSLが通信媒体に依存しないことの一例である。

【図73】[0079]DNAが情報に変換されることの一例である。

【図74】[0080]テキストベースユーザインターフェース（TUI）の一例である。

【図75】[0081]NSL技術翻訳フレームワーク（TTF）及びNSL技術再翻訳フレームワーク（TRF）の一例である。

【図76】[0082]クエリベースソリューションデベロップメント（QBSD）の一例である。

【図77】[0083]NSLのイニシアチブ様の自然言語ソリューションの一例である。

【図78】[0084]NSLにおける自動動詞割り当て（AVA）の一例である。

【図79】[0085]プラクティスベースソリューションデベロップメント（PBSD）の一例である。

50

例である。

【図 8 0】[0086]N S Lにおける基盤独立性の一例である。

【図 8 1】[0087]ソリューションの定量化の一例である。

【図 8 2】[0088]冗長性の識別を通じた論理圧縮の一例である。

【図 8 3】[0089]N S L原理をエンジニアリング設計に適用することの一例である。

【図 8 4】[0090]処理とサーチベース手法との間の差の一例である。

【図 8 5】[0091]コンテナの一例である。

【図 8 6】[0092]N S Lフォーマットに変換されているドキュメントの一例である。

【図 8 7】[0093]センススペースユーザインターフェースの一例である。

【図 8 8】[0094]フローチャート論理変換の一例である。

10

【図 8 9】[0095]リソース最適化及びアイドル時間識別フレームワークの一例である。

【図 9 0】[0096]メタソリューション論理実体及びトランザクションソリューション論理実体の一例である。

【図 9 1】[0097]ユーザが理解する自然言語を使用して、プログラムコードを使用せずにコンピュータ実施ソリューションを構築する方法を表すフローチャートを表す。

【図 9 2】[0098]プログラミングコードを自然言語に変換して、自然言語ベースのコンピュータ実施ソリューションを構築する方法を示すフローチャートを表す。

【図 9 3】[0099]自然言語に基づいて構築されたコンピュータ実施ソリューションをプログラミングコードに変換する方法を示すフローチャートを表す。

【図 9 4】[00100]N S Lの一般化コンピュータネットワーク構成を表す。

20

【図 9 5】[00101]Javaプログラム及び対応するN S Lソリューションの表現の一例を示す図である。

【図 9 6】[00102]技術翻訳フレームワークを使用したJavaプログラムからN S Lソリューションへの翻訳の表現の一例を示す図である。

【図 9 7 A】[00103]JavaプログラムからN S L均等物への1対1マッピングを通しての属性を有する実体作成の表現の一例を示す図である。

【図 9 7 B】[00104]JavaプログラムからN S L均等物への1対1マッピングを通してのN S Lにおける関数の表現の一例を示す図である。

【図 9 8 A】[00105]ソリューションが構築されるとき、静的状態の段落の表現の一例を示す図である。

30

【図 9 8 B】[00106]「顧客情報が捕捉されたとき、イベント到来」として静的から動的に変わる文章の表現の一例を示す図である。

【図 9 8 C】[00107]情報が満杯になったとき、動的状態である段落の表現の一例を示す図である。

【図 9 9 A】[00108]Javaプログラムの入力及び出力の表現の一例を示す図である。

【図 9 9 B】[00109]N S Lソリューションの入力及び出力の表現の一例を示す図である。

【図 1 0 0 A】[00110]管理データベースの表現の一例を示す図である。

【図 1 0 0 B】[00111]管理ユーザインターフェースの表現の一例を示す図である。

【図 1 0 0 C】[00112]実体をエージェントに接続する表現の一例を示す図である。

40

【図 1 0 1】[00113]N S L及びJavaでホテル予約システムソリューションを構築するのに必要な時間、労力、及びコストを説明する動作メトリックの表現の一例を示す図である。

【図 1 0 2 A】[00114]キーワード「C h a r」がJavaで使用される方法及びN S Lで表される方法の表現の一例を示す図である。

【図 1 0 2 B】[00115]キーワード「S t r i n g」がJavaで使用される方法及びN S Lで表される方法の表現の一例を示す図である。

【図 1 0 2 C】[00116]キーワード「D o u b l e」がJavaで使用される方法及びN S Lで表される方法の表現の一例を示す図である。

【図 1 0 2 D】[00117]キーワード「F o r」がJavaで使用される方法及びN S Lで表される方法の表現の一例を示す図である。

50

【図102E】[00118]キーワード「While」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図102F】[00119]キーワード「If」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図102G】[00120]キーワード「Break」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図103】[00121]プロセスを情報に変換する表現の一例を示す図である。

【図104】[00122]NSL技術翻訳フレームワーク方法論の表現の一例を示す図である。

【図105】[00123]Pythonプログラム及びNSLソリューションの表現の一例を示す図である。

10

【図106】[00124]少数のJava構造体及び対応するNSL均等物の例を示す表である。

【図107A】[00125]キーワード「Static」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107B】[00126]キーワード「Switch」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107C】[00127]キーワード「Continue」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107D】[00128]キーワード「Transient」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

20

【図107E】[00129]キーワード「Long」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107F】[00130]キーワード「Short」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107G】[00131]キーワード「Byte」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107H】[00132]キーワード「Default」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107I】[00133]キーワード「Super」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

30

【図107J】[00134]キーワード「Protected」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107K】[00135]キーワード「This」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107L】[00136]キーワード「Synchronized」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107M】[00137]キーワード「Strictfp」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【図107N】[00138]キーワード「final」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

40

【図108A】[00139]Pythonプログラムの入力及び出力の表現の一例を示す図である。

【図108B】[00140]NSLソリューションの入力及び出力の表現の一例を示す図である。

【図109】[00141]NSL技術再翻訳フレームワーク方法論の表現の一例を示す図である。

【図110】[00142]入力及び出力を有するNSLソリューション及び同等のPython及びCプログラムの表現の一例を示す図である。

【図111】[00143]共通論理ツリーにより生成された出力の一例を示す図である。

【図112】[00144]NSLからプログラミング言語へのコード生成の一例を示す図である。

50

【図 1 1 3】[00145]ソリューションを設計し展開するシステムを表す。

【図 1 1 4】[00146]プログラミング言語コードを自然言語に翻訳し変換するシステムを表す。

【図 1 1 5】[00147]NSLで構築されたソリューションをプログラミング言語に翻訳し変換するシステムを表す。

【図 1 1 6】[00148]DLL情報フロー及びDLLエンジンに存在するコンポーネントの一例を示す図である。

【図 1 1 7】[00149]SOPをNSLに変換する一例を示す図である。

【図 1 1 8】[00150]順次変化単位の一例を示す図である。

【図 1 1 9】[00151]CU2又はACU2の何れか一方がCU3をトリガーし、他方が消失する代替の変化単位の一例を示す図である。

10

【図 1 2 0】[00152]CU2及びPCU2の両方がCU3をトリガーするものとする並列変化単位の一例を示す図である。

【図 1 2 1】[00153]下位変化単位の一例を示す図である。

【図 1 2 2】[00154]一時的な変化単位の一例を示す図である。

【図 1 2 3】[00155]組込下位変化単位の一例を示す図である。

【図 1 2 4】[00156]ソリューションクラス及びトランザクションクラスの一例を示す図である。

【図 1 2 5】[00157]コンテキストトリプルアイデンティティの一例を示す図である。

【図 1 2 6】[00158]ハブとして作用するNSLの一例を示す図である。

20

【図 1 2 7】[00159]プログラミングコードを使用せずに、ビデオ及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する方法を示すフローチャートを表す。

【図 1 2 8】[00160]プログラミングコードを使用せずに、標準作業手順書(SOP)ファイル及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する方法を示すフローチャートを表す。

【発明を実施するための形態】

【0008】

詳細な説明

【00161】システム、デバイス又は装置、及び方法は例及び実施形態として本明細書で説明されるが、ソリューションを提供するシステム及び方法が、説明される実施形態又は図面に限定されないことを当業者ならば認識する。図面及び説明は、開示される特定の形態への限定を意図しないことを理解されたい。むしろ、意図は、添付の特許請求の範囲の趣旨及び範囲内にある全ての変更、均等物、及び代替を包含することである。本明細書で使用されるあらゆる見出しは、単に編成を目的とし、説明の範囲又は特許請求の範囲の限定を意味しない。本明細書で使用される場合、「し得る(may)」という言葉は、義務の意味(例えば、しなければならないことを意味する)ではなく許容の意味(例えば、～する潜在性を有することを意味する)で使用される。同様に、「含む(include)」、「含んでいる(including)」、及び「含む(includes)」という言葉は、～を含むがそれに限定されないことを意味する。

30

【0009】

【00162】以下の説明は、本願出願時に発明者に知られている本開示の実行に最良であると現在考えられている方法及びシステム、デバイス、又は装置の詳細で情報価値のある説明である。当然ながら、以下の説明、添付図面、及び添付の特許請求の範囲に鑑みて、多くの変更及び適応が当業者に容易に明らかになる。本明細書に記載されるシステム、デバイス、又は装置及び方法は、特定の具体性をもって提供されるが、本技法は、ユーザのニーズに応じて、より大きな具体性又はより少ない具体性をもって実施し得る。さらに、本技法の特徴によっては、以下の段落で説明される他の特徴の対応する使用なしで有利に使用され得るものがある。したがって、本説明は、本技法の原理の単なる例示として見なされるべきであり、本技法の原理の限定として見なされるべきではなく、本技法は特許

40

50

請求の範囲によってのみ規定される。

【0010】

【00163】 前付けとして、以下の考察及び添付の特許請求の範囲における「又は」という用語の定義は、包含的「又は」であることが意図される。すなわち、「又は」という用語は、2つの相互に排他的な代替の区別を意図しない。むしろ、「又は」という用語は、2つの要素間の接続詞として利用される場合、一方の要素を単独で包含すること、他方の要素を単独で包含すること、並びに両要素の組合せ及び順序を包含することとして定義される。例えば、「A」又は「B」という用語を利用した考察又は記述は、「A」単独、「B」単独、及び「AB」及び/又は「BA」等のそれらの任意の組合せを含む。本考察が例示的な実施形態に関し、添付の特許請求の範囲は、本明細書で考察される実施形態に限定されるべきではないことに留意する価値がある。

10

【0011】

【00164】 本明細書における説明では、プロセッサは、マイクロプロセッサ、マイクロコンピュータ、マイクロコントローラ、デジタル信号プロセッサ、中央演算処理装置、状態機械、論理回路、及び/又は演算命令に基づいて信号を操作する任意のデバイスとして実施し得る。機能の中でも特に、プロセッサは、プロセッサに結合された非一時的コンピュータ可読記憶媒体に記憶されたコンピュータ可読命令をフェッチし実行し得る。非一時的コンピュータ可読記憶媒体は、例えば、揮発性メモリ（例えばRAM）及び/又は不揮発性メモリ（例えば、EPROM、フラッシュメモリ、NVRAM、メモリスト等）を含み得る。

20

【0012】

【00165】 本明細書における説明では、メモリは、計算デバイスのメモリであり得、例えば、揮発性メモリ（例えばRAM）及び/又は不揮発性メモリ（例えば、EPROM、フラッシュメモリ、NVRAM、メモリスト等）を含め、任意の非一時的コンピュータ可読記憶媒体を含み得る。

【0013】

【00166】 本明細書における説明では、モジュールは、とりわけ、特定のタスクを実行し、又は特定のデータ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造等を含む。モジュールは、計算デバイスでアプリケーションを補足するモジュール、例えばオペレーティングシステムのモジュールを更に含む。

30

【0014】

自然言語ソリューション（NSL）に関する用語

【00167】 以下に列記する各用語は、コンピュータ実施NSL方法論に関して特定の役割及び適用を有する。これらの個々の技術要素及び方法論要素は、NSLが生み出す一般的な新しいパラダイムに関して説明される役割を有する。

【0015】

【00168】 自然言語ソリューション（NSL）は、自然言語のように機械と通信することにより、プログラミング言語に効果的に取って代わると主張される。自然言語ソリューション（NSL）が変革的技術効果を生じさせるためには、コンピュータ実施方法の適用が求められる。この方法は、状況により、実体（エンティティ）及びそれらの関係への敏感な手法を必要とする。実体及びそれらの関係には様々なバリエーションがあり、それらのバリエーションのそれぞれ1つを適宜定義し適宜扱う必要がある。

40

【0016】

【00169】 十分に理解されるようにNSLの働きを説明するために、以下のステップがとられた：a) 関連する全てのNSL用語のリストを提供し、これらは、NSLのよりよい理解のために、NSLの特定の状況で理解される必要がある；b) 各用語を適宜定義又は説明した；c) 図面により用語を適宜補足し、適宜、個々の図面がぴったり合うわずかな統合レベル図面を提供した；及びd) 必要な場合、用語を例によって補足した。

【0017】

【00170】 ソリューション設計：NSLの状況では、ソリューション設計とは、実体の

50

潜在性状態を個別に、そして実体状態組合せの形式で設定することであり、実体ソリューション状態組合せソリューションへのアクセスは、操作時に制御された変更の連鎖を通して行われる。

【0018】

[00171] 実体：別個のものはすべて実体である。すなわち、それ自体の立場を有し、情報の観点から表現できるものはすべて実体である。言語では、実体は単語、記号、又は数字で表される。例：車が実体として適格であるのと全く同じように、砂粒も実体として適格である。

【0019】

[00172] 分化実体：別個であり、それと同時に他の実体と異なるものはすべて分化実体である。すなわち、分化実体は、何らかの他の実体と比較して、異なる。自然言語では、これらは単語によって表される。例：「ペン」は「紙」と異なる。

10

【0020】

[00173] 未分化実体：別個であり、且つ何らかの他の1つ又は複数の実体と同じであるものはすべて、それ(ら)の1つ又は複数の他の実体から未分化であると見なされる。そのような「再帰」は空間及び時間において発生する。これらの再帰は、数学の領域に入り、数字で表される。ソリューション設計の観点から、結果に影響せずに1つの実体を何らかの他の実体で有効に置換することができる場合、そのような実体は空間又は時間の何れかにおいて再帰と見なされる。例：ペンと紙が卓上にある場合、卓上に「ペンと紙がある」と言う。しかし、1本のペンと「もう1本のペン」が卓上にある場合、「卓上に2本のペンがある」と言う。

20

【0021】

[00174] 世界の仕組み(WWW: Way World Works)原理：単なる従来の技術ではなく、NSLは、ソリューションアーキテクチャにおいて新しいパラダイムを生み出す、科学と技術からの理解の融合である。WWWとは、今日の科学により理解され認識されている世界又は自然が機能する仕組みを導く原理である。NSL論理は、世界の仕組みに関して全ての科学的洞察を利用し、人間エージェントがコンピュータの使用を通して探し求めるソリューションに関して特定の革新的な方法論をもらす。例：全ての物事は粒子から構成されること、粒子が結合されると、創発特性が生じること、全ての物事は時間及び空間で生じること、全てのイベントはエネルギーにより駆動されること、その他諸々のことが世界の仕組みである。そしてこれらは、エージェントが何をするか、そして、それらが方向性のある変化を介してある所望の状態から別の所望の状態への動きをどのように引き起こすかに関係する。

30

【0022】

[00175] 潜在実体vs非潜在実体：ソリューション設計には実体及びそれらの関係が関わる。これらの実体は、ソリューション設計のコンテキストで現実世界から選択される。ソリューションに関して有する潜在性で選択された実体は、「潜在実体」と宣言される。「潜在性セット」のソリューション一部ではない実体は除外され、ソリューションエコシステムの観点から関連性がない。

【0023】

[00176] 潜在実体状態：ソリューションエコシステムは、「潜在実体」と呼ばれる「ソリューション設計者」により「現実世界」から「ソリューション世界」に選択された情報実体で構成される。これらは、「重要な実体」、「適格な実体」、又はソリューションエコシステムに「影響する潜在性」を有する実体である。換言すれば、「潜在実体」は、ソリューションエコシステムでは存在する可能性があるか、又は存在することが認められているが、現実には存在しない。例：ある人はクラブのメンバとして認められている。そのメンバがクラブに参加することができる潜在性がある。メンバの参加は、クラブのメンバシップを有する場合のみ可能である。その一方で、そのメンバは、所与の時点でクラブに存在している可能性もあれば、又は存在していない可能性もある。存在していない場合、そのメンバは「潜在」状態にあり、存在している場合、そのメンバは「現実」状態にあ

40

50

る。

【 0 0 2 4 】

[00177] 現実実体状態：現実状態とは、実体が（個別にまたは組合せ形式で）存在することができ、且つ実際に存在している状態である。例：クラブに参加する資格を有するクラブのメンバが姿を現す。

【 0 0 2 5 】

[00178] ソリューションエコシステム：ソリューション設計者は、「現実世界」から潜在実体を選択して使用し、潜在実体間の関係を確立する。潜在実体間の関係は、確立された変化の経路に沿ってソリューションを得ることができる方法で結合し、相互作用するようなものである。

10

【 0 0 2 6 】

[00179] 全能実体：これらは、「独立した実体」ステータスのみが付与される最も一般化された又は最も「未分化」な実体である。例：「部屋」は実体であり、「鍵」は実体である。それらの全能形態では、これらは2つの実体としてカウントされる。

a . 実体は、ますます多くの実体と組合せが発生するにつれて、分化する。例：ある人は一般化状態であり、70億人の人々及びこれまでに生きた人々のうちの一人である。この実体人が空間、時間、色、及び人種等の属性及び他の実体と組み合わせられると、その人はますます分化する。

b . ソリューションエコシステムでは、すべての実体が一般化された全能レベルである場合、有するものは、ソリューションエコシステムに参加した実体の総数のカウントだけである。これは、分化細胞になる前の細胞の状態に類似する。

20

【 0 0 2 7 】

[00180] 独立実体：これらは、バイナリイベントが生じるレベルにある実体である。すなわち、潜在と現在との間の切り替えが行われると、それらが参加する実体状態組合せが変わる例：ペンは存在に移ることができ、又は存在から消失に移ることができる。独立実体イベントが生じると、参加している組合せエコシステムの状態が変わる可能性がある。潜在実体として紙と共に存在している場合、それらが一緒になって 2^2 、すなわち4つの潜在組合せ状態を生じさせる可能性がある。

【 0 0 2 8 】

[00181] しかしながら、独立実体は、その属性（依存実体）と一緒にドラッグし、イベントがその属性のレベルで生じる場合、多くの方法で更に変更されることができる。

30

a . 黙示実体：多くの場合、他の実体に接続された実体は暗黙的に無視される。例：ある人が通りに入る場合、その人が着ている衣服に関連してそこにいることが黙示される。衣服は黙示される。同様に、人間エージェントが呼吸する空気があることが黙示される。部屋の予約の場合、情報を入力する「エージェント」の存在が黙示され得る。ソリューションの設計者は、そのような黙示実体を当たり前として解釈する。大半の場合、属性であっても黙示される。全ての変化は、「変化単位」（S I）内で生じる。変化は、物理的相互作用が空間及び時間において生じる場合のみ生じる。1つの独立実体の空間及び時間が既知である場合、他の実体の空間及び時間属性は黙示され得る。さらには、ソリューション設計者にとって、これらの属性を指定しなくても設計の本質が変わらず、設計者がそれらの属性を指定しなくなる場合もあり得る。

40

b . 凍結実体：ソリューションアーキテクチャノ設計は、制御された分化の原理に基づく。ソリューション設計者が、ソリューションに参加する潜在実体としてある実体を選択する場合、選択された実体は既に高度に分化した状態にある。人間エージェントのレベルでは、全ての独立実体のそれぞれ1つは膨大な数の粒子（原子又は素粒子）で構成されているため、全ての独立実体は既に高度に分化した状態にある。ソリューション設計者はそれらを潜在又は現実のバイナリ状態にあるものと見なすため、それらは凍結実体と見なされる。

c . 実体の削除：独立実体又は組合せ実体が「現実状態」から「潜在状態」に切り替わる場合、それは削除されたと見なされる。

50

d . 実体の作成：独立実体又は組合せ実体が「潜在状態」から「現実状態」に切り替わる場合、それは作成されたと見なされる。

e . 実体の変更：独立実体の依存実体の値が変更される場合、その実体は変更された状態と見なされる。重要であると見なされるのに十分に有意な凍結状態での任意の変化は、実体が削除され、新しい実体が作成されることになることに留意されたい。例：棒が、プレスレットになるように曲げられる場合、「棒」は削除されたと見なされるべきであり、「プレスレット」は作成されたと見なされるべきである。

【0029】

[00182] 属性：属性もまた実体であるが、存在のために何らかの他の実体に依存する実体である。そのような依存実体は属性と明示的に呼ばれる。依存性は、別の独立実体の存在のおかげによるある実体の存在として定義される。例：ペンは空間及び時間に存在し得る。したがって、空間及び時間は属性又は依存実体と見なされる。なお、「空間単位」又は「時間単位」は、自己だけで実体と呼ばれる資格を有する。それらはより高レベルの実体に依存するため、本明細書では属性と呼ばれる。ペンが削除される場合、属性は自動的に削除される。

10

【0030】

[00183] 属性タイプ：3つの属性タイプがある：a) 必須属性：実体に関連して常に存在する属性である。例：空間、時間、アイデンティティ、数、論理、ユーザインターフェース(UI)等；b) 標準属性：ソリューション環境の性質に基づいて標準化された属性である；c) 一般属性：実体がソリューション環境内で置かれた状況に基づいて、ソリューション設計者が使用するために選択される属性である。

20

【0031】

[00184] 属性レベル：属性が存在することができるレベルの数に制限はない。1番目のレベルは「1次属性」と呼ばれ、2番目のレベルは「2次属性」と呼ばれ、3番目のレベルは「3次属性」と呼ばれる、等々。例：1番目のレベルにおいて、空間がインドとして定義される場合、各州は2次レベルに存在し、各都市は3次レベルに存在する。

【0032】

[00185] 現実実体：エージェントやエージェントがその現実を表現するために作成する人工的な表現実体に関係なく、自然(現実)に物理的に存在する実体。

【0033】

[00186] 具象実体：具象実体は、他の現実実体又は他の具象実体を効果的に表すことができるように、エージェントにより作成される。具象実体は、それらが表す現実実体又は具象実体から値を導出する。具象実体はまた、現実実体と全く同じように物理的でもある。すなわち、具象実体も時間及び空間に存在する。一般に、具象実体が表す実体と比較して、具象実体と結び付けられる分化のレベルははるかに低い。例：ペンは 10^{24} 個の原子からなり得、一方、「ペンという単語」を構成する原子はその 100 万分の一の原子しか有していなくてもよい。具象実体には、永続実体及び情報実体の2つのタイプがある。

30

【0034】

[00187] 知覚実体：知覚実体は、人間エージェントにより彼ら自身の脳で自然に所与のニューロン状態の形態で作成される。自然は、脳の進化の過程を通して、他のエージェントと意思疎通を図るために作り出した現実世界における実体、つまり現実実体又は情報実体を表すように脳を設計してきた。脳はこれらの具象実体を記憶、処理、検索する能力を有する。感覚はこれらの具象実体を入力するのに役立ち、身体は、コミュニケーションを通して他のエージェントにこれらを伝播させる能力を有する。

40

【0035】

[00188] 情報実体：情報実体とは、所有している具象実体をエコシステム中の他のエージェントに伝えるために、人間エージェントにより人工的に作成される実体である。

【0036】

[00189] 真理値：具象実体は、「現実実体」又は「他の具象実体」を表すとされているが、真理値は、自然における固有の不確実性又は人間エージェントの理解、動機、若し

50

くは意図を含む多くの要因に起因して変化し得る。例：「X」は「Y」の場所にあると表され得るが、それは正しい文章である可能性もあれば、又はそうではない可能性もある。正しい場合、その文章は「真」であると見なされ、正しくない場合、その文章は「偽」と見なされる。

【0037】

[00190] 情報内容 - 現実実体 v s 具象実体：理論上、現実世界の全ての実体は、例外なく、情報実体により表すことができるという単純な原理に従うことができる。これは、「現実実体」についての十分な知識が情報として表されることを前提としている。その知識を所与として、理論上、全ての実体は情報実体によって表すことができるのみならず、それらの情報内容は情報のビットの観点からも測定することもできる。問題は、その「現実実体」を表す「具象実体」に関して、情報内容が何であるかということである。一般に、具象実体の情報内容は極めて低い。「現実エージェント」の情報内容は 2^{70} ビットにも昇る情報であることができるが、そのエージェントの名前は、「現実エージェント」を有効に表すために数ビットの情報しか必要としない。「現実実体」に固有の全ての情報が「具象実体」によって表されるわけではなく、目下の目的のために現実実体を有効に表すのに十分な情報が使用される。最適化原理に基づいて、分化と統合とのバランスをとるエージェントによる情報の使用には多大な経済性がある。

10

【0038】

[00191] 物理的現実：全ての実体 - 現実実体、知覚実体、及び情報実体は、物理的世界に「物理的」に存在する。現実実体に関してその結論に至るのは直観的であるが、知覚実体及び情報実体であっても物理的現実 - 時間及び空間 - に存在することを明示的に留意されたい。知覚実体及び情報実体が具象実体であることからそれらの値を導出するということは、物理的でもあるそれらの特性を変えはしない。

20

【0039】

[00192] 変更ドライバ：変化単位の変更を引き起こす実体とその属性は、変更ドライバとして知られている。属性は実体でもあるが、独立実体に依存する。上述したように、独立実体を更に分化させる任意の数の属性レベルが存在することができる。各ドライバは、それ自体の一意で別個のアイデンティティ及び固有の情報を持つ。

【0040】

[00193] 変化単位 (CU)：変化単位は、自然言語の視点から意図言明により記述されるものである。あらゆる類いの価値も、制御される変化を通してのみ生じ、それは変化単位内部でのみ生じる。人間エージェントが望んでいる、又は意図しているのは、ローカル変化単位をまとめる (文章を組み合わせる段落を作る) ことによって、グローバル変化単位に到達することである。簡潔にするために、NSLはこれらの変化単位を意図言明 (SI: statements-of-intent) と同義として扱う。したがって、CU及びSIは、本文書では同義で使用される。ローカル意図言明は、自然言語に関して意図言明 (文章) として表現されるローカルの「変化単位」である。グローバル意図言明は、自然言語に関してグローバルな意図言明 (段落) として表現される「グローバル変化単位」である。

30

【0041】

[00194] 「情報技術」には、情報の観点から世界中の全ての実体を表す能力を有することに留意されたい。これは、変化単位で生じたものが意図言明において完全に捕捉される方法である。

40

【0042】

[00195] 変化単位 (CU) のサイズ：CUのサイズは、意図されるCUに達するための「CUの数の増大」又は「CU内の独立実体 (変数) の数の増大」の適切なバランスをとることにより最適化される。例：理論上、CUの数は無限に増大させることができる。家から空港までのある人の移動は、CUであることができる。空港に辿り着くまでのある交差点から別の交差点までの各移動はCUであることができる。車のホイールの各回転はCUであることができる、等々。しかし、それは複雑性を増大させるとともに、複雑性を扱う本発明の能力を増大させる。それどころか、CUのサイズが増大する場合、CU内の

50

変数の増大に繋がり、CU内の変数状態を劇的に増大させ、ここでも大きな複雑性に繋がる。インドにある家から米国内の特定の遠隔場所に行くことは、変化単位であることができる。しかし、その変更は多すぎる実体(変数)を消費し、手に負えなくしてしまう。このCUに関わる実体の数はあまりに多すぎて、実体の組合せはシステムをかなり複雑にすることになる(あまりに多すぎる車、航空機、ホテル、パスポート、ビザ等の実体、及びそれらの全ての属性が存在する)。何れの場合の複雑性も考慮して、設計者は最適化原理を適用して、2つの極端な場合のバランスをとる。CUは、「意図言明」により表され、ソリューション設計の目的では意図言明と同義と見なされる。

【0043】

[00196] イベント：全てのイベントは、何れかのローカル意図言明に届き、全てのイベントは、個々の実体がある状態から別の状態に、又はその逆の可能性のある状態に変わることに伴うものである。LSIへの又はLSIからの個々の実体の到着又は出発によるイベントの発生により、組合せセット全体の状態が変化する。LSIに6つの変実体がある場合、組合せ実体が存在する可能性がある64の異なる状態が存在し得る。個々の実体レベルでのバイナリ状態の変化は、LSIにおける64の異なる状態の何れか1つに繋がる可能性がある。他の63の状態は非トリガー実体状態組合せであり得るが、64番目の状態は、他のLSIの状態又はそれ自体に影響するトリガー状態になる。

10

【0044】

[00197] バイナリ状態：NSLソリューション設計では、全ての実体状態は、バイナリ状態でのみ存在するものとして表現される。すなわち、潜在状態又は現実状態の何れかで存在する。あらゆる状態は離散し、中間状態はない。物事のこの見方では、各単語はバイナリ状態で動作し、文章及び段落であってもバイナリ状態で存在する。エージェントは、実体を見る視座を変更し続けるだけである。ズームイン又はズームアウトすると、視座は変化するが、各視座はバイナリ状態である。NSLソリューション設計におけるバイナリ状態の選択は離散状態vs連続状態の選択であり、デジタルvsアナログの選択に似ている。イベントが発生すると、中間が除外された状態での状態遷移がある。理論上、これらのバイナリ状態は、「真又は偽」である値を割り当てることにより表すこともできる。

20

【0045】

[00198] 変動性：実体の大半がバイナリ状態で存在し、変動性がそれらの状態間のみであることが一般に好ましい。変動性は2つの理由で高くなる可能性がある：第1に、より高い視座では、全てのサブセット実体値が一緒になって実体状態組合せの数、ひいては変動性を増大させることができる。例：10個のサブセット実体がある場合、それらは 2^{10} 、すなわち1024の変動性を生じさせる。第2に、任意の数の状態は、多くのバイナリ状態の組合せによって表現することができるが、簡潔にするために、システムは、必要な場合、情報として分化ラダー(differentiation ladder)の最後の段と一緒に記述される多数の状態を提供する。例：人が特定の都市にいる間、その人は100万もの空間座標点の何れか1つにいることができる。

30

【0046】

[00199] 不変：状態が常に同じままである実体は、不変と見なされる。例：関連期間中、建物の存在が前提として解釈される場合、建物の存在は前提として見なされる。

40

【0047】

[00200] 入力：本質的にLSIに関してのイベントである。これらは、人間エージェントにより又は機械エージェントにより促進されるようにシステムに供給されるイベントである。例：時刻を入力する人間エージェント又は機械エージェントのフィードに基づいてシステムにより直接捕捉される。入力は、既存の実体を潜在状態又は現実状態に切り替えるという形態であってもよいし、又は代替的には、実体適格性判断プロセスを通して新しい実体をエコシステムに導入し、若しくは既存の実体をエコシステムから削除するという形態であってもよい。入力の導入は一般に、エージェントが所有する「決定権」に基づいてエージェントにより行われる。

【0048】

50

[00201] 出力：本質的に、1つ又は複数の他のLSI又はそれ自体におけるLSIにより生じたイベントである。

【0049】

[00202] 経過時間：説明されるように、トリガー状態が意図言明において獲得されると、それ自体の意図言明を含む1つ又は複数の意図言明における1つ又は複数の変化を引き起こす。全ての変化には、「経過時間」と呼ばれる時間が掛かる。変化が人間エージェントにより駆動されるか、それとも機械エージェントにより駆動されるかに関係なく、経過時間は常に関与する。そのような変化は数分の一秒で生じることができるともあれば、時間長が数時間又は数日と長いこともある。必要とされる変化を生成する相互作用に関わる全ての実体は、経過時間中、占有され、前の行動の完了後でのみ、任意の他のトリガーへの関与に利用可能になる。

10

【0050】

[00203] ソリューションの全ライフサイクル：あらゆるローカル及びグローバル意図言明に関してフラクタルのような自己相似性がある。この対称性及びパターンの再帰、活動を生み出すあらゆる値及び指示された変化に関して現れる。システムは、環境を「感知し(sense)」、環境から「選択(select)」し、それに対して「行動(act)」する能力を提供することを目的としている。これは「SSA」サイクルとも呼ばれる。NSLは、あらゆるソリューションの背後にあるこの全ライフサイクルの存在を認識し、それに効率的に対処する。システムは、あらゆる変化がまず「感知」され(人間の感覚が行うのと同じ)、可能なことから1つ又は複数の実体を「選択」し(人間のマインドが行うのと同じ)、1つ又は複数の選択された実体に対して「行動」する(人間の身体が行うことと同じ)ことに関して全ての可能性を提供する。ソリューション要件のこの全ライフサイクルの本質は、続けて説明する「エージェント機能」の存在を生じさせる。

20

【0051】

[00204] 自然言語：自然言語が働く仕組みは、各単語が、同じ言語を話す人々のみが認識するコードのようなものである。文章は、変化単位に対処するように文法の縛り内で単語を組み合わせるにより構築される。文章を更に続けることにより、自然言語は、複雑な現実を伝える能力を生み出す。自然言語は、現実と結ばれたコードの形態であることに気付かれない。コードは、単語及び文章を特定の方法で組み合わせることにより機能する。あるコード実体が別のコード実体と組み合わせられる都度、分化したコード実体組合せが生成される。各組合せは、可能性の範囲からの選択の結果である。そのような組合せの連鎖は、分化原理により駆動される一意の組合せを生成する。自然言語は、プログラミング言語と全く同じように、分化の原理に頼って目的を達成する。プログラミングコードを自然言語で置換することは、「自然言語構造」の背後にある十分にテストされた基本原理の使用を通してNSLがもたらすパラダイムシフトである。

30

【0052】

[00205] コードvs自然言語：コードは、具象実体を解読し理解する能力を有するエージェントによってのみ理解される具象実体である。プログラミングコードは、機能をコンピュータに伝えることができるモノである。コードの意味及びその適用はプログラマの範囲内である。プログラマはこのコードを使用して、適用論理をコンピュータに伝える。自然言語もコードの形態である。それが、英語話者がロシア語話者を理解せず、その逆も同様であることの理由である。NSLが行うことは、もはや別個のコードライタを有する必要がなくなるように、適用論理をコンピュータに伝える能力を自然言語に近づけることである。ユーザは、NSLの使用を通してユーザインターフェースレベルで直接、自然言語に対処することができる。

40

【0053】

[00206] 自然言語ソリューション(NSL: Natural Solution Language)：これは、自然言語をわずかに変更したものであり、コンピュータ実施方法で、意図言明のみを捕捉し、すべての記述ステートメントを意図言明に従属させる形で作り直される。これらの意図言明には2つの状態がある：

50

【 0 0 5 4 】

【00207】 静的意図言明：意図のみを表現し、意図を現実に変形する能力を有さない。静的実体状態は、他の状態の変化をトリガーすることができる特性を有さない実体状態である。システムに6つの変数がある（独立実体及びそれらの属性）場合、潜在的に64個の異なる状態で存在することができる。しかし、全ての変数が「現実」状態で存在する場合、64番目の状態のみが、変化をトリガーすることができる。他の全ての実体状態は「静的実体」状態と呼ばれる。これに関して留意すべきポイントは、意図言明（S I）が、「変化単位」に参加している所望の変化の性質を記述するもう1つの別の独立実体であることである。S Iの存在は、続く任意の行動について、エージェントの意図によりバックアップされる必要があることに基づく。

10

【 0 0 5 5 】

【00208】 動的意図言明：独立実体レベルでイベントによる影響を受けて、意図言明の背後にある実体は、トリガー実体状態組合せと呼ばれる特定の所望の状態の獲得を集合的にトリガーする、基礎となる変形を引き起こす。動的実体状態とは、それら自体を含め、1つ又は複数の実体状態を更に変化させる能力を有する実体状態である。先の例では、64番目の状態が「動的実体」状態である。換言すれば、静的意図言明が動的になり、意図言明を履行するには、トリガーCESによって駆動される必要がある。

【 0 0 5 6 】

【00209】 自然言語 vs NSL：自然言語では、イベントは文字毎及び単語毎に順次生じる。しかしながら、NSLでは、潜在実体は、最初に単語毎及び文章毎に段落レベルまで設計者により構築される。イベントは、段落全体の単語レベル又は句レベルで生じるが、必ずしも連続して生じるわけではない。しかしながら、イベントは、NSL論理若しくは文法又はコンピュータにより実施される規定の方法の縛り内で生じる。自然言語は、技術的なプログラミング言語の代わりに自然言語を使用して、コンピュータに向けてアプリケーション論理を成文化する。ユーザは、いかなる技能にも頼らずに、はるかに容易且つ効率的にアプリケーション論理を構築又は使用することができるようになる。このプロセスでは、複雑なプログラミングコードの形態の抽象化された層はなくなる。

20

【 0 0 5 7 】

【00210】 意味 vs 価値：自然言語では、情報分化は意味の伝達を生じさせる。一連の単語は、別の一連の単語と比較して1つの意味を伝達する。NSLコンテキストにおける値は、「ソリューション」又は「意図状態の履行」（グローバル意図言明）と同義で使用される。自然言語において連なった分化単語が意味を伝えるのと全く同じように、連なった分化実体は値又は「意図言明の履行」に繋がる。

30

【 0 0 5 8 】

【00211】 値の定量化：全ての変化は独立実体レベルで生じ、実体 - 状態組合せを生じさせる。これらの変化はイベントにより生じる。イベントは、潜在状態から現実状態への間の切り替え及びこの逆の切り替えである。あらゆる意図言明では、変化トリガーを発生させるためには、その全ての独立実体が現実状態を獲得する必要がある。これにより、ローカル意図言明を履行するためには、現実状態に変わる独立属性及び関連する属性の数が明確に確立される。拡大解釈すると、各グローバル意図言明が所与の数の定量化可能な累積実体現実状態を必要とすることも証明した。情報が「ビット」のカウントによって定量化することができるのと全く同じように、値もバイナリ実体状態のカウントを通して定量化することができる。

40

【 0 0 5 9 】

【00212】 価値の相対性：あるエージェントにとって有価値なものは、別のエージェントにとっては有価値ではないことがある。有価値なものは、その価値を探し求めている1つ又は複数のエージェントに対するものである。NSLでは、各段落は、グローバル意図言明を有するエージェントに対するグローバル価値単位を表す。本明細書ではこれを価値の消費と呼んでいる。価値の消費者の観点から、ローカル意図言明は、価値創造者の役割を果たす様々なエージェントを有することができ、それらの変化単位は価値のイネーブラ

50

である。ここで、仮にローカル意図言明に関わるエージェントに観点を切り替えた場合、そのローカル意図言明はグローバル意図言明に変わる。ローカル意図言明は、関わるエージェントの消費ニーズに応える段落中の意図言明の最後の履行になる。相対価値を考慮すると、分化経路が互いに交差して、異なるエージェントにサービスする網状構造をもたらす。

【 0 0 6 0 】

【00213】 自然言語と NSL との間の共通性：両方とも分化原理により駆動され、同義実体を有する： a) 名詞： NSL における独立実体に類似する、 b) 形容詞： NSL における、独立実体を修飾する属性に類似する、 c) 動詞： NSL における意図言明において反映される変化の記述子である、 d) 副詞： NSL における意図言明に含まれる動詞を修飾する属性である、 e) 文章： NSL における、必要な命令された変化をトリガーすることができる全ての潜在実体で構成される意図言明に類似する、及び f) 段落： NSL ソリューション設計の観点から、段落は、制御される変化の連なりを通して意図言明の履行に繋がることのできる多くの意図言明からなるものである。段落は全体として、グローバル意図言明として見られる。

10

【 0 0 6 1 】

【00214】 論理の段落： NSL は、全ての実体の幾つかの「必須属性」に頼る。空間及び時間と全く同じように、 NSL 実体の必須属性の 1 つは、実体の相互作用を定義する「論理」の仕様である。各実体はその「論理対応物 (logic counterpart)」を指定すべきである。この論理対応物は、各グローバル意図言明に対する段落を定義する論理への参加者である。これにより、グローバル意図言明がエージェントに対して生まれる都度、エージェントは、その意図に対する論理の段落を探し出す。次に、エージェントは、「論理の段落」を使用して、それ自体のカスタマイズされた潜在段落を作成する。同様に、この新しい段落に参加している各実体は、論理の段落において対応する実体の役割を果たす。

20

【 0 0 6 2 】

【00215】 NSL ライブラリ： NSL は、全ての潜在実体及びそれらの関係を捕捉するために、全てのソリューションを縮小する。 NSL は、最も単純なレベルから最も複雑な分化レベルにおける潜在実体のライブラリを作成する。これらのライブラリは、様々なソリューション環境を網羅する可能性を有する。その後、サーチ及びドラッグアンドドロップ技法を使用することにより、最小限の追加労力で潜在実体を再使用することができる。

30

【 0 0 6 3 】

【00216】 再利用可能なコンポーネント：種々の視座における既存の実体を使用することによりソリューション論理を構築する際の冗長性を最小限に抑える能力が、コンポーネントの再利用である。これらは、 NSL が構築するライブラリで提供される再利用可能な実体コンポーネントである。

【 0 0 6 4 】

【00217】 分化原理：人間存在のレベルにおいて、全ての実体は状態組合せでのみ存在し、高度に分化している。 1 つの塵であっても 1 兆もの原子を有し、それ自体の構造を有する。他の同じままのものは、数々の可能性から選択された独立実体が既存の実体状態組合せと組み合わせられる都度、生成された実体状態組合せはさらに分化する。これが分化の原理である。独立実体価値が無視されて、未分化、一般化、又は統合が生じる場合、この逆が生じる。実体の追加は分化を生じさせ、実体価値の減算は統合を生じさせる。原理は、「算法」での分化及び統合にいくらか類似する。例：実体状態組合せ「 A - B - C 」があり、独立実体「 D 」をそれに加える場合、生成される実体状態組合せは「 A - B - C - D 」であり、これは、実体「 D 」の追加から生じた A - B - C からの分化形態である。「 D 」の値が無視され、実体状態組合せがそのままである場合、未分化が生じ、「 A - B - C 」及び「 A - B - C - D 」はここで、同じ値を有し、「 2 つの A - B - C 」としてカウントされる。

40

【 0 0 6 5 】

【00218】 人間エージェントに資する、人間エージェントに有害な、又はエージェント

50

が不可知の分化又は統合（一般化）があることができる。ソリューションアーキテクチャにおいては、情報実体は現実実体又は他の情報実体を表す。ソリューション設計中の全ての実体は、独立であれ組合せであれ関係なく、まるで潜在又は現実の離散（凍結）バイナリ状態で存在するかのように扱われる。そして、新しい実体状態組合せが作成される都度、これらのバイナリ実体は他のバイナリ実体と組合せられる。設計は、これらの新しい結合された状態のそれぞれもバイナリ状態で存在するものとして扱う。全ての実体 - 状態組合せは、何れかの変化単位のエコシステム内に存在する。これは、NSLのローカル意図言明と同じである。

【0066】

[00219] エージェントは、「分化」及び「統合」（一般化）の両方を得意とする。分化すると、他のことが同じままの場合、より多くの情報ビットを消費する。一般化の場合、より少数の情報ビットを消費する。これらの2つの良いバランスが生き残りにとって重要である。脳は限られたキャパシティのみを有する。

10

【0067】

[00220] 値を無視することにより、一般化又は統合に向かう能力を有する。全ての実体が存在について認識されるが、全ての値が無視される場合、一方の端に実体カウントのみが存在するという最も統合された状況となる。しかし他方、全ての実体を分化させることにより、全ての実体が最も分化した状態である分化ツリーのもう一方の端に移る。

【0068】

[00221] 興味深いことに、ある「変化単位」における実体 - 状態組合せが、それらの「変化単位」内の実体 - 状態組合せを変えることによって他の「変化単位」を変化させる場合、新しい状態は古い組合せ状態と組み合わせられて、変化のカスケードを作成する。自然言語の観点から、これらは、文章をつなぎ合わせて段落を形成するのと同じである。NSLの観点では、各変化単位は、ローカル意図言明が組み合わせられてグローバル意図言明を形成するのと同じである。

20

【0069】

[00222] 視座：各基本実体は、個々のレベルにおいてバイナリ状態で存在する。実体はまた組み合わせられて、組合せ実体も形成する。視座とは、実体を見得る相対位置を指す。仮に実体を全体的により高い視座 - その全てのサブセットからなる分化ラダーのより高い段 - から見た場合、接続される実体のカウントはかなり大きくなる。逆に、仮に、より低い視座 - 分化ラダーのより低い段 - にある実体を見た場合、接続される実体のカウントははるかに小さくなる。例：分化の方向において、より高い視座「A」を考える。「A」は分化したサブセット「A - B」を有することができる。それが「C」と組合せられる場合、第2のレベルのより分化したサブセット「A - B - C」を有する。「A - B」はサブセット「A - B - C」のみを包含する。「A - B」は、「A」と比較して分化方向においてより少数の接続実体を有する。言い換えれば、視座が高い実体ほど、視座が低い実体と比べて多くの情報を有する。

30

【0070】

[00223] 方向性：これは、実体価値の追加又は実体価値の削除の何れかに基づく分化ツリーの上又は下への移動についてである。新しい実体価値を追加すると、分化の正方向に移動する。価値を削除することにより、分化の負方向 - すなわち、「未分化」、又は「一般化」、又は「統合」の方向 - に移動する。

40

【0071】

[00224] 実体状態組合せ（CES：Combinatorial-Entity-States）：独立実体（一緒に属性を引きずる）が他の独立実体と組み合わせられる場合、それらは分化し、それら自体の新興属性を有する組合せ状態が生成される。CESは、変化単位 - 文章と同等である - をそれぞれ表すローカル意図言明（LSI）に収容される。より大きな所望の変化に繋がるそのようなLSIの集まりはグローバル意図言明（GSI）と呼ばれ、段落と同等である。LSIにおけるCESのサイズは、参加している独立実体の数に比例する。各独立実体は「潜在」又は「現実」の状態の何れかで存在することができるため、各LSIは

50

2^n 個の状態を有し、ここで、「 n 」は独立実体の数である。基本的に、各ローカル意図言明には2種類の実体-状態組合せがある。

【0072】

[00225] 非トリガーCES：他のローカル意図言明又は属する同じLSI内で実体状態組合せを変えない実体組合せである。例：4つの独立実体がある場合、バイナリ変数として、 2^4 個の実体状態-すなわち、16個のCES-を生じさせる。これらの状態のうち、15個の状態は、いかなる変化もトリガーしないため、非トリガーCESである。

【0073】

[00226] トリガーCES：これらの実体状態組合せは、1つ若しくは複数の他のLSIにおいて又は同じLSI内で変化をトリガーする。トリガーCESは、全ての独立実体及びそれらの属性が現実状態であるCESである。1つのLSI内のCESが他のLSI内のCESに影響する場合、それらの文章は接続される。LSIは、GSIの現実化への関わりに基づいて一緒にグループ化されて、段落を作る。例：先の例を続けると、16番目の状態は、その状態で4つ全ての独立実体が現実状態で存在するため、トリガー状態である。

10

【0074】

[00227] 独立実体が一緒に接続されると、分化した実体状態組合せが生じることを既に述べた。ローカル意図言明も一緒に接続されると、実体状態組合せは、段落レベルにおける分化原理の適用まで更に拡張される。理論上、段落を組合せて、章又は更に高レベルの抽象にすることができる。

20

【0075】

[00228] APIvsプログラム：アプリケーションプログラミングインターフェースは、まず第1に、プログラミングコードと同様の方法を使用して他のプログラミングソリューションへのアクセスを助ける。NSLは、プログラミング言語で行うのと同じようにしてAPI論理をコンピュータに通信する能力を有し、同じ機能を効率的に実行することができる。

【0076】

[00229] 共有実体：共有実体とは、多くのローカル又はグローバル意図言明にわたり共通する実体である。所与の意図言明でトリガー状態の一部である多くの独立実体がある。意図言明がトリガー状態である場合、そのトリガーに関連する経過時間中、その実体は、いかなる他の意図言明に対する参加にも利用可能ではない。しかし、トリガーされた変化が完了すると、参加実体は、関連する全ての意図言明にわたる共有実体として再び利用できるようになる。

30

【0077】

[00230] 変化の経路：変化の経路は、グローバル意図言明の履行に繋がるローカル意図言明内のトリガー実体-状態組合せ変化のカスケードにより作成される。NSL論理は、変化のこれらの経路を定める。プログラミング論理では、ある状態から別の状態へのそのような遷移は、異なる状況下で異なる名で通っている原理を適用することにより達成される。そのような少数の例は以下である：a) 制約：多くの可能性間で行われる選択に課される制限がある。NSL論理では、分化原理が適用されて、同じ結果を達成する。これらは単に、選択が行われる際、任意の変化の入口における「可能性のセットの低減」である。b) ルール：殆どの場合、状況の影響を受けやすいシステムに課される一連の制約である。c) アルゴリズム：「ルール」と同じであるが、多くの場合、「情報システム」の状況で使用される。

40

【0078】

[00231] 情報へのプロセスの変換：NSLが行う最も劇的なことの1つは、プロセスも情報に変換することである。NSLは厳密に分化原理に依拠し、関連する変化を実体-状態組合せとして一緒にストリング化することにより、プロセス及び関数を情報に変換する。あるCESが原因となるトリガーを通して別のCESに影響を与えると、両CESは統一され、これは「拡張CES」と呼ばれる。全てのプロセス及びアプリケーション論理

50

を情報領域に持ち込むのは、これらの拡張CESである。情報への変換が行われると、それらの実体は、任意の他の情報と同様にサーチ原理の対象となる。情報が定量化可能であるのと全く同じように、ソリューションも定量化可能になる。

【0079】

[00232] ITにおける抽象化レベル：情報技術は複数の抽象化レベルにより駆動される。この抽象化ラダーの最下端部には、0及び1のバイナリ状態を示す電磁状態がある。他端部には、ユーザが経験するアプリケーション論理及びソリューションがある。NSLは、プログラミングコードレベルを自然言語のような特徴で効率的に置換する。

【0080】

[00233] エージェント：エージェントも実体である。エージェントは、ソリューションの作成者及びソリューションの消費者の両方である。エージェントは、目的により駆動されるため、「自然」から分化する。換言すれば、エージェントは、好都合な変化を探し求め、不都合な変化を回避する。あらゆるソリューションは、制御される変化に対処するため、全ての変化単位がエージェントにより-人間エージェントであれ機械エージェントであれ関係なく-影響を受けることが前提とされる。全ての変化はエネルギーを必要とするため、エージェントは固有のエネルギーを使用するか、又は1つ若しくは複数の組合せ実体からエネルギーを借り、以下の所定の経路により又は自由意志の適用を通して方向性を変化に提供する。

10

a. 人間エージェント：人間エージェントは、「ステークホルダ」とも呼ばれ、ソリューション環境により課される要件に応じて複数の役割を演じる。幾つかの変化単位は必然的に、人間エージェントにより駆動される。例：幾つかの「有形資産」の物理的搬送は、人間エージェントの関与を必要とする。

20

i. 価値創造者：あらゆる意図言明（変化単位）は何れかのエージェントにより駆動される。それらの意図言明又は「意図言明と接続されたそれらのエージェント」は、「ローカル意図言明」に関連するが、「グローバル意図言明」に関連しない場合、価値創造者と見なされる。全てのローカル意図言明及びグローバル意図言明が、同じエージェントで駆動されている場合、「創造者」機能及び「消費者」機能の両方に、関与する同じエージェントを有する。

ii. 価値消費者：グローバル意図言明（意図言明を履行する言明）と接続されたエージェントは、価値消費者と見なされる。これらのGSIは、段落の終わりの意図言明に列席する。

30

iii. チーム：LSIの変化が人間エージェントにより集合的に駆動される場合、2人以上の人間エージェントが存在する。そのようなチーム努力では、方向性の責任はリーダーにより又はチームメンバー間で指定される決定権に従ってとられる。

iv. 所有権：任意の実体の所有権は、エージェントが制御される変化に参加する際、実体に直接又は間接的に、物理的に影響するエージェントの能力に基づく。具象実体も同等に物理的性質であるため、物理的影響は具象実体も含む。影響する能力はまた、エージェントに割り当てられる決定権にも依存する。すなわち、エージェントは影響力を持つ権利も有するべきである。情報権自体は、エージェントが影響を与える権利を持たない場合であっても、情報を保有する権利をエージェントに付与する。

40

v. エージェント機能：意図言明内で行われるエージェント機能は3層の意図言明に分割することができる。(a)第1の層-物理機能：物理機能は、意図言明のバックボーンを形成する実体-状態組合せ(CES)を生成する独立実体の参加に関連する。物理機能は、ソリューション促進という主機能を果たし、一方、他の2つのカテゴリは直接又は間接的に、物理機能をサポートする。(b)第2の層-情報機能：情報機能は、実体-状態組合せに接続される実体に関連し、情報を提供する機能のみを果たし、物理機能に関わらない。拡大解釈すれば、これらの情報機能は意図言明及びそれらを駆動するエージェントに接続される。情報機能は、エージェントを情報に通じた状態に保ち、動的ソリューション再設計並びに分析、機械学習、及び人工知能のような他の付加価値機能において役割を演じる。(c)第3の層-マインド機能：マインド機能は、コンピュータ実施NSLの

50

観点から、現実世界における人間のマインドの機能を模倣する。これらの機能は、実体状態を「予測」し、所望の変形をもたらすプロセスにおいて物理機能をガイドする。予測は一般に、将来に関する「時間態様」に適用される。しかし、予測は、不確実性がある全ての状況に適用することができる。理論上、不確実性は、過去又は現在のことにも関することができる。例：昨日何が起こったか又は現在、他の部屋で何が起きているかについて完全に知ることができないことがあるが、予測しようとすることはできる。「推定」又は「予測」の改訂は、あらゆるイベントの瞬間に生じることができる。そのような予測は、現在における物理機能と関係がある場合、影響を及ぼす実体として物理機能にフィードバックされる。情報機能等の同様の役割を演じることの他に、マインド機能は事前計画及び最適化の件にも役立つ。

10

v i . 情報権：特定の独立実体又は組合せ実体及びそれに接続された意図言明に関する情報についての人間エージェントの権利である。

v i i . 決定権：独立実体又は組合せ実体及びそれに接続された意図言明の潜在状態又は現実状態を変える人間エージェントの権利である。

b . 機械エージェント：入力にตอบสนองして適切な出力が生成されるように、変化単位が機械エージェントにより駆動される「コンピュータ」と同義である - 人間エージェント又は他の機械エージェントにより設計される - 。機械エージェントは本質的に、入力を消費し、入力の中から制御された変化を生成（それらの入力を処理）し、出力を生成する人間エージェントの能力を模倣する。ある意味では、人間エージェントは、目的により駆動されるという性質を機械エージェントに付与する。

20

c . マザーネイチャー：マザーネイチャーは、変形に影響を与えることが可能な第3種のエージェントである。しかしながら、マザーネイチャーは目的を保有しないため、マザーネイチャーにより生成されるそのような変形結果は、人間エージェントによるソリューション設計において黙示実体として解釈される。

【0081】

[00234] ソリューション設計者の指示に従ったエージェント行動：ソリューション設計者はソリューションの建築者である。ソリューションは、1つ又は2つ以上の変化単位で構成される。各変化単位はエージェントにより制御される。このソリューションは潜在としてレイアウトされる。例えば、ディレクターは台本の語り手になり、俳優が、ディレクターが望むように台本に忠実であることを確実にする。この場合、ディレクターはソリューション設計者であり、俳優はエージェントであり、個人又はチームであることができる。

30

【0082】

[00235] 「エージェント」として扱われるべきチーム：エージェントは、個人又は複数のメンバで構成されるチームであることができる。チームとは、それ自体のアイデンティティを有するエージェントの組合せである。NSLでは、ソリューション設計者の選択に基づいてチームをエージェントとして扱うことができる。例：「取締役会」による承認が関わるソリューションでは、取締役会は、チームであるが、エージェントとして扱われる。

【0083】

40

[00236] 協同的エージェント及び対立的エージェント：NSLでは、エージェントは協同及び対立として分類することができ、共通の目標/願いの達成に向かって一緒に働くエージェントは協同的エージェントと呼ばれ、一方、互いに競合する（「ゲーム環境」と同様に）エージェントは対立的エージェントと呼ばれる。

【0084】

[00237] 判決システム：一般に、1つの「段落」は1つのグローバル意図言明（GSI）のみを収容する。GSIが多いほど、より多くの段落が存在する必要がある。共有ローカル意図言明を有することから生じる関連段落があり得る。これは、一般のプロセス指向手法における「シナリオ」に類似する。同じグローバル意図言明が、含まれるローカル意図言明のバリエーションを使用して、異なる段落にサービスを提供できる場合があり得

50

る。これは、異なるサブプロセスを通して同じ結果を達成することに類似する。NSLは、分化原理を使用してこの問題に対処する。NSLは、追加の分化「実体をシステムに」注入して（制約と同様に）、グローバル意図言明にサービスを提供するために残っている1つの段落のみに可能性を微調整する。グローバル意図言明にサービスを提供する段落が残っていない場合、システムは幾つかの実体をなくす可能性を探索することができ、それにより、未分化を作成し、潜在的に、グローバル意図言明を発生させる少なくとも1つの段落を生成する。これは、幾つかの制約をなくすことと同じである。

【0085】

[00238] 実体三つ組み：エージェント、変化単位、及び一般実体（独立実体又は依存実体の何れか一方）は同時に存在し、意図された変化を生じさせる。一方は他方なくしてはあり得ない。宇宙には膨大な実体があるが、重要な実体は文脈上一緒になっている。変化は、エージェントなしでは起こり得ない。変化は常にエージェントにより駆動され、エージェントは意図を有し、目的により駆動される。任意の変化を生じさせるには、実体間の関係が生じなければならず、変化は物理的に生じるべきである。例：文字を書くためには、以下の実体が集まる必要がある：（a）エージェント（文字を書く人）、（b）一般実体（ペン、紙、及びテーブル）、（c）意図（文字を書く）。この三つ組みは、相互作用を通して指示された変化を起こすことにより意図／望みを遂行する。

10

【0086】

[00239] 最近傍：所望の状態になるためには、多くのローカル実体状態組合せを通して遷移する必要がある。これらの変形状態は全て、指示された変化を通して獲得される。これらの状態は全て、最近傍原理により動作し、最近傍は実体三つ組み、例えば、エージェント、変化単位、及び一般実体に等しく適用される。願いを遂げるためには、接続された変化単位は順序の影響を受ける。これは、変化単位間の接続が、順序が重要な成分になる相互作用又は原因結果原理を通して確立されることの単純な理由である。原因結果が生じる場合、変化はある順序で生じ、したがって、原因は常に t_1 にあり、結果は常に t_2 にあることになる。例：文字は、書かれない限り、掲示することはできない。次の変化単位は、選択された任意の変化単位の最近傍になる、等々。これは、エージェント及び一般実体にも同様に当てはまる。

20

【0087】

[00240] ステークホルダエンゲージメントセンタ（SEC）：これはユーザインターフェースと同義である。SECは、任意のエージェントに関して潜在性を保持する全ての実体を認識し、それらをユーザインターフェースレベルで構造化して提示する能力を有する。システムは、全ての変化単位がある何れかのエージェントにより駆動される - m明確に所有権を確立する - ことに基づいて、関連実体を認識する。全ての実体が明確に指定された決定権及び情報権を有するという追加の事実により、エージェント間での実体の分散とナビゲーションが容易になる。

30

【0088】

[00241] 測定フレームワーク：測定は客観的な現実の言明である。同じことが起こるためには、何かを最適な明確レベルで提示する必要がある。通常、これは、提供される情報には、十分な粒度レベル又は十分な分化がある必要があることを意味する。例：「何かを送られた」と言うことと、「何か30分以内に送られた」と言うことは別のことである。そのレベルの詳細を求めている場合、2番目の言明は測定になる。測定フレームワークにおける測定の別の態様は、追加の分化がいいか悪いかを判断することである。これらの価値判断は、「規範」と呼ばれるものを通して導入される。したがって、「何かを30分以内に送る」ことが良いと考えられる場合、それは規範になる。一般に、緑色をいいと見なされることに割り当て得、赤色を悪いと見なされることに割り当て得る。

40

【0089】

[00242] 分散ミッションコントロールセンタ（MCC）：MCCは、人間エージェントが義務を実行又はニーズを履行する通常の過程で重要な実体を一緒にまとめる。重要なこれらの実体は、エージェントに関する実体（それは情報権又は決定権の何れか一方を有

50

する実体)の中からまとめられる。分散MCCは、まず、特定の1つ又は複数のエージェントに関する実体を自動的に認識し、次に、エージェントの機能に対して重要な実体をピックアップするシステムの能力に関するものである。

【0090】

[00243] 動的ミッションコントロールセンタ：動的MCCは、特定の「時間」、「イベント」、及び「エージェント」のクエリに関してカスタマイズされた関連エージェントに重要な実体が提示されるという点でのみ、分散MCCと異なる。分化/イベントは各実体レベルで生じるため、各文字を入力して、情報サーチを行うのと全く同じように、エージェントは、単語/数字(実体)を入力して、任意の他のクエリに応答させることができる。

10

【0091】

[00244] ユーザインターフェース(UI)：実体は、データベースにおいて異なる抽象化レベルで又はユーザインターフェースレベルで存在することができる。人間エージェントが「情報権」又は「決定権」を行使するのはユーザインターフェースのレベルにおいてである。「ユーザ」がソリューションの作成又は使用を担当するためしたがって、全ての物事はシステム内ではなくシステム外で生じるため、UIはNSLにおいてかなり大きな役割を演じる。NSLは、UIを「必須属性」にすることによりUIレベルで実体の挙動を駆動する。これらの属性は、実体がUIレベルでどのように現れるか、そのアドレス(どの画面及び画面上のどの場所か)、及び情報のナビゲート又は入力に対して敏感であるか否かを指定する。

20

【0092】

[00245] テキストベースユーザインターフェース(TBUI)：NSLパラダイムでは、全ての実体はより直観的な自然言語表現層において対処され、したがって、NSLはテキストベースユーザインターフェースを使用する。自然言語は、数千年にわたり存在してきており、高度に直観的である。自然言語は、一般から細部に流れる分化構造体を通して分化を捕捉し、高度に構造化され、自然に進歩しているように見せる。TBUIは、自然言語(テキスト)設計の原理に基づいているので、従来のグラフィカルユーザインターフェース(GUI)よりも構造化されている。TBUIは、電子書籍と同様に従来の本設計の原理を科学技術の革新と合成し、画像及び動画をテキスト表現に追加する追加機能を有する。TBUIは、ユーザのナビゲーション経験をよりよくし、各視座における俯瞰図をユーザに提供する。これとは対照的に、グラフィカルユーザインターフェース(GUI)は、部分を一般的且つランダムに分散させた状態で情報を提示し、俯瞰統合図を提供しない。

30

【0093】

[00246] NSL技術フレームワーク：NSLがもたらす革新的技法効果は、独自の手法と技術フレームワークとの組合せに起因する。NSLは、「ユーザ」がソリューションの主駆動者である広範囲のアプリケーション要件に応える標準化技術フレームワークを乗り越える。本明細書に記載される全ての革新的方法をカプセル化した技術フレームワークは、オペレーティングシステムの上に置かれ、コンピュータに伝達されるあらゆる種類のアプリケーション論理に対応する。事実上、ユーザはこの基礎となる技術フレームワークにとらわれず、自然言語のようにNSLを使用する権限を与えられる。まとめると、NSL技術フレームワークは、分化原理に基づいてNSLの背後にある原理及び方法に命を与える、既存のオペレーティングシステムの上の薄い追加の層である。加えて、NSL技術フレームワークはまた、人間エージェント以外の機能の大半の自動化にも役立つ。しかし、随時拡張するために、NSL技術フレームワークは一定のままである。オペレーティングシステムと全く同じように、いかなるコードも使用せずに、自然言語構造体を通してNSL論理をコンピュータに伝達する。

40

【0094】

[00247] リエンジニアリング：NSLは、アプリケーション論理をコンピュータに自然言語のように伝達する。そうすることで、NSLは、システム内の動作であるプログラ

50

ミングコードを書く必要性をなくす。ユーザは、プログラミングコードの場合、ソリューションの構成の性質又はそのアクションを認識していない。NSLは、ユーザフレンドリであり、ソリューションの作成及びアプリケーションの使用をUIレベルにする。NSLは、冗長性を減じ、ソリューションコンポーネントの再使用率を上げ、最も重要なことには、分化又はソリューションの経路を最適化する多くの特徴を有する。

【0095】

[00248] 行列ベース/タイルベース手法：情報技術の世界は機械エージェントを中心にして築かれている。各プログラミング言語は独自のキーワード、演算子、記号、及び関数のセットを有する。機械エージェントへの命令は明確であるべきであるため、各キーワード、演算子、記号、及び関数は明確な意味を有する。これらの関数は、各プログラミング言語で同じ意味を有するか、又は異なる方法で表現され得る。タイルベース手法では、各タイルは、各プログラミング言語及びNSLの下で特定の関数と同等に扱われている。各タイルにおけるこれらの同等の関数は、機械エージェント/システムへの同じソリューション論理を表現する。技術的には、NSL+技術フレームワークはプログラミング言語に等しい。

10

【0096】

[00249] 技術翻訳フレームワーク(TTF)：NSLの最も重要な属性の1つは、NSL方法及びNSL技術フレームワーク(NSL-TF)を使用して制御される分化に対処する同じ原理に基づいて任意のプログラミングコードを自然言語様のNSLフォーマットに変換し、それにより、ユーザ又はステークホルダの直接使用及び影響に向けて論理を表面化する能力である。TTFは、各プログラミング言語のキーワード、演算子、記号、及び関数並びにNSLにおけるその表現をカプセル化する行列ベース/タイルベース手法を乗り越える。TTFは、コードの構造を分析し、コードが書かれたプログラミング言語を識別し、行列を使用して、あらゆるキーワード、演算子、記号、関数、又はそれらの組合せに一致するNSL均等物を取得する。

20

【0097】

[00250] 技術再翻訳フレームワーク(TRF)：NSLがもたらす革新的フレームワークは、NSLで構築された任意のソリューションを任意のプログラミング言語に変換する能力である。TRFは、TTFが基づくのとおり行列ベース手法に基づく。TRFはNSL構造体を理解し、プログラミング言語における一致するキーワード及び関数を識別し、それにより、ユーザにより選択される任意のプログラミング言語でそのコードを構築する。

30

【0098】

[00251] 360° Any to Any(A2A)：TRFはTTFと一緒に、任意のプログラミング言語又は任意の自然言語のソリューションを任意の他のプログラミング言語又は自然言語に変換する能力である360° Any to Any(「A2A」)と呼ばれるレジームの全ライフサイクルを完成させる。意味が一定であり、種々の自然言語にわたり表現することができるのと全く同じように、ソリューション論理も任意のプログラミング又は自然言語の基盤で表現することができる。NSLでは、ソリューションは規範的情報として知られる特別なクラスの情報である。潜在状態で表現されている規範的情報は、作用を受けると現実になり、その規範はクラスレベルで実行され、それらのメンバが到着すると、現実はトランザクションレベルで生じる。定義されたクラスに到着した任意のメンバは、クラスと同じように挙動することになる。これらは従来のプロセスに取って代わる。到着したイベントは潜在テキストから選択され、それが生じると、全てが自然言語フォーマットで表現可能になる。換言すれば、分化が適宜コンテキスト的に表現される、NSLが従う特定の分化サイクルがあり、NSLによるそのような分化は、任意の基盤で表現可能である。NSLは、TTF及びTRFを通して、各基盤に組み込まれたソリューション論理を抽出し、元の基盤で扱われるのと同じようにそれらを扱うことが可能である。このA2Aは、入力と同じ場合、種々のプログラミング又は自然言語で構築されたあらゆるソリューションで出力が同じであった原理を用いてテストされてきた。端的に言えば、NSL構造

40

50

体はハブとして作用し、他のプログラミング言語はスポークのようである。例えば、プログラミング言語を別のプログラミング言語に翻訳する必要がある場合、そのプログラミング言語はNSLと呼ばれるハブにアクセスしてから、別のプログラミング言語として分岐する必要がある。

【0099】

[00252] NSL言語非依存性：NSLは実体ベース手法をとり、意図履行の実体 - 状態組合せの作成に基づいてソリューションを作成する。基礎は分離原理への依拠にあるため、実際に重要なのは、「現実実体」及び「具象」実体の背後にある現実である。具象目的で使用される言語は重要ではない。その意味で、NSLは言語非依存である。

【0100】

[00253] クエリベースソリューション開発(QBSD)：全てのソリューションは、変化を理由としてのみ生じる。変化がない場合、ソリューションはない。変化が生じるには、常に方向性が必要とされ、その方向性はエージェントにより提供される。エージェントは、方向性のみならず、変化に必要なエネルギーも提供する必要がある。したがって、エージェントの存在下で、全ての変化は状況に応じて行われ、全ては1つの所望のCESから別のCESへの移動に関するものである。これは一連のステップを通して生じる。一連のステップに到達するためには、疑問文が必要であり、すなわち、正しい質問を尋ねる必要がある。ソリューションは、何が、何故、いつ、及びどこでという全ての質問が応えられる場合のみ、構築される。各質問は行動を開始し、物理的な性質を持ち、時間及び空間において表現することができる。NSLは、クエリベースソリューション開発(QBSD)と呼ばれるクエリを使用してソリューションを開発する非常に単純で強力なツールを提供する。QBSDは予め定義される一連の質問であり、応答された場合、NSLでのソリューション作成に役立つ。NSLでのソリューションの作成は自然言語テキストのような構造に基づくため、NSLでのソリューション作成は、これらの予め定義された1組の質問に応答するのと同じくらい単純である。これは、ユーザが、機械により提示された質問に応答して、所望のソリューションを構築するインタラクティブシステムである。

【0101】

[00254] 自然言語様のイニシアチブ：NSLは、アプリケーション論理をコンピュータに自然言語のように伝達する。しかしながら、文章がその自然言語の文法を用いて組み立てられていない場合、構造体は自然言語のように読めないことがある。NSLは、ソリューション構造体において提供される情報を使用し、既存の言語ライブラリの助けを借りて、自然言語のように読める文章を作成し、本でソリューションを読むのに似た経験をユーザに提供する。NSLは、文章の構造において最適な情報を使用するため、構造体をあまり冗長的にしない。情報が最適ではない場合、NSLは、エージェント、実体、属性等の種々のソリューション設計レベルで利用可能な情報をチェックすることにより、サーチ範囲を広げる。サーチ範囲を広げても最適な情報がない場合、文章は、一般レベルでありあまり細分化されていないが、適切なコンテキストで組み立てられる。表現すべき情報が、文章構造体の一部として望まれるよりも多い場合、追加情報は情報層に付加されて、変化単位を更にサポートする。

【0102】

[00255] 自動動詞割り当て(AVA)：大半のシナリオでは、動詞は、意図言明において反映される変化の記述子に過ぎない。記述は一般レベル又は詳細なレベルにすることができ、物事は人間エージェントにとって意味のあるように表現される。一般に、動詞は、設計者が機能を与えることを意図する場合を除き、NSLでいかなる機能的価値も有さない。しかしながら、動詞は、意図言明が自然言語のように読めることを保証するために、あらゆる意図言明で必須である。NSLで新しいソリューションを作成するに当たり、意図言明において動詞を選ぶという柔軟性がある。しかしながら、TTFを使用してプログラミング言語からNSLに変換された既存のソリューションでは、ベースプログラミングコードは一般に動詞を含まないため、動詞は自動的に生成されない。自動動詞割り当ては、自然言語に近いように動詞を機能に自動的に割り当てる、NSLにおける拡張ツール

10

20

30

40

50

である。AVAは、NSLで作成された動詞ライブラリを使用して、動詞句構造体に最も適した正しい動詞を選択する。

【0103】

【00256】 実施ベースソリューション開発 (PBSD) : 問題を解決するあらゆる物事はソリューションである。問題は常に、エージェントのコンテキストにある。NSLパラダイムでは、ソリューションはもはや、一般に理解されるように技術に限定されない。NSLは、世界の仕組み原理に基づく実体ベースのソリューションモデルである。ソリューションは日々のプラクティスに存在する。問題を解決するあらゆるプラクティスは、特定の結果を提供するために実体がいかに相互作用するかについてである。NSLは、エージェントにより記録された最良のプラクティスを捕捉し、所与のプロセスを通して重要な実体を識別する能力を有する。その後、特定の実体をクラスのステータスに昇格させることにより、一般的なソリューション論理を捕捉することができる。NSLは、ユーザが、プラクティスから導出されるアクションを実行するための機械支援指針を提供する。

10

【0104】

【00257】 基盤独立性 : 同じ意味を様々な言語で表現することができるのと全く同じように、ソリューション論理も多くの基盤を通して表現することができる。NSLでは、プロセスと情報との区別がなくなりプロセスは、情報に基づいて動作する特別なクラスである。情報はそれ自体、非物理的であり、概念的な性質のものである。情報は、実体の独自性により表される。全ての独自性は、何れかの基盤を通して表現すべきである。情報がそれ自体を表現するための基盤の存在が必要である。換言すれば、分化が適宜コンテキスト的に表現される、NSLが従う特定の分化サイクルがあり、NSLによるそのような分化は任意の基盤で表現可能である。例 : 「0」又は「1」は、空若しくは満杯であるグラスによって表すことができ、又は空のトランジスタ若しくは電子で一杯になったトランジスタにより表すことができる。

20

【0105】

【00258】 ソリューションの定量化 : 情報が情報理論において「ビット」で定量化されるのと全く同じように、NSLは、実体関係間の距離の識別を通してソリューションを定量化する。(a) バイナリイベント : 実体関係間の距離は、あるCESから別のCESに行くために生じるバイナリイベントの最小数に関して測定することができる。例えば : 実体1が「A」であり、実体2が「AB」であり、実体3が「ABC」である場合、実体「1」から実体「3」に行くまでに生じる必要があるバイナリイベントの最小数は「2」である。その原理は、それらの分化が、無視又は認識される場合、融合又は2つの実体が同一になることを生じさせるといものである。NSLは構造とプロセスとの違いをなくし、重要なのは分化の方向性のみである。(b) 空間 : 各CESは明示又は黙示され、空間において動作するため、どれくらいの距離がカバーされたかを集めることができる。(c) 時間 : 各CESに関連付けられた明示又は黙示の時間があるため、潜在的に時間を評価することができる、したがって、距離は時間で測定することができる。(d) リソース消費 : 物事があるCESから別のCESに移る場合、リソースと呼ばれる、1組の参加する実体を必要とする。あらゆるリソースの背後には、割り当てることができる値があり、したがって、各CESに必要なリソース消費を測定することができる。

30

40

【0106】

【00259】 冗長性をなくすことによる論理圧縮 : プログラミング言語では、任意の既存のシステムで論理をデコードするのに特別な労力を必要とする。複雑な論理のデコードは、新しいコードを書くよりも時間が掛かるため、プログラマは既存のコードを利用しない。プログラマが書く新しいコードは、システム間の冗長性を増大させる。NSLは自然言語であり、完全にトランスペアレント且つサーチ可能であるため、冗長性をなくすことができ、既存のソリューションコンポーネントを再使用することができる。

【0107】

【00260】 ソリューション論理圧縮を理由にした計算リソースの低減 : NSLは冗長性をなくすことにより論理圧縮に役立ち、あらゆることはNSLでトランスペアレントであ

50

り、ソリューションは単純なサーチメカニズムを通して処理されるため、任意のプログラミング言語と比較して、NSLでのソリューション処理ではRAM及びCPUのような計算リソースが低減される。

【0108】

【00261】 エンジニアリング設計へのNSL原理の適用：エンジニアリング設計プロセスは、エンジニアが機能的な製品及びプロセスを作成するに当たり使用するステップの方法論的な連なりである。プロセスは、プロセスの部分が多くの場合、別の部分に入ることができるようになる前、多数回、繰り返す必要があるという点で高度に反復的である。繰り返される部分及び任意の所与のプロジェクトにおけるそのようなサイクルの回数は様々であり得る。NSLにおけるあらゆる実体は、情動的で高度に分化しており、実体の組合せは更なる分化に繋がる。加えて、NSL原理は、所望の組合せ状態に達するまで、各部分が分化する都度、全体設計を構成するコンポーネントを識別し、部分を組合せて、新しい別個のものを作成することによりエンジニアリング設計に適用することができる。それらの部分間の反復は、関数又はプロセスと同等である同じ拡張CES原理により動作する。

10

【0109】

【00262】 プロセスvsサーチ：プログラムは、イベントにどのように反応するかの論理をレイアウトする。入力であるイベントは、論理により定められたように適切なデータベース項目を使用し、出力を生成する。出力は、データベースの不可欠な部分になる。プログラミングには、プログラムに含まれる論理に基づく多くの処理及びデータベースに含まれる実体の処理が必要である。これとは対照的に、NSLでは、システムは、ソリューションが取り扱われる方法を完全に変換するサーチ原理にのみに依拠する。NSLは、ソリューションに適切のように、任意のイベントがCESを通してとることができる、特別なクラスの情報に似た可能な情報経路を識別する。結果として、ソリューション論理及びトランザクション論理は同じままでありながら、計算リソースの消費が大幅に低減し、したがって、プロセスを終了させる。

20

【0110】

【00263】 コンテナ：NSLでは、基礎的な変化及び値の作成は変化単位で行われる。あらゆる変化単位は、定義された1組の境界を有する。NSLでは、あらゆる変化単位はコンテナであり、トランザクションはコンテナ内で生じる。あるコンテナからの出力は、ソリューション論理で設計されているように後続のコンテナの1つにおける可変実体である。コンテナシステムは、アイデンティティを割り当てる状況で重要であり、変化単位の一部としてエージェント及び実体を定義し、実体の一部として属性を定義する等々により階層及び構造を作成するように最良に設計される。様々な種類のコンテナがあり、様々なコンテナは様々な数のスロットを有する。各コンテナには、最後のCUまで、CU1、CU2等々のシーケンスに基づいてアイデンティティが自動的に割り当てられる。NSL内のあらゆる実体は一意的アイデンティティを有する。実体は複数のアイデンティティを有することができる：1つは基本アイデンティティであり、残りは状況的アイデンティティである。それは、特定の変化の状況では、スロットが役割に基づいて割り当てられることを意味する。実体が複数の変化単位に参加している場合、複数のアイデンティティを有することになる（参加している各変化単位で1つのアイデンティティ）。したがって、実体は1つの個々のアイデンティティと、その他の変化ユニットで占有している空間を理由にした1つ又は複数のアイデンティティとを有することになる。各CUに参加しているエージェント、実体、及び属性はまた、ある順序で自動的に付番される。各CUにおける潜在的な各CESは、生成された個別のIDを有するが、設計により、CU中の1つのみのCESがトリガー状態である資格を有する。

30

40

【0111】

【00264】 NSLフォーマットに変換されたドキュメント：あらゆる自然言語は、記述的、規範的、疑問的、及び感嘆的言明を有する。規範的言明は、アクション又は変化に関わる場所であり、記述的言明は情報を提供する。感嘆文は強調することに関するものである。全ての記述的言明及び規範的言明は、疑問的言明に答えることから生じる。質問及び

50

答えは、実体、実体状態組合せ、又は拡張CESに関し、そしてこれらは、動作の優先度に基づいて時間及び空間に適用される。NSLは、任意のドキュメント中のテキストを記述的言明及び規範的言明に変換する能力を有し、ここで、規範的言明はCUに類似しており、記述的言明は情報として規範的言明に添付され、情報層に配置される。

【0112】

[00265] 感覚ベースのユーザインターフェース：NSLの範囲は技術ソリューションを超え、その範囲は世界に存在する全てのソリューションに及ぶ。NSLは、実体ベースのソリューションモデルであり、パターン意識を通して実体を認識するための高度モデルを使用し、それらの実体を入力としてとり、環境への影響に関連する実体及び関連しない実体を分離した後、他の無人車両で行われていることと同様に、正しい種類の出力を作成する。ソリューションは主に技術関連ソリューションであるため、殆どの場合、インターフェースは画面及び音声に制限される。NSLは、技術ソリューションを超えて拡張されているため、感覚ベースのユーザインターフェースと呼ばれるインターフェースに追加された重要な機能であり、主な五感：視覚、触覚、聴覚、味覚、及び嗅覚並びに追加の感覚により入力を生成することができる。追加の感覚は、x線、電波等のように、人間の感覚では捕捉できないが、機械エージェントならば捕捉することができるものである。感覚は、周囲環境の計測に役立ち、十分な情報を得た上で決定するのに役立つ。NSLはデバイス間で通常の画面を超えてユーザインターフェースを拡張し、その範囲を大幅に拡大する。システムへの入力は、マインドに残った環境中のパターンに基づく。NSLは、適しているものを記憶し、不適切な入力を無視する能力を有する。NSLは、現実世界における実体を認識し、ユーザが探し求めているソリューションに適宜応答する。

10

20

【0113】

[00266] NSL及びDNA：私達は情報により駆動される世界に生きている。ソリューション環境への情報の貫入を妨げる既存のプロセスバリアと同様に、生体のDNAに含まれる情報に関してもバリアが存在する。サーチ可能且つ読み取り可能になるように、DNAに含まれている情報をNSLフォーマットに変換する機会がある。

【0114】

[00267] 人工ニューラルネットワーク：各意図言明(SI: statement-of-intent)は、まるでニューロンであるかのように動作する。SIにおけるイベントは、電位の閾値に達したときにニューロンが発火し、変化を他のニューロンに、この場合は別のSIに伝播するようなものである。ソリューション環境におけるSIの集まりは、ニューラルネットワークのような機能を果たす。人工ニューラルネットワークは勢いを得ており、NSLに固有のソリューション設計は、これらの原理の有効な実施に最も資する。

30

【0115】

[00268] フローチャート論理コンバータ：これは、ソリューション論理が既存のフローチャートから抽出される、実施ベースソリューション開発の一形態である。フローチャートの構造体は、別の文章に繋がるある文章のようであり、例えば、NSL用語では、GSIを実現するために一連のLSIが相互作用する。しかしながら、フローチャートは一般に、各文章に参加している実体、エージェント、及び属性のいかなる追加情報も有さない。NSLは、ソリューション論理を任意のフローチャート中のプロセスから抽出し、それを情報に変換する。抽出された情報を使用して、NSLでの変形経路を構築することができる。プロセスが、同じGSIに到達する複数のシナリオに分岐する場合、NSLでの関連段落として作成される。

40

【0116】

[00269] リソース最適化及びアイドル時間識別フレームワーク：NSLはまた、ユーザが利用可能なリソースの全ての最適な使用を自由に定義することができるリソース最適化及びアドレス時間識別フレームワークも有する。このフレームワークは、価値創造活動への実体の参加をリアルタイムで追跡する自動化された技法を有する。あらゆる変化単位及び実体状態組合せは、空間及び時間のイベントログに接続されているため、NSLは実体の使用状況メトリックをリアルタイムで捕捉できる。NSLは、実体がトリガー状態に

50

参加した回数及び変化単位がトリガーされた回数を捕捉する能力を有する。これらの使用率は、一定期間にわたり又は任意の所与の時点で捕捉することができる。最適な分化原理は、ユーザが分析を行い、リソース最適化フレームワークに関する意思決定能力を向上させるのに役立つ。リソースの使用量及び最小限のアイドルは、コストパフォーマンスを最大化するのに役立つ。

【 0 1 1 7 】

[00270] メタソリューション論理実体：NSLでは、ソリューションは規範的情報として知られる特別なクラスの情報である。潜在状態で表現される規範的情報は、作用を受けると現実になる。規範はクラスレベルで行われ、メンバが到着すると、現実がトランザクションレベルで生じる。ソリューション設計はクラスレベルで設定され、クラス自体は、ソリューション設計が必要とされるのと同じくらい分化できる。例えば：人がクラスレベルである場合、読み書きができる人及び読み書きができない人は、そのクラス内のサブクラスである。関係はクラスレベルで定義される。土台となる基礎原理は、相互作用及び結果に関して、メンバの挙動は、それが属するクラスの挙動と同じであることである。したがって、トランザクションが生じる場合、システムは、トランザクションが属する実体クラスを特定する。論理は、一般クラスから借りられ、視座は最も具体的なメンバから高度に一般的なクラスまで様々であることができる。これは、「ソリューションレベルトランザクションカウント」と呼ばれるどのくらいの人々がソリューションの論理を使用したかを識別するのに役立つ。これは、動画が試聴回数を表示する方法と同様である。

【 0 1 1 8 】

[00271] トランザクションソリューション論理実体：ソリューションの建築者はクラス及びサブクラスを定義し、潜在レベルの段落を作成する。イベントはメンバレベルに到達し、適切なクラスを選択する。望みが形成され、エージェントが選択されることから開始される。クラスはそれら自体、任意の数のサブクラスを追加することにより高度に分化することができる。トランザクションソリューション論理実体は、ユーザがリアルタイムのソリューション設計の使用状況メトリックについての有意な洞察を抽出するのを助ける。最も需要の多いソリューション設計コンポーネントは容易に識別され、全ての参加ステークホルダの繰り返しの活動に広く使用される。全てのトランザクションは、ソリューションクラスのメンバである。これは、「実体使用カウント」（特定の期間中に実体が何回使用されたか又は特定の時間に実体が何回使用されていたか等）の識別に役立つ。トランザクション及びそれらのカウントは、ユーザが価値判断の原理をアンロックして、実体の在庫及び計画メカニズムに優先度を付けるのに役立つ。

【 0 1 1 9 】

[00272] NSLに変換された論理演算子：ソリューションの世界（World of Solution）は、技術又は非技術の何れか一方で3つの主な論理演算子「and」、「or」、及び「not」をソリューション設計で使用する。NSLはソリューションに対処するため、NSLが基づく分化原理に組み込まれたこれら3つの論理演算子に対処するメカニズムが存在するのは必須である。（a）演算子「AND」：NSLの構造体は、GSIに到達するためには、全てのLSIが履行される必要があるようなものである。したがって、各文章を順番に履行する必要がある場合、論理演算子「AND」はNSLの設計により自動的に対処される。（b）演算子「OR」：NSLは、共通のGSIに到達する複数の経路を提供する関連段落を提供する。関連段落は、論理演算子「OR」がNSLにおいて対処される方法を説明する。（c）演算子「NOT」：NSLは、論理演算子「NOT」の機能を包括するあらゆる実体の潜在状態と現実状態との間の切り替えを提供する。

【 0 1 2 0 】

[00273] NSL API：APIは、特定のプロトコルで一方のシステムからの要求を送達し、応答が他方のシステムにより提供される2つのシステム間の仲介者である。NSLは、これらのAPIを自然言語構造体で表現する能力を有し、それによりこれらのAPIをより直観的なものにする。NSLは、任意の既存のシステム、ミドルウェア、又はオペレーティングシステムをNSLフォーマットに変換することができる。端的に言えば

10

20

30

40

50

、NSLは、ユーザ経験を維持しながら、任意の既存のソリューションとシームレスに統合する自然言語構造体で開発されたAPIを有する。

【0121】

[00274] 会話的ソリューション設計：会話的ソリューション設計はQBSDの変形である。NSLが行う意義深いことの1つは、機械エージェントとの対話を、特定の規定された手順に従って人間エージェントと対話するのにより近い感じにすることである。したがって、会話的ソリューション設計とは、機械エージェントとの会話によりソリューションを作成する能力である。

【0122】

[00275] NSL構造体へのプログラミングコードの添付：プログラミング言語がTTFを使用してNSLに変換される場合、プログラミング言語でのコード構造体とNSLでのソリューション構造体との間に所与のレベルの相関が確立される。したがって、NSLはプログラムのセグメントをNSLで構築された各文章に添付して、ドキュメント化の目的を果たす。ユーザは、変換されたコードをNSL構造体の適切な部分に添付するに当たり大きな価値を導出し得る。

10

【0123】

[00276] プログラムでのオープンソースソリューション論理のキュレーション：これはリエンジニアリングと同様であり、オープンソースプログラムがTTFを使用してNSLに変換されると、NSLは、ソリューションにおける冗長性をなくし、効率を改善するように構造体に対して適切な価値付加を実行するに当たりガイドする。論理は複数のセクションに拡散し、そのような結果に到達するには膨大な労力が必要とされるため、これはプログラミング言語ではかなりの難問であった。

20

【0124】

[00277] リバースエンジニアリングTBUI：TBUIはそれ自体、実体、エージェント、又はCUがどこに現れるかの明確なビジュアルを確立する分化の層である。NSLは、TBUIがソリューションで生じさせた分化を識別し、これらの分化は代替的には、適切な属性としてTBUIを参加する各実体に追加することにより表現することができる。

【0125】

[00278] 実体間の相関係数及び関連：NSL構造体では、あらゆるものは実体であり、視座のみが異なる。任意の視座実体は、任意の他の視座実体と比較されて、それらの間に任意の相関があるか否かを調べることができる。相関がない場合、0であり、その他の場合、0から1の間で変化することができる。完全に陽性の相関又は完全に陰性の相関が両極端にある。

30

【0126】

[00279] NSLの視覚的マッピング：NSLは、視覚的に表すこともできる分化原理についてのものである - 変化単位がいかに見えるか、変化単位がいかに接続されるか、及び変化単位がいかに存在するようになるかも視覚的に表現することができる。各プログラミング翻訳をそれ自体の所与の価値付加を有する視覚的構造体のこれらの要素の1つに適直接続する機会がある。

【0127】

40

[00280] 3つの視点 - 学習、開発、及びユーザ：あらゆる実体はNSLではトランスペアレントであるため、NSLは3つの視点、すなわち、学習、開発、及びユーザの全てをTBUIレベルで提供する。(a) 学習視点 - プログラミング言語のパラダイムでは、プログラミングを学習するプロセス、例えば、必要とされる学位、スキルセット、及び実践の獲得は、場所、時間、及び媒体にわたり分散している。NSLは、TBUIのレイアウトの一環として、任意の新しいユーザが直観的な学習モジュールを通してNSLを学習し、取得した学習スキルの即時査定を行うことができるインタラクティブシステムを提供する。これらの学習モジュールは、非常に短時間で、必要とされるスキルセットを新しいユーザに備えさせ、学習モジュールの終了までにユーザの自己認証を支援する。この学習視点は絶対的にトランスペアレントであり、NSLとのユーザのジャーニーの一環として

50

見ることができる。(b)開発視点：プログラミング言語を使用して準備されたソリューションは、ソフトウェア開発ライフサイクルを辿り、サイクル中に生じることの多くはトランスペアレントではなく、最終ソリューションが機能するシステムから離れて生じる。NSLでは、ソリューション開発ライフサイクル自体は、1組の質問についての取得情報に対して短縮され、NSLにトランスペアレントに現れる全て。(c)ユーザ視点：ユーザレベルでは、データベースの要素の幾つかのみが、プログラミング言語で選択的に現れ、残りは、システム内であるプログラム及びシステム内であるデータベースで生じる。NSLでは、あらゆるものはトランスペアレントであり、ユーザが必要なのは、適切な情報及びその情報を見る決定権だけである。

【0128】

[00281] 実体相互作用に基づく追加情報の抽出：全ての変化は、実体間の相互作用を理由として生じ、したがって、相互作用する実体の1つについての情報が既知である場合、他の実体についての情報を抽出することができる。ペン、紙、テーブル及び相互作用する意図がある人を想像し、プロセスにおいて、書かれた紙が生成された。ペンのみの場合と、ペンがその場所に存在した時間についての情報を有する場合、より広い相互作用原理のこの限られた情報を用いて、他の実体についての情報を集め得る。これは、分析の視点から非常に大きな重要性を有することになる。

【0129】

[00282] 基本変化単位：基本変化単位は、実体と結果として生成されるトリガー状態との間の全てのトランザクション相互作用が生じる基礎単位である。基本変化単位は、基本変化単位が存在する視座に関係なく同じように存在し動作する。視座が高いほど、情報層は多くの情報を有する傾向がある。

【0130】

[00283] 順次変化単位：これらは、基本CUが、トリガーされた場合、1つ又は複数のイベントを通して影響を及ぼすことが可能なCUである。順次CUは「AND」演算子原理を利用する。CES状態は、基本CU及び順次CU内に維持されて、E-CESを形成する。例：お茶を用意するためには、ティーバッグをカップの中に入れ、やかんに水を入れ、やかんの中の水を沸かし、沸いた湯のいくらかをカップに注ぎ、砂糖をカップに入れる必要がある。これらは、お茶の用意における順次ステップである。

【0131】

[00284] 代替変化単位：代替CUは、履行状態又は次の許可状態に達するのに複数の代替が利用可能な場合、ソリューション設計者により利用される。代替のうちの1つのみが展開され、他は消失する。このCUは「OR」演算子原理を利用する。例：支払い選択肢：クレジットカード、ネットバンキング、代金引換払い、商品券の1つを使用して支払うことができる。

[00285]

【0132】

[00286] 並列変化単位：同じGSIに属する2つのLSIが共有実体を有さず、直接的又は間接的な依存性を有さない場合、それらは並列CUと呼ばれる。システムは、並列CUを自動的に識別し、それに従ってラベルを付ける能力を有する。例：フードデリバリーアプリにおいて、顧客がレストランから食べ物を選択し、支払いをすると、食べ物の準備についての通知がレストランに対してトリガーされ、ピックアップ及びデリバリーについての通知が配達人にも送信される。2つのイベントは並列にトリガーされる。

【0133】

[00287] 入れ子式変化単位：入れ子式CUは、トランザクションレベルでのほかに多くの情報という力を与えるために、基本CUに追加される更なるタイプのCUである。入れ子式CUは、基本CUのSSAサイクル内のSSAサイクルのようである。

【0134】

[00288] 下位変化単位：ソリューションエコシステムでは、ある視座からより高い視座に移ると、より低い視座における全てのCUは、より高い視座におけるCUに組み入れ

10

20

30

40

50

られる。より低い視座におけるそれらのCUは、より高い視座におけるCUに対する下位CUである。

【0135】

【00289】 組み込み変化単位：NSLで構築されたソリューションは複数の層からなり得、ソリューションが活動である場合、複数のタスクからなり得る。各タスクは複数のサブタスクを更に含み得、各タスクの視座から、それ自体が変化単位である。NSLは、CU内のCUを層化するこの能力を提供し、上位レベルCUは、下位レベルCUが履行を得た場合のみトリガーされる。2つのタイプの組み込みCUがある - (1) 数駆動式 (number-driven) の再帰変化単位及び (2) 変化単位内の変化単位であるサブ変化単位。

【0136】

【00290】 一時的変化単位：一時的変化単位は、イネープリング変化単位として見られるべきである。一時的変化単位は、他の段落に接続された段落のシナリオがあり、情報交換又はE-CESが複数の段落又は章に及ぶ場合、重要性を増す。例：手紙を投函するためには、「手紙の筆記」が完了している必要がある。最後のLSI、すなわち、それ自体がある変化エージェントのGSIである書かれた手紙を準備する状態は、郵便局のGSIの一時的変化単位になる。

【0137】

【00291】 ソリューションクラス：ソリューションの建築者は、クラス及びサブクラスを定義し、潜在レベル段落を作成する。イベントはメンバレベルに到達し、適切なクラスを選択する；望みが形成され、エージェントが選択されることからいつも開始される。ソリューション論理の視座にある実体をトランザクション論理における実体から分離することが重要である。ソリューション論理では、実体関係の原理をレイアウトする変数は、「任意のLSI」、「任意の人間エージェント」、「任意のペン」、「任意の紙」等である。

【0138】

【00292】 トランザクション的クラス潜在性：トランザクションクラスは、ソリューションクラスのメンバである。トランザクションが実行される場合、ユーザは、ソリューションクラスにより許され、潜在状態にある異なる可能性及び選択肢から選択することができる。例：文字を書くとき、ユーザはボールペン、インクペン、又は鉛筆から選び、緑であれ、青であれ、オレンジであれ関係なく任意の紙を選ぶことができる。

【0139】

【00293】 トランザクションクラス現実性：イベントがトランザクションクラス潜在性に到達する場合、トランザクションクラス現実性が生まれる。各トランザクションは一意である。任意の数のトランザクションクラスを生み出すことができるのはソリューションクラスである。

【0140】

【00294】 入れ子式マインド：全てのソリューションは、接続された分化クラスを訴えて構築することができ、分化はCESで生じたこと又はE-CESを理由とすることができる。あらゆる変化単位で固有なのは、SSAサイクルのようなフラクタルの展開である。エージェントの存在下で、感覚は環境中の物事を検出し、マインドは多くの可能性から適切な選択を行い、身体は、変化サイクルの完了に必要なエネルギーを提供する。存在することになるSSAサイクルに沿うに当たり、実体及び実体に関するCESを収容するあらゆる変化単位に3つの層が添付される。感知は情報層に関し、選択はマインド層に関し、行動は物理層で実行される。SSAサイクルは、ソリューションクラス及びトランザクションクラスの両方に存在する。NSLは、エコシステムの状態の感知及びそれが貢献するか否か（規範の割り当てを通して）の判断、潜在性の査定、潜在性からの選択、並びに物理層における適切な変化ドライバを呼び出すことによる行動に関する全ての可能性を提供する。入れ子式マインドは、あらゆる変化単位に固有の入れ子式変化単位のマインド層である。入れ子式マインドは常に、全ての視座における全ての独立実体の状態及びCESを評価し、評価される状態は良好、不良、又は曖昧であることができる。付随的に、それらのそれぞれ1つの程度に対処することができる。マインドにより査定される状態は、既

10

20

30

40

50

知の状態又は未知の状態の何れか一方である。未知の状態は過去、現在、又は将来に属することができる。マインドは、利用可能な情報に基づいて未知の状態についての物事を予想／査定／推測する。マインドは、可能性を生成し、的外れな可能性を無視する。入れ子式マインドは、必然的な可能性についての価値判断を行い（良好／不良／曖昧）、又は付随的に機会及びリスクを識別する。これは、これらの結果確率に尤度／確率を割り当てる。入れ子式マインドは、情報層において機会／リスクについてエージェントに選択的に通知し、変形経路に介入し、物理機能層において変化ドライバを改変することにより変形経路を改変する。入れ子式マインドの行動は、エージェントが望むリソース最適化に関することを含む。マインド層は、ユーザの選択によりバッチモード又はリアルタイムモードの何れか一方で動作する。

10

【 0 1 4 1 】

[00295] 状況的トリプルアイデンティティ：ソリューションエコシステムは、一般実体及びエージェントが変化単位外部に存在することができず、一般実体又はエージェントによる占有なしでは変化単位が存在してはならないという単純な原理により動作する。エージェント及び一般実体は、変化単位で一緒になり、願いを履行する。エージェント及び一般実体が一緒になると、それらは、更に独自であり分化した実体の組合せを形成する。NSLにおける各一般実体及び変化単位は、情報的であり、一意のアイデンティティを有する。エージェント及び一般実体は、複数の変化単位に参加することができる。それら（エージェント及び独立実体）が参加する変化単位に基づいて、エージェント及び独立実体は状況的アイデンティティを導出する。これらの状況的トリプルアイデンティティは、ユニバーサルソリューション論理ビルダとして機能する。

20

a . 変化単位アイデンティティ：変化単位は、クラスレベル分化した実体がいかに一緒になり、ソリューションに向けての変化の経路を敷設するかに関してソリューション論理を定義する。アイデンティティは、以下のようにして各CUに割り当てられる。

i . ソリューションエコシステムアイデンティティ：NSLは、存在し得る最大数のサブセットを許容する。例：エコシステムが組織である場合、システムが対処する組織数がサブセットのサイズを定義することになる。サブセットが恐らく10,000の組織であることができる場合、5桁アイデンティティ（例：「a b c d e」は5桁アイデンティティである）を提供することができる。

i i . レベル1変化単位サブセット：組織が最大で1000のビジネスを有する場合、4桁を提供する必要がある（例：「a b c d e . 1 2 3 4」はレベル1変化単位アイデンティティであり、ここで、「a b c d e」はソリューションエコシステムアイデンティティである）。

30

i i i . レベル2変化単位サブセット：レベル1変化単位サブセット内の可能な最大サブセットに基づいて、潜在空間、例えば6桁を決定することができる（例：「a b c d e . 1 2 3 4 . A B C D E」、ここで、「a b c d e . 1 2 3 4」はレベル1変化単位サブセットアイデンティティである）。

i v . それよりも多くのレベルのサブセット：上述したレベル1及びレベル2変化単位サブセットアイデンティティと同様に、NSLでは、ユーザは、任意の必要とされる粒度レベルでアイデンティティを作成することができる。

40

v . プロセス：エコシステムの視座を下ると、旧パラダイムにおけるプロセスに似たものに直面する。プロセスは、一緒にストリング化されて所与の結果をもたらす複数の組の「基本変化単位」である。最後の視座が含むプロセスの数に応じて、システムに存在することになると予期されるプロセスの最大数に基づいて、より多くの分化数を割り当てることができる。

v i . 基本変化単位：基本変化単位は、実体と結果として生じるトリガー状態との間の全てのトランザクション相互作用が生じる基礎単位である。プロセス又はサブプロセス内に存在することができるCUの最大数に応じて、メンバシップ数を割り当てることができる。

v i i . 接続された基本CUアイデンティティ：以下のタイプの接続された基本CU

50

のそれぞれは、基本CUアイデンティティのタイプにより識別可能な一意のアイデンティティを有する。

A . 代替CU : これらは、「A」で始まるアイデンティティにより基本CUを更に分化させ、定義される最大サイズまでのメンバを収容する。例：(..... A x x x)。

B . 順次CU : これらは、基本CUが、トリガーされると、1つ又は複数のイベントを通して影響することが可能なCUである。基本CUも同様に他のCUにより影響されるため、先行する順次CUを指定することは冗長である。後続する順次CUには、「S」で始まる分化数が割り当てられる。例：(..... S x x x)。

C . 再帰CU : これらは、再帰的に機能するCU内のCUであり、「RC」で始まる数を割りあてることができる。例：(..... RC x x x)。

10

D . サブプロセスCU : これらは、CU内のCUであるが、各サブプロセスCUへのアイデンティティの割り当てを必要とするCU内のサブプロセスである。「SU」で始まるアイデンティティを有することができる。例：(..... SU x x x x)。

E . トランザクションCU : これらは、クラスレベル基本CUによる許容されるようにトランザクションウィッシュが生まれたとき、生まれる。クラスレベルCUが許容するトランザクションの最大数に基づいて、一意の分化トランザクション数を添付することにより識別される。1つのトランザクションCUが生まれる場合であっても、関連付けられたクラスレベルプロセス全体が、トランザクションプロセスとしてそれ自体を複製することになり、それに従って付番されることに留意されたい。

v i i i . CU内の層 : 各基本CU内には3つの層がある：物理(P)、情報(I)、及びマインド(M)。これらのそれぞれはそれらのイニシャルにより識別することができる。

20

i x . 独立実体スロット : これらは、各レベル内の独立実体スロットである。「E」から始まるアイデンティティを有する。例：(..... E x x x)。

x . レベル1属性スロット : これらは英数字L 1 Aにより識別することができる。

x i . より多くの属性レベルスロット : 同じ方法論に従って、L 2 A、L 3 A等のアイデンティティを割り当てることができる。

x i i . 実体状態組合せ : システムは、基本CU内の2ⁿバイナリ変数の原理に基づいて各スロットに自動的にCES数を割り当てることができる可能性を更に提供する。

x i i i . 状態 : 各スロットは、独立実体スロットであれ、属性スロットであれ、CESスロットであれ関係なく、「潜在」(P)又は「現実」(R)の何れか一方の状態が存在し、そのように識別することができる。

30

x i v . このシステム実施モデルを使用して、変化単位内の全ての視座、層、及びスロットにおけるCU数は一意に識別することができる。付番システムは、分化原理に基づいて任意の実体と任意の実体との間の距離も提供する。

b . 一般実体アイデンティティ : 別個であり、分化(一意)又は未分化(同一)の一方である任意の実体は、一意のアイデンティティを通して表されるものとする。全ての一般実体は、現実であれ具象であれ関係なく、物理的(及び情動的)である。一般実体のアイデンティティ割り当ては、以下のように行うことができる。

i . クラス : これらの実体は、ソリューション設計者の選択に基づいて少なくとも、エージェント、意図、有形資産、無形資産、属性、及び金銭を含む限られたクラスに属する。クラスはCUの原理と同様の原理に基づくアイデンティティを有する。各サブセットの最大サイズがまず決定され、各サブセットの最大サイズに基づいて、サブセットに必要とされる桁数が適宜決定される。

40

i i . 任意の数のサブクラス : 最も粒度の細かい一般実体が導入されるようなときまで、必要に応じて任意の数のサブクラスが生まれ、識別されるものとする。粒度は、ソリューション設計者の視点から何が重要なのかという単純な原理に基づく。

c . エージェントアイデンティティ : 任意の別個のエージェントは、一意のアイデンティティを通して表されるものとする。エージェントは、人間エージェント又は機械エージェントの何れか一方である。アイデンティティ割り当ては、エージェントのタイプに特有

50

である。

【 0 1 4 2 】

[00296] ハブとしてのNSL：NSLでは、自然言語ベースのNSLフォーマットの
みならず、プログラミング、実施、テキスト、プロセス等の幾つかの基盤でのソリュー
ション構築の開発をサポートするプラットフォームとして動作することにより、それらの幾
つかの基盤の何れかでも構築されるソリューションの開発が可能である。同じ意味を様々
な言語で表現することができることと全く同じように、ソリューション論理も多くの基盤
を通して表現することができる。NSLは、プロセスと情報との区別をなくし、NSLで
は、プロセスは、作用を受ける情報の特別なクラスである。情報は、実体の独自性により
表される。全ての独自性は、何れかの基盤を通してそれ自体を表現するものとする。それ
自体を表現するための情報の基盤の存在は必須である。換言すれば、分化が適宜状況的に
表現される、NSLが遵守する特定の分化サイクルが存在し、NSLによるそのような分
化は任意の基盤で表現可能である。NSLは、関連付けられた真の価値を失うことなく、
上述した幾つかの基盤の何れかでソリューションを実行できるようにするハブとして動作
する。

10

【 0 1 4 3 】

[00297] 半トランスペアレントソリューション環境からのソリューション論理の抽出
：半トランスペアレントソリューション環境とは、ソリューションの一部がトランスペア
レントであり、そのユーザインターフェースにより表されるようにシステム外にあり、バ
ランスがシステム内にあり、論理が隠される環境である。NSLは、TTFの使用を通し
て、トランスペアレントであるソリューションの部分をNSLに変換し、ソリューション
ライブラリに存在するものと同様のソリューションを使用してバランスを再び生み出す。

20

【 0 1 4 4 】

[00298] NSLへのSOPの変換：図117を参照すると、標準動作手順（SOP：S
tandard Operating Procedure）は、規範的言明、記述的言明、フローチャート、又は
それらの組合せを含む。規範的言明は、行動又は変化が関わる場所であり、記述的言明は
情報を提供し、フローチャートは、別の言明に繋がるある言明である。NSLは、任意の
ドキュメント中のテキストを記述的言明及び規範的言明に変換する能力を有し、ここで、
規範的言明はCUの物理層に似ており、記述的言明は、情報として規範的言明に添付され
、情報層に座し、フローチャートは、GSIを達成するように相互作用する一連のLSI
と同様である。NSLは、抽出された情報に冗長がある場合、その冗長をなくし、NSL
フォーマットでソリューションを一緒に編み合わせる。NSLは内部に、自然言語で書か
れた数千ものSOPドキュメントをパズして、NSL均等物を作成する分散ドキュメン
トパズシステムを含む。「Parse2Run」と呼ばれこのシステムは、広い適用可
能性、スケーラビリティ、高性能、及び高可用性を達成しており、自然言語で書かれたS
OPドキュメントをパズし理解することができる。SOPのパズは、実体及び実体が
いかに相互作用してプロセスの理解を形成するかについて教えることができる。Par
se2RunがSOPをパズすると、Parse2Runは、NSLプラットフォームの
一番上にあるプロセスの理解を翻訳することができ、次に、NSLプラットフォームの一
番上にあるプロセスはこの手順を実行する。Parse2Runは、自然言語ソリュー
ション（NSL）構造体を利用する、コアコンポーネントの1つである。Parse2Ru
nのコアは、プロセスを定義するテキストの実体及び要素を見つけるNLPエンジンであ
る。ドキュメントを理解する最初のステップは、ドキュメントの発話部分を見つけ出す
ことである。発話部分が理解されると、それらの先行詞により代名詞等の同一指示を解明す
る。Parse2Runは継承関係をトラバースして、曖昧さを更になくす。この後であ
っても、ドキュメントのいくらかの部分は、SOPで書かれた情報に2つ以上の等しく可
能な解釈が存在し得るため、理解されないことがある。Parse2Runは、このポイ
ントをマークし、ソリューション設計者と会話して曖昧さをなくす。Parse2Ru
nはマルチパスシステムであり、あらゆるパスにおいて、特定の側面にフォーカスする。P
arse2Runは、各パスでSOPドキュメントの理解を改善し続ける。パスの種々の

30

40

50

レベルは以下の通りである。

a . コアパス : P a r s e 2 R u n は、ドキュメントのコア言語ベースの理解を形成する。単語の基本形、発話の部分を有し、日付け、時間、及び数量を正規化し、句及び統語的依存関係に関して文章構造を校正する。

b . 参照パス : 基本レベルの理解後、P a r s e 2 R u n は、どの名詞句が同じ実体を参照するかを見つける。これは、概念のサーチ空間を最小化し、ドキュメントの異なる部分にわたり相関を生じさせられるようにする。

c . 関係パス : P a r s e 2 R u n は、決まり文句を見つけ、実体間の特定の関係又はオープンクラス関係を抽出し、人々が言った引用を取得する、等々。

d . 理解パス : システムは、自然言語で表現された概念をより深く理解する能力を有する。

10

e . プロセスパス : P a r s e 2 R u n は、S O P ドキュメントに書かれたものとしてプロセス自体の情報を見つける。意味論を使用して、この情報はN S L 構造体に変換される。

f . エンリッチメントパス : P a r s 2 R u n は、既にキュレートされた情報から実体を富化する。実体情報は、既存のデータモデル又は利用可能なドメインモデルから富化することができる。この富化は、属性を有する実体を更新する。

g . コンプライアンスパス : P a r s e 2 R u n は、コンプライアンスの視点からプロセスを富化する。コンプライアンスは、組織により提供することができ、又は所与の産業に関して政府により指定することができる。このエンリッチメントパスは、A I システムにより違反についてプロセスを監視できるようにする。

20

【 0 1 4 5 】

[0 0 2 9 9] 開発者レス開発 : N S L は、開発者レス開発 (D L D : Developer less development) と呼ばれるプロセスを通して、極最小の人間介入でソリューションを構築する能力を提供する。D L D は、機械支援設計を提供して、ソリューション設計を加速化させる。ソリューションの開始点及び終了点について明確である場合、D L D は、ソリューション設計者が、D L D がソリューションライブラリを分析し、類似するソリューションを識別し、ソリューションへのありそうな経路を表示することに基づいて関連する経路を取り上げて選ぶ複数の経路選択肢を提供する。開発者レス開発 (D V D) は、教師あり又は教師なし機械学習モデルを実施することにより、ソリューションクラスの自動化及び推奨を可能にする。D L D は、教師あり学習手法を使用してL S I 又は変化単位の最近傍推奨の自動化にフォーカスしてソリューションクラスの完全自動化を提供する。本開示は、N S L における最近傍概念を使用して、機械学習予測を行う。N S L は、変化ドライバから変化単位への高レベルの類似性を有する。複数のトランザクションが生じる場合、明確なパターンがあり、それは機械により認識される。D L D は、ソリューション創造者に対する推奨を行うのにソリューションクラス情報及びトランザクションクラス情報の両方を使用する。図面 1 1 6 に示されるように、ソリューションの内容は、種々のプログラム、動画、S O P 、ドリーム開発者作成コンテンツ (Dream Developer Created Content) からT T F 層を通して抽出されている。D L D エンジンは、N S L ソリューションライブラリの一部であるリッチソリューション内容をマイニングする。さらに、トランザクションクラスからトランザクションデータが抽出され、D L D エンジンに供給され、種々のレベルでの距離の識別を可能にする。距離計算は、機械学習アルゴリズムを使用してN S L 実体値、時間、及び空間を使用することにより達成することができる。D L D エンジンは、N L P 、A N N 、最近傍技法 / コンポーネントをソリューション内容の処理に使用する。N L P 技法は、指名実体認識、単語曖昧さ除去、実体同義化コンポーネントを含む。A N N 技法は、確率モデル、文章エンコーダ、ディープラーニングコンポーネントを含む。最近傍技法は、P a y s p a r r - N N 、実体距離、相関係数コンポーネントを含む。D L D エンジンは、種々の順列及び組合せに確率モデルを使用して予測を行う。

30

40

【 0 1 4 6 】

[0 0 3 0 0] N S L は物理的バリアを克服する : 世界は、情動的であるという点からます

50

ます見られつつある。情報は物理的実体を表すのに使用され、複数の物理的実体が各ソリューションで表される。物理的実体は、真理値が保存される限り、複数の方法で表すことができる。一意の実体は単語で表され、同一の実体は数字で表される。例：本、2冊の本。プログラミングの世界では、ソリューションは作成されるが、データがデータベースに記憶される場合、コンテキストは失われる；その物理的実体が意味をなすには、コンテキストを再作成する必要がある。したがって、その物理的実体についての記憶された情報は、実際の物理的物体を正確に識別するには不適切である。また、1つの物理的実体の多くの推論が作成され、したがって、多くのドキュメンテーションが作成されることに繋がる。物理的実体の任意の変化は、その実体が使用されている全ての参照に浸透するわけではない。NSLはシステム全体を維持し、「真理値」が保存される限り、情報的モデルにおける物理的実体の提示は正確且つ動的である。例：人は一塊の情報であり、その人の名前はその人の属性であり、その人のアイデンティティはその人の数値表現である、等々。正しい基盤は、同じ情報を正しい媒体で使用することもできる。例：機械は数値にうまく対処し、人間は、同じ物理的実体の視覚表現によりよく対処する。

10

【0147】

[00301] NSLはプロセスバリアを克服する：従来、プログラミングの世界では、命令セット（プロセス/関数/アルゴリズム）と（入力/出力）データセットとの間に大きな分離がある。したがって、プロセスフローに含まれるもの、プロセスを通して流れる情報、プロセスのステータスの論理全体は全て不透明であり、システム内にある（プログラムは時折、ステータスをファイルにログして、幾つかのビューを提供する）。NSLでは、ソリューション論理全体は、分化に基づいてソリューションクラスとして表現される。そうして構築された論理は、実行時、命令が到着し始めるとき、展開される。これは、ソリューション全体を非常にトランスペアレントに定義するCES状態及びE-CES状態に繋がる。したがって、不透明なプロセス構造体全体は、非常にトランスペアレントで「システム外」の情報に変換される。情報として表現可能であると、自動的にサーチ可能である。

20

【0148】

[00302] NSLはクラスバリアを克服する：ソリューション論理はクラスレベル（ソリューションクラスと呼ばれる）で定義される。トランザクションクラスは、インスタンスがこれらのソリューションクラスに到着し（実行時に）、定義されたソリューション論理が展開されるとき、形成される。したがって、（分化した）クラスレベルにおいて定義される論理は「ペン、紙、人」であり、トランザクションレベルでは、「このペン、この紙、この人」になる、等々。プログラミング言語では、追加のコンテキストなしで記憶された（データベースに）データの場合、そのデータセグメントを、属する特定のクラスに関連付けることができない。したがって、クラス（ソリューション論理を含む）とインスタンス（ソリューション論理を展開する）との間の任意の関係は失われ、データを分析する必要があるたびに、そのコンテキストを再度作成する必要がある。NSLでは、ソリューション論理は情報の特別なクラス（文章として表現される）として表現され、展開される際、メンバ（データベース）を自身に取り付ける。したがって、NSLは、旅程全体及びコンテキスト全体（状況的IDを使用して）を記録し、各クラスとのメンバの関連付けは、データベース中のデータとして維持される。したがって、NSLは、実体間の距離を測定し、相関及び確率を確立する効率的で効果的な方法を提供する。

30

40

【0149】

[00303] NSLはUIバリアを克服する：プログラミング言語では、（グラフィカル）UIは通常、入力フィールド値及び出力フィールド値をレンダリングする。コンポーネントの位置は、プログラマがシステムを構築する際にプログラマにより決定されるように固定される。これは、情報をレンダリングするかなり柔軟性のない方法であり、あらゆる変更はコードレベルで行われ、再テストされ、次に公開される必要がある。NSLは、分化という根本原理に基づくため、レンダリングする必要がある実体の更なる分化としてユーザーインターフェース特性を利用し、例えば、フォントサイズ、色、コンポーネントがど

50

の画面に現れるか、その画面におけるコンポーネントレンダリングの開始位置等は、属性として実体に添付される追加の分化にすぎない。したがって、システムは、これらの最適な分化値（種々の画面サイズに）に基づいてそれ自体をレンダリングすることができる。

【0150】

[00304] NSLは画面バリアを克服する：従来のシステムは、コンピュータ画面上の限られたUIからGUIのレンダリングを有する。したがって、大抵、ユーザは端末に貼り付き、それが一見したところ、技術的ソリューションと相互作用する唯一のメカニズムである。NSLは、感覚ベースのインターフェース、例えば、触覚、視覚、嗅覚、聴覚、味覚を利用して、種々のソリューションと相互作用することができるソリューションを使用して、「画面外にジャンプ」する能力を有するユーザインターフェース及びユーザ経験を提供する。これは、NSLが実体ベースモデルであり、パターン認識を通して実体及び相互作用を理解することができることを理由とする。例えば：カメラは目として機能することができる、世界を見ることができる。

10

【0151】

[00305] NSLは曖昧性バリアを克服する：従来、ビジネスステークホルダと技術者との間には巨大な不一致がある。それぞれが各自の知識境界を有していた。知識及び解釈の移転にはギャップがあった。その場合、チームは解釈を更に複雑にした。技術チームメンバーの人員削減はプロセスを更に悪化させた。ソリューションを民主化することにより、NSLでソリューションを構築するには技術専門家は必要ないため、NSLは、ソリューションを自分達で構築する力をビジネスステークホルダに与え、それにより、曖昧性バリアを亡くす。

20

【0152】

[00306] NSLはマインドバリアを克服する：通常、現在の世界は分析を特殊な機能と見ており、選択事例で分析を使用する。NSLでは、マインドが組み込まれた入れ子式変化単位があらゆる変化単位と一体化されて、エコシステムを常に記録し分析する。入れ子式マインドは、意識を模倣し、状況認識及び状況応答を保有する。入れ子式マインドは、リアルタイム分析を行い、過去、現在、及び未来を分析し、現在又は将来に行うべき全ての行動の選択を行う力を有する。

【0153】

[00307] NSLは人間バリアを克服する：自然言語を理解するように機械をトレーニングし、上手く構築されたライブラリの世界を用いて機械に力を与えることにより、NSLは、ドライバーレスカーと似た開発者レス開発の時代の先駆けとなっている。機械は、開始点及び願いが指定されると、精密な構造体を与えることができ、それにより、ソリューションを書くことにおける人間への過度の依存をなくす。NSLは、人間を機械と同等と見なすことによりソリューション開発に全く新しいシフトをもたらす。開発者レス開発は、クラスレベル及びトランザクションレベルの両方でSSAサイクルを使用してソリューションを改変する動的意思決定に繋がる、利用可能な情報を利用する。

30

【0154】

[00308] NSLはマン - マシンバリアを克服する：今のところ、2つのコードがある - 自然言語及びプログラミング言語。SSAサイクルは、人間エージェント及び機械エージェントの両方で固有である。両コードを同等と見なすことにより、NSLは人間及び機械エージェントシステムに存在するSSAサイクルを遠ざけ、それにより、違いを消失させる。NSLは、機械及び人間が友人のように話すようにする、ソリューションの汎用言語である。これらは、変化単位が、入力に回答して適切な出力が生成されるように機械エージェント - 人間エージェント又は他の機械エージェントにより設計される - により駆動される「コンピュータ」と同義である。機械エージェントは本質的に、入力を消費し、入力間に制御された変化を生じさせ（それらの入力を処理し）、出力を生成する能力において人間エージェントを模倣する。ある意味では、人間エージェントは、目的により導かれる資質を機械エージェントに付与する。

40

【0155】

50

実例及び図面の説明

【00309】 実例 1 は、Javaプログラミング言語で書かれたホテル予約システムプログラムの状況で行われた NSL のテストに関する。この Java プログラムは 700 行のコードを含み、NSL で有効に置換された。その後、NSL は、「技術フレームワーク (NSL - T F) の使用を通して、Java プログラムと同じ結果を生成することについてテストされた。

【0156】

【00310】 実例 2 は、NSL 技術翻訳フレームワーク (NSL - T T F) のテストに関する。NSL - T T F は、Python プログラミング言語で書かれたホテル予約システムプログラムを消費し、NSL に自動的に翻訳する。そのような NSL - T T F が生成した NSL は NSL - T F に供給されて、元の Python プログラムと同じ結果を有効にもたらし、その NSL が過去に書かれた任意のコードに対処することができることを確立する。

【0157】

【00311】 実例 3 は、NSL 技術再翻訳フレームワーク (NSL - T R F) のテストに関する。NSL - T R F は、NSL で書かれた ATM サービスソリューションを消費し、Python 及び C プログラムに自動的に翻訳する。プログラムは、NSL での元のソリューションと同じ結果を生成した。

【0158】

実例 1

【00312】 実例 1 に関連付けられた図面は図面 95 ~ 103 を含む。Java プログラムの目的は、ユーザが、3 つのタイプの客室 - デラックス、スーパーデラックス、及びラグジュアリー - を 3 つのタイプの交通機関及びランドリーサービスを選択するオプションと共に予約できるようにするホテル予約システムアプリケーションである。

【0159】

【00313】 実例 1 ステップ 1 : Java プログラム及び NSL ソリューションの概要 : a) ホテル予約システム Java プログラムは 700 行のコードからなり、b) コードは Java プログラムの 16 のキーワード、6 つの演算子、及び 5 つの記号を使用し、c) このシステムは 96 のシナリオに対応し、d) 同じソリューションが、13 の独立実体、90 の独立実体 (属性)、及び 18 の L S I を使用して自然言語ソリューションで構築される。これらの 18 の L S I は 96 のシナリオに対応することができ、e) 図面 95 は、Java プログラム及び対応する NSL ソリューションの表現の一例を示す図であり、f) そして図面 96 は、技術翻訳フレームワークを使用した NSL ソリューションへの Java プログラムの翻訳の表現の一例を示す図である。

【0160】

【00314】 実例 1 ステップ 2 : NSL がプログラミングコードにおけるソリューション論理を捕捉する様式 : NSL 構造体は要は、重要な実体を一緒にすることについてである。自然言語に大体似ているため、極めて直観的である。NSL でソリューションを構築するシーケンスは、a) 独立実体 (名詞と同等) を作成し、b) 分化のために属性 (形容詞及び副詞と同等) を実体に添付し、c) 変化単位 / 組合せ実体 (文章と同等) を作成し、d) 変化単位 / ローカル意図言明を一緒にストリング化して、グローバル意図言明 (段落と同等) を作成し、e) 妥当な場合、属性としてルール、制約、公式を添付することを伴う。f) 図面 97 A は、Java プログラムから NSL 均等物への 1 対 1 マッピングを通しての属性を有する実体作成の表現の一例を示す図である。g) 図面 97 B は、Java プログラムから NSL 均等物への 1 対 1 マッピングを通しての NSL における関数の表現の一例を示す図である。

【0161】

【00315】 実例 1 ステップ 3 : 静的状態から動的状態への変形 : a) ソリューションが NSL で構築されると、L S I は静的 / 潜在状態にある。イベントが到着する限り (例えば、トランザクションが生じた場合)、実体状態組合せは静的 / 潜在から動的 / 現実に変形する。b) 図面 98 A は、ソリューションが構築されるとき、静的状態の段落の表現の

一例を示す図である。c) 図面 98B は、「顧客情報が捕捉されたとき、イベント到来」として静的から動的に変わる文章の表現の一例を示す図である。d) 図面 98C は、情報が満杯になったとき、動的状態である段落の表現の一例を示す図である。

【0162】

[00316] 実例 1 ステップ 4 : 出力が NSL と Java と同じであるか否かを検証 : Java アプリケーション及び NSL で構築されたソリューションは、複数のシナリオをカバーする異なる入力を使用してテストされた。Java アプリケーションの出力及び NSL ソリューションの出力は、両アプリケーション下の入力と同じであることを所与として、同じである。a) 図面 99A は、Java プログラムの入力及び出力の表現の一例を示す図である。b) 図面 99B は、NSL ソリューションの入力及び出力の表現の一例を示す図である。

10

【0163】

[00317] 実例 1 ステップ 5 : NSL をサポートする技術フレームワーク : NSL は、任意のオペレーティングシステムと共存する技術フレームワークによりサポートされる。フレームワークの基礎原理が実体及び実体の関係の分化経路を定める。NSL の全ての原理は定められ、任意のアプリケーション論理に関係なく同じままであるため、技術フレームワークは標準化される。

【0164】

[00318] フレームワークは以下の主要コンポーネント : a) 管理データベース、b) 管理ユーザインターフェース、c) 実体をエージェントに接続、d) 情報サーチ原理を有し、e) フレームワークは以下のように全ての種類のシナリオに対処する : i) 判決 : フレームワークで提供される属性構造は、ユーザが各段落を一意に分化させるのに役立つ。フレームワークは、独立段落として記憶されるのに可能なあらゆるシナリオに提供された。ii) 共有実体 : 技術フレームワークは、定義されるときは常に空間及び時間イベントを捕捉し、それにより、パターン認識及び推定を可能にする。これにより、ユーザは、いつ実体が次の活動に利用可能になるか推定することができる。iii) 独立段落 : 技術フレームワークは、独立段落としてあらゆる段落を分化させる。

20

【0165】

[00319] 図面 100A は、管理データベースの表現の一例を示す図である。図面 100B は、管理ユーザインターフェースの表現の一例を示す図である。図面 100C は、実体をエージェントに接続する表現の一例を示す図である。

30

【0166】

[00320] 実例 1 ステップ 6 : 動作メトリック - Java 対 NSL : 図面 101 は、NSL 及び Java でホテル予約システムソリューションを構築するのに必要な時間、労力、及びコストを説明する動作メトリックの表現の一例を示す図である。

【0167】

[00321] 実例 1 ステップ 7 : NSL がキーワード、演算子、及び記号に対処する様式 : ホテル予約システム Java プログラムは、16 のキーワード、6 つの演算子、及び 5 つの記号を使用した。以下は、NSL がこれらのキーワード、演算子、及び記号のそれぞれにいかに対処するかの説明である。

【0168】

[00322] キーワード「Public」 : 図面 97A 及び 97B は、キーワード「Public」が Java で使用される方法及び NSL で表される方法を表す。

40

【0169】

[00323] キーワード「Int」 : 図面 97A 及び 97B は、キーワード「Int」が Java で使用される方法及び NSL で表される方法を表す。

【0170】

[00324] キーワード「Char」 : 図面 102A は、キーワード「Char」が Java で使用される方法及び NSL で表される方法の表現の一例を示す図である。

【0171】

[00325] キーワード「String」 : 図面 102B は、キーワード「String」

50

」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0172】

[00326] キーワード「Double」：図面102Cは、キーワード「Double」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0173】

[00327] キーワード「Boolean」：図面97Aは、キーワード「Double」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0174】

[00328] キーワード「For」：図面102Dは、キーワード「For」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

10

【0175】

[00329] キーワード「While」：図面102Eは、キーワード「While」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0176】

[00330] キーワード「If」：図面102Fは、キーワード「If」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0177】

[00331] キーワード「Else」：図面102Fは、キーワード「Else」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0178】

[00332] キーワード「Break」：図面102Gは、キーワード「Break」がJavaで使用される方法及びNSLで表される方法の表現の一例を示す図である。

20

【0179】

[00333] キーワード「Return」：図面97Bは、キーワード「Return」がJavaで使用される方法及びNSLで表される方法を表す。

【0180】

[00334] キーワード「Print」：Javaにおける「Print」キーワードは、プリント機能に使用される。NSLでは、プリントは、実体に属性として添付される「ユーザインターフェース」として表される。

【0181】

[00335] キーワード「Class」：図面97Aは、キーワード「Class」がJavaで使用される方法及びNSLで表される方法を表す。

30

【0182】

[00336] キーワード「Package」：図面97A及び97Bは、キーワード「Package」がJavaで使用される方法及びNSLで表される方法を表す。

【0183】

[00337] キーワード「Import」：図面97Aは、キーワード「Import」がJavaで使用される方法及びNSLで表される方法を表す。NSLでは、再使用可能なコンポーネントである。

【0184】

[00338] 記号「{ }」：図面97Aは、記号「{ }」がJavaで使用される方法及びNSLで表される方法を表す。

40

【0185】

[00339] 記号「()」：図面97Bは、記号「()」がJavaで使用される方法及びNSLで表される方法を表す。

【0186】

[00340] 記号「;」：Javaにおける「;」記号は、ステートメントの終わらせるのに使用される。NSLでは、「;」は文章を終わらせるのに使用される。

【0187】

[00341] 記号「,」：Javaにおける「,」記号は、変数の分化に使用される。NSL

50

では、実体は、実体に添付される属性により分化する。

【0188】

[00342] 記号「[]」：Javaにおける「[]」記号は、機能の列記に使用される。NSLでは、分化は次のレベルの属性を通して行われる。

【0189】

[00343] 演算子「=」：NSLでは、この算術演算子は、値を実体に添付することにより使用される。

【0190】

[00344] 演算子「==」：図面97Bは、演算子「==」がJavaで使用される方法及びNSLで表される方法を表す。

10

【0191】

[00345] 演算子「<」：図面102Dは、演算子「<」がJavaで使用される方法及びNSLで表される方法を表す。

【0192】

[00346] 演算子「+」：これは、NSLでは公式として定義され、属性として添付される。NSL OSは全ての算術機能を扱うことができる。

【0193】

[00347] 演算子「-」：これは、NSLでは公式として定義され、属性として添付される。NSL OSは全ての算術機能を扱うことができる。

【0194】

20

[00348] 演算子「*」：これは、NSLでは公式として定義され、属性として添付される。NSL OSは全ての算術機能を扱うことができる。

【0195】

[00349] 実例1ステップ8：NSLがプロセスをなくす様式：NSLはプロセス及び関数を情報に変換し、アプリケーション論理を情報サーチ原理の範囲に持ち込む。したがって、Javaプログラムにおいてシステム内にあったあらゆる関数は、NSLではエンドユーザによっても閲覧可能である。

【0196】

[00350] 図面103は、プロセスを情報に変換する表現の一例を示す図である。

【0197】

30

実例2

[00351] 実例2に関連付けられた図面は、図面104～108Bを含む。NSLは、任意のプログラミングコードを消費し、同等のNSLアプリケーション論理を自動的に生成するプロプライエタリ技術翻訳フレームワークを有する。次に、NSL技術フレームワークに乗り、任意のプログラミング言語で書かれたプログラムと同じ結果を生成することができる。

【0198】

[00352] 実例2は、NSL技術翻訳フレームワークがPythonプログラミングコードに関して機能することを証明する。これをテストするために、ホテル予約システムプログラムコードを取り上げた。Pythonプログラムの出力は、ユーザが3つのタイプの客室 - デラックス、スーパーデラックス、及びラグジュアリー - を3つのタイプの交通機関及びランドリーサービスを選択するオプションと共に予約できるようにするホテル予約システムアプリケーションである。

40

【0199】

[00353] 実体、属性、変化単位、及び変化単位間の関係（アプリケーション論理）は、トランスレータによりPythonプログラムから自動的に抽出され、抽出されたコンポーネントはNSLに自動的に供給されて、Pythonプログラムと全く同じ出力を自動的に生成した。

【0200】

[00354] 図面104は、NSL技術翻訳フレームワーク方法論の表現の一例を示す図

50

である。

【0201】

【00355】 実例2ステップ1：Pythonプログラム及びNSLソリューションの概要：a) ホテル予約システムPythonプログラムは391行のコードからなり、b) コードはPythonプログラムの8つのキーワード、4つの演算子、及び4つの記号を使用し、c) このシステムは96のシナリオに対応し、d) 同じソリューションが、13の独立実体、90の独立実体(属性)、及び18のLSIを使用して自然言語ソリューションで構築される。これらの18のLSIは96のシナリオに対応することができ、e) 図面105は、Pythonプログラム及びNSLソリューションの表現の一例を示す図である。図面106は、少数のJava構造体及び対応するNSL均等物の例を示す表である。

10

【0202】

【00356】 実例2ステップ2：NSL技術翻訳フレームワーク使用される方法論：NSLの技術翻訳フレームワークは、従来的一般に使用されているプログラミング言語からのキーワード、演算子、及び記号の組合せ及びそれらのNSL均等物を使用する。以下は、トランスレータが以下の用語に対処する少数のキーワード及びそれらのNSL均等物のリストである。

【0203】

【00357】 キーワード「Static」：図面107Aは、キーワード「Static」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

20

【0204】

【00358】 キーワード「Switch」：図面107Bは、キーワード「Switch」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0205】

【00359】 キーワード「Case」：図面197Bは、キーワード「Case」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0206】

【00360】 キーワード「Continue」：図面107Cは、キーワード「Continue」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

30

【0207】

【00361】 キーワード「Transient」：図面107Dは、キーワード「Transient」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0208】

【00362】 キーワード「Long」：図面107Eは、キーワード「Long」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0209】

【00363】 キーワード「Short」：図面107Fは、キーワード「Short」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

40

【0210】

【00364】 キーワード「Byte」：図面107Gは、キーワード「Byte」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0211】

【00365】 キーワード「Default」：図面107Hは、キーワード「Default」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0212】

【00366】 キーワード「Super」：図面107Iは、キーワード「Super」が

50

プログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0213】

[00367] キーワード「Protected」：図面107Jは、キーワード「Protected」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0214】

[00368] キーワード「This」：図面107Kは、キーワード「This」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0215】

[00369] キーワード「Synchronized」：図面107Lは、キーワード「Synchronized」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

10

【0216】

[00370] キーワード「Strictfp」：図面107Mは、キーワード「Strictfp」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0217】

[00371] キーワード「final」：図面107Nは、キーワード「final」がプログラミングで使用される方法及びNSLで表される方法の表現の一例を示す図である。

【0218】

[00372] 実例2ステップ3：出力がNSLとPythonと同じであるか否かを検証：Pythonアプリケーション及び技術翻訳フレームワークを使用してNSLで構築されたソリューションは、複数のシナリオをカバーする異なる入力を使用してテストされた。Pythonアプリケーションの出力及びNSLソリューションの出力は、両アプリケーション下の入力と同じであることを所与として、同じである。

20

【0219】

[00373] 図面108Aは、Pythonプログラムの入力及び出力の表現の一例を示す図である。

【0220】

[00374] 図面108Bは、NSLソリューションの入力及び出力の表現の一例を示す図である。

30

【0221】

実例3

[00375] 実例3に関連付けられた図面は、図面109～112を含む。NSLは任意のNSLソリューションを消費し、全ての主要なプログラミング言語の同等のプログラミングコードを自動的に書くプロプライエタリ技術再翻訳フレームワークを有する。

【0222】

[00376] 実例3は、NSL技術再翻訳フレームワークが、Python及びCプログラムに変換させるように機能することを証明する。これをテストするために、ATMサービスソリューションを取り上げた。NSLソリューションの出力は、ユーザが、入金伝票に記入し、銀行出納係により伝票を確認し、小切手を検証できるようにする能力である。

40

【0223】

[00377] NSL-TRFの共通論理ツリーは、全ての実体、属性、変化単位、及び変化単位間の関係（アプリケーション論理）をNSLノード毎に抽出し、構成パラメータの追加、変数及びデータ型リストの作成、及びディクショナリの作成のような一連のステップを実行することにより、タイルベース構造を使用してプログラミング言語でコードを書く。

【0224】

[00378] 図面109は、NSL技術再翻訳フレームワーク方法論の表現の一例を示す図である。

50

【 0 2 2 5 】

[00379] 実例 3 ステップ 1 : N S L ソリューション並びに同等の Python 及び C プログラムの概要 :

【 0 2 2 6 】

[00380] N S L ソリューションは 3 つの変化単位及び 8 つの実体からなる。

【 0 2 2 7 】

[00381] 同ソリューションは、Python プログラムの 2 1 行のコードに変換される。

【 0 2 2 8 】

[00382] 同ソリューションは、C プログラムの 3 3 行のコードに変換される。

【 0 2 2 9 】

[00383] 図面 1 1 0 は、入力及び出力を有する N S L ソリューション及び同等の Python 及び C プログラムの表現の一例を示す図である。

【 0 2 3 0 】

[00384] 実例 3 ステップ 2 : N S L 技術再翻訳フレームワーク使用される方法論 : N S L の技術再翻訳フレームワークは、従来的一般に使用されているプログラミング言語からのキーワード、演算子、及び記号の組合せ並びにそれらの N S L 均等物を使用する。

【 0 2 3 1 】

[00385] 実例 3 ステップ 3 : N S L - T R F は共通論理ツリーを使用して、実体及び実体の関係を抽出する。コード生成器は、選択されたプログラミング言語のコードを生成する。

【 0 2 3 2 】

[00386] 図面 1 1 1 は、共通論理ツリーにより生成された出力の一例を示す図である。

【 0 2 3 3 】

[00387] 図面 1 1 2 は、N S L からプログラミング言語へのコード生成の一例を示す図である。

【 0 2 3 4 】

図面に関連する N S L 用語

[00388] ソリューション設計 : 図面 1 0 7 A ~ 1 0 7 C は、ソリューション設計の表現の例を示す図面である。

[00389] 実体 : 図面 9 7 A はサンプル実体を表す。図面 2 も、あらゆる別個のものが実体であることを表す。

[00390] 分化実体 : 図面 9 7 A は分化実体を表す。図面 3 は、ペン及び紙が別個で有り、異なる実体である分化実体を表す。

[00391] 未分化実体 : 図面 4 は、2 本のペンは別個であるが、それでもなお他と同じ実体である未分化実体を表す。

[00392] WWW 原理 : 自己説明的。

[00393] 潜在実体 v s 非潜在実体 : 図面 9 7 A は潜在実体「客室」を表す。

[00394] 潜在実体状態 : 図面 1 及び図面 9 8 A は潜在実体状態を表す。

[00395] 現実実体状態 : 図面 1 及び図面 9 8 C は現実実体状態を表す。

[00396] ソリューションエコシステム : 図面 6 はソリューションエコシステムを表す。

[00397] 全能実体 : 図面 8 は全能実体を表す。

[00398] 独立実体 : 図面 9 7 A は、独立実体である「客室」を示す。図面 9 も、属性と一緒にドラッグする独立実体を表す。

[00399] 黙示実体 : 図面 9 7 B では、機械エージェントは黙示実体である。図面 1 0 は黙示実体を表す。

[00400] 凍結実体 : 図面 9 7 A では、客室は凍結実体である。図面 1 1 は、ソリューション設計者が潜在又は現実というバイナリ状態でのみ考慮する凍結実体を表す。

[00401] 実体の削除 : 図面 1 2 は実体の削除を表す。

[00402] 実体の作成 : 図面 9 7 A は、実体「客室」の作成を示す。図面 1 2 も実体の作成を表す。

10

20

30

40

50

- [00403] 実体の変更：図面 1 3 は実体の変更を表す。
【 0 2 3 5 】
- [00404] 属性：図面 9 7 A は、実体「ホテルの部屋」への属性の追加を示す。図面 1 4 はまた、存在について何らかの他の実体への属性の依存を表す。
【 0 2 3 6 】
- [00405] 必須属性：図面 1 5 は、常に実体への参照と共に存在する必須属性を表す。
【 0 2 3 7 】
- [00406] 標準属性：図面 1 6 は、ソリューション環境に基づく属性の標準化を表す。
【 0 2 3 8 】
- [00407] 一般属性、図面 1 7 は、実体がソリューション環境内で置かれた状況に基づいて使用に選択される一般属性を表す。 10
【 0 2 3 9 】
- [00408] 属性のレベル：図面 1 8 は属性のレベルを表す。
【 0 2 4 0 】
- [00409] 現実実体：図面 1 9 は、自然界に物理的に存在する現実実体を表す。図 9 7 A も現実実体「ホテルの部屋」を表す。
【 0 2 4 1 】
- [00410] 具象実体：図面 2 0 は、他の現実実体又は他の具象実体を表す実体を表す。
【 0 2 4 2 】
- [00411] 知覚実体：図面 2 1 は、人間エージェントにより彼ら自身の脳で自然に所与のニューロン状態の形態で作成される実体を表す。 20
【 0 2 4 3 】
- [00412] 情報実体：図面 2 2 は情報実体を表す。
【 0 2 4 4 】
- [00413] 真理値：図面 2 3 は真理値についての理解を表す。
【 0 2 4 5 】
- [00414] 情報内容 - 現実 v s 具象：図面 2 0 は、他の現実実体又は他の具象実体を表す実体を表す。
【 0 2 4 6 】
- [00415] 物理的現実：図面 9 7 A の「ホテルの部屋」は、物理的現実中存在する実体である。図面 2 4 も、物理世界で物理的に存在している全ての実体を表す。 30
【 0 2 4 7 】
- [00416] 変化単位：a) ローカル意図言明：図面 9 8 A は L S I 及び G S I を含む。図面 2 5 は、ローカル意図言明及びグローバル意図言明を表す。b) グローバル意図言明：図面 9 8 A は L S I 及び G S I を含む。図面 2 5 は、ローカル意図言明及びグローバル意図言明を表す。
【 0 2 4 8 】
- [00417] 変化単位のサイズ：図面 2 6 は、変化単位のサイズを表す。
【 0 2 4 9 】
- [00418] イベント：a) バイナリ状態：図面 9 9 A は、潜在状態での各実体を表す。図面 2 7 は実体のバイナリ状態を表す。b) 変動性：図面 2 8 は変動性を表す。c) 定数：図面 2 9 は、状態が常に同じままである実体を表す。 40
【 0 2 5 0 】
- [00419] 入力：図面 9 9 B はソリューションにおける入力及び出力を表す。図面 3 0 は入力及び出力を表す。
【 0 2 5 1 】
- [00420] 出力：図面 9 9 B はソリューションにおける入力及び出力を表す。図面 3 0 は入力及び出力を表す。
【 0 2 5 2 】
- [00421] 経過時間：図面 6 0 は経過時間を表す。 50

【 0 2 5 3 】

[00422] ソリューションの全ライフサイクル：図面 6 7 は、ソリューションの全ライフサイクルを表す。

【 0 2 5 4 】

[00423] 自然言語：図面 3 1 は、自然言語が機能する方法を表す。

【 0 2 5 5 】

[00424] コード v s 自然言語：図面 5 8 は、コードと自然言語との間の差を表す。

【 0 2 5 6 】

[00425] 自然言語ソリューション) a) 静的意図言明：図面 9 9 A は、静的状態の意図言明を表す。図面 3 2 も、意図を表現するだけで、現実に変形する能力を有さない静的意図言明を表す。b) 動的意図言明：図面 9 9 C は、動的状態の意図言明の表現の一例を示す図である。図面 3 3 も実体を生じさせる基礎変形である動的意図言明を表す。

10

【 0 2 5 7 】

[00426] 意味 v s 価値：図面 5 9 は、意味と価値との間の差を表す。

【 0 2 5 8 】

[00427] 価値の定量化：図面 6 1 は価値の定量化を表す。

【 0 2 5 9 】

[00428] 価値の相対性：図面 6 2 は価値の相対性を表す。

【 0 2 6 0 】

[00429] 自然言語と N S L との間の共通性：図面 5 2 は、自然言語と自然言語ソリューション (N S L) との間の共通性を表す。

20

【 0 2 6 1 】

[00430] 論理の段落：図面 6 3 は論理の段落を表す。

【 0 2 6 2 】

[00431] N S L ライブラリ：図面 5 5 は N S L ライブラリを表す。

【 0 2 6 3 】

[00432] 再使用可能コンポーネント：図面 9 7 A は、ホテルの部屋及び再使用可能コンポーネントであるその属性を表す。図面 5 3 もコンポーネントの再使用可能性を表す。

【 0 2 6 4 】

[00433] 分化原理：図面 3 4 は分化原理を表す。

30

【 0 2 6 5 】

[00434] 視座：図面 3 5 は視座を表す。

【 0 2 6 6 】

[00435] 方向性：図面 3 6 は、分化ツリーを上又は下に移動する状況での方向性を表す。

【 0 2 6 7 】

[00436] 実体状態組合せ：図面 9 9 B は、イベントが到着したとき、トリガーが生じるトリガー実体状態組合せを表す。図面 3 7 も非トリガー及びトリガー実体状態組合せ (C E S) を表す。

【 0 2 6 8 】

[00437] A P I v s プログラム：図面 6 5 は、A P I とプログラムとの間の差を表す。

40

【 0 2 6 9 】

[00438] 共有実体：図面 5 6 は共有実体を表す：変化の経路：

a) 制約：図面 9 7 A、客室稼働状況は、システムに課される制約である 2 つの状態のみを有する。図面 3 8 も、多くの可能性間で行われる選択に課される制限にすぎない制約を表す。

b) ルール：図面 9 7 B は、室料を計算するためのルールを述べている。図面 3 9 も殆どの場合、システムに課される一連の制約であるルールを表す。

c) アルゴリズム：図面 4 0 は、ルールと同じであるが、多くの場合、情報システムの状況で使用されるアルゴリズムを表す。

50

【 0 2 7 0 】

[00439] 情報へのプロセスの変換：図面 1 0 3 は、いかにプロセスが情報に変換されるかを表す。

【 0 2 7 1 】

[00440] ITにおける抽象化のレベル：図面 6 6 は、ITにおける抽象化のレベルについての理解を表す。

【 0 2 7 2 】

[00441] エージェントは人間エージェント、機械エージェント、及びマザーネイチャーを含むことができる。

【 0 2 7 3 】

[00442] 人間エージェント：図面 4 1 は人間エージェントを表す（例えば、a）価値創造者：図面 4 2 は価値創造者及び消費者を表す。b）価値消費者、c）チーム、d）所有権：図面 4 3 はエージェントに関する実体の所有権を表す。e）エージェント機能：図面 4 4 はマインド機能、物理的機能、及び情報機能を表す）。

【 0 2 7 4 】

[00443] 機械エージェント：図面 4 7 は機械エージェントを表す。

【 0 2 7 5 】

[00444] マザーネイチャー：図面 4 8 は、変形に影響を及ぼすことが可能な第 3 の種類のエージェントであるマザーネイチャーを表す。

【 0 2 7 6 】

[00445] 判決システム：図面 4 9 は、2つの段落があり、どの段落が意図の履行に必要とされるかをエージェントが選ぶ判決システムを表す。

【 0 2 7 7 】

[00446] ステークホルダエンゲージメントセンタ：図面 5 7 は、ステークホルダエンゲージメントセンタ（SEC）を表す。

【 0 2 7 8 】

[00447] 測定フレームワーク：図面 5 0 は測定フレームワークを表す。

【 0 2 7 9 】

[00448] 分散ミッションコントロールセンタ：図面 7 0 は分散ミッションコントロールセンタを表す。

【 0 2 8 0 】

[00449] 動的ミッションコントロールセンタ：図面 7 1 は動的ミッションコントロールセンタを表す。

【 0 2 8 1 】

[00450] ユーザインターフェース：図面 5 1 は、ユーザインターフェースを通して提示されるエージェントに関する実体を表す。

【 0 2 8 2 】

[00451] NSL技術フレームワーク：図面 6 9 はNSL技術フレームワークを表す。

【 0 2 8 3 】

[00452] リエンジニアリング：図面 5 4 はリエンジニアリングを表す。

【 0 2 8 4 】

[00453] NSL言語非依存性：図面 6 8 は、NSLが言語非依存性であることを表す。

【 0 2 8 5 】

[00454] NSL及びDNA：図面 7 3 は、DNAが情報に変換されることを表す。

【 0 2 8 6 】

[00455] 人工ニューラルネットワーク：図面 6 4 は人工ニューラルネットワークを表す。

【 0 2 8 7 】

図面に関連するNSLの概念

[00456] 世界におけるあらゆるものは、実体及び実体の関係に関して説明することが

10

20

30

40

50

できる（同じことが、ホテルの部屋が属性としてのWi-Fi、レート、及び客室ステータスに関連する図面 2 及び図面 9 7 A を参照して更に詳細に説明されている）。

【0288】

[00457] 任意の情報技術ソリューション設計も実体中心の方法で行うことができる。（同じことが、日数、レート、及びシングル/ダブル過剰状況を組合せとして一緒にすることにより、ホテル室料計算が実体中心の方法で表現されている図面 2 及び図面 9 7 B を参照して更に詳細に説明されている）。

【0289】

[00458] エージェントの観点から別個であるあらゆるものは実体である。（同じことが、ホテルの部屋及びホテル室料が 2 つの別個の実体である図面 2、図面 9 7 A、及び図面 9 7 B を参照して更に詳細に説明されている）。

10

【0290】

[00459] 異なるタイプの実体がある。（同じことが、実体、属性、変化単位、及び経路が異なるタイプの実体として機能する図面 2、図面 3、図面 4、及び図面 9 9 B を参照して更に詳細に説明されている）。

【0291】

[00460] 全ての物事は、実物又は非実物、有形又は無形に関係なく、実体と呼ばれる資格を有する（同じことが、ホテルの部屋が有形の実物資産であり、一方、室料は無形の非実物資産である図面 2、図面 9 7 A、及び図面 9 7 B を参照して更に詳細に説明されている）。

20

【0292】

[00461] 実体は一意である（他と異なり、分化している）ことができ、又は再帰であることができる（他と同じであり、分化していない）。（同じことが、再帰が、一意のラグジュアリー室の部屋数の形態で説明されている図面 3、図面 4、及び図面 1 0 2 D を参照して更に詳細に説明されている）。

【0293】

[00462] 人間エージェントから独立して存在する実体は「現実実体」と呼ばれる。（同じことが、ホテルの部屋が現実実体として（物理的に存在する）存在する図面 1 9 及び図面 9 7 A を参照して更に詳細に説明されている）。

【0294】

[00463] それ自体に価値はないが、ソリューションに関して他の実体を表すことを理由にして価値を導出する実体は、具象実体と呼ばれる。（同じことが、室料が具象実体である図面 2 0 及び図面 9 7 B を参照して更に詳細に説明されている）。

30

【0295】

[00464] 具象実体には 2 つのタイプがある：知覚実体及び情報実体。（同じことが、図面 2 1、図面 2 2、及び図面 9 7 B を参照して更に詳細に説明されている）。

【0296】

[00465] 知覚実体とは、感覚により知覚され、又はマインドにより処理される人間エージェントのマインドに残される実体である。（同じことが、図面 2 1 を参照して更に詳細に説明されている）。

40

【0297】

[00466] 情報実体とは、任意の他の実体：現実、知覚、又は情報を表すためにエージェントにより作成される実体である（同じことが、ホテルの予約がグローバル意図言明を有し、情報実体として機能する図面 2 2 及び図面 9 7 B を参照して更に詳細に説明されている）。

【0298】

[00467] 全ての実体は情報実体により表すことができる。（同じことが、ホテルの予約がグローバル意図言明を有し、情報実体として機能する図面 2 2 及び図面 9 7 B を参照して更に詳細に説明されている）。

【0299】

50

[00468] データも、処理の程度が低い状況で一般に使用される情報である。(同じことが、スーパーデラックス室のレートが、料金についての情報を与えるデータとして機能する図面 99B を参照して更に詳細に説明されている)。

【0300】

[00469] 一意の実体は「単語」により表すことができ、再帰実体は「数」により表すことができる。(同じことが、一意の実体が単語ラグジュアリー室の形態で説明され、数：3 が再帰を表す図面 102D を参照して更に詳細に説明されている)。

【0301】

[00470] 情報実体は真理値を携帯する。(同じことが、図面 23 を参照して更に詳細に説明されている)。

10

【0302】

[00471] 全ての実体は、それらが「現実、知覚、又は情報」実体であることに関係なく物理的である；すなわち、空間及び時間に存在する。(同じことが、現実実体であるホテルの部屋が空間及び時間に存在し、属性である年齢：60、性別：男性、ID証明：アドナンバーは、空間及び時間に到着する情報実体である図面 28 及び図面 98A を参照して更に詳細に説明されている)。

【0303】

[00472] 情報技術 (IT) レベルで対処される全ての実体は、現実実体又は他の具象実体を表す情報実体である。(同じことが、現実実体であるホテルの部屋が空間及び時間に存在し、属性である年齢：60 が、名前ラジュを有する現実実体にタグ付けされる図面 22 及び図面 98A を参照して更に詳細に説明されている)。

20

【0304】

[00473] エージェントも実体である。(同じことが、顧客がエージェントとして機能する図面 98A を参照して更に詳細に説明されている)。

【0305】

[00474] 全ての変化はエージェントによって生じる(同じことが、名前、客室タイプ、稼働状況、及び部屋数の入力があるホテル予約エージェントにより行われる図面 99A を参照して更に詳細に説明されている)。

【0306】

[00475] 3種類のエージェントがある - マザーネイチャー、人間エージェント、及び機械エージェント。(同じことが、室料計算及び予約IDが機械エージェントにより生成され、一方、客室宿泊情報、稼働状況、及び顧客の名前は人間エージェントにより入力される図面 41、図面 47、及び図面 99B を参照して更に詳細に説明されている)。

30

【0307】

[00476] 人間エージェント及び機械エージェントは目的により駆動される。人間エージェント及び機械エージェントは何らかの実体状態を好み、何らかの実体状態を避け、何らかの実体状態にはニュートラルである。(同じことが、室料計算及び予約IDが機械エージェントにより生成され、一方、客室宿泊情報、稼働状況、及び顧客の名前は人間エージェントにより入力される図面 99B を参照して更に詳細に説明されている)。

【0308】

[00477] これとは対照的に、マザーネイチャーにより生じる変化の背後に目的はない。

40

【0309】

[00478] 以下から、マザーネイチャーを明示的に言及する場合を除き、人間エージェント及び機械エージェントの限られた状況でのみエージェントについて言及するものとする。

【0310】

[00479] 単語「エージェント」は一般に、人間エージェント及び機械エージェントを暗黙的に示すものとする。(同じことが、室料計算及び予約IDが機械エージェントにより生成され、一方、客室宿泊情報、稼働状況、及び顧客の名前は人間エージェントにより入力される図面 99B を参照して更に詳細に説明されている)。

50

【 0 3 1 1 】

【00480】 人間エージェント及び機械エージェントは便宜上、ステークホルダ及びコンピュータと代替的に呼ばれることができる。

【 0 3 1 2 】

【00481】 NSLは、分化原理と呼ばれ得るものを前提としている。(同じことが、ホテルの部屋及び室料が完全に分化される図面34、図面97A、及び図面97Bを参照して更に詳細に説明されている)。

【 0 3 1 3 】

【00482】 他のことが同じままで、新しい実体が既存の実体状態組合せ(CES)に追加又は作成される場合、新しいCESは分化がより進む。(同じことが、年齢、性別、及びID番号を添付する場合、生徒は分化がより進む図面107Aを参照して更に詳細に説明されている)。

10

【 0 3 1 4 】

【00483】 同様に、他のことが同じままで、実体が既存のCESから削除される場合、新しいCESは分化が退化する。(同じことが、予約されたラグジュアリー室は、より多くの情報が名前、タイプ、レート、稼働状況、宿泊日数、及び予約IDの携帯で添付される場合、完全に分化する図面103を参照して更に詳細に説明されている)。

【 0 3 1 5 】

【00484】 一般的には、実体が分化するほど、実体が消費する情報(ビットで定量化される)は多くなる。分化の程度が低いほど、実体が消費する情報は少ない。(同じことが、予約されたラグジュアリー室は、より多くの情報が名前、タイプ、レート、稼働状況、宿泊日数、及び予約IDの形態で添付される場合、完全に分化する図面103を参照して更に詳細に説明されている)。

20

【 0 3 1 6 】

【00485】 ソリューション設計では、存在する分化の階層がある。(同じことが、英数字データ型が更に添付される(構造階層)階層形態で添付された予約IDを予約確認が有する図面34及び図面102Bを参照して更に詳細に説明されている)。

【 0 3 1 7 】

【00486】 分化ツリーは、分化の1つ又は複数の層に「属性」呼ぶ複数の依存実体を潜在的に携帯する「独立実体」から始まる。(同じことが、独立実体である予約確認が、英数字データ型が更に添付された依存実体として予約IDを有する図面34及び図面102Bを参照して更に詳細に説明されている)。

30

【 0 3 1 8 】

【00487】 次に、文章に等しい、ローカル意図言明(LSI)のドメイン内に独立実体の組合せから生じる実体状態組合せがある。(同じことが、顧客情報の取得、客室タイプの選択等がCESに存在するLSI又は文章である図面32、図面98A、及び図面98Bを参照して更に詳細に説明されている)。

【 0 3 1 9 】

【00488】 LSIは静的形態で、変化単位を表現又は記述するにも関わらず、独立実体の1つとしてもカウントされる。(同じことが、顧客情報の取得、客室タイプの選択等が静的実体状態で存在する図面98Aに示されている)。

40

【 0 3 2 0 】

【00489】 各独立実体に添付された属性は、文章中の形容詞に等しい。(同じことが、顧客が独立実体であり、年齢、性別、及びID証明が形容詞である図面14及び図面98Aを参照して更に詳細に説明されている)。

【 0 3 2 1 】

【00490】 各LSIに添付された属性は文章中の副詞に等しい。(同じことが、図面14を参照して更に詳細に説明されている)。

【 0 3 2 2 】

【00491】 実体状態組合せの中には、 $2n - 1$ 個のバイナリ非トリガーCESがあり、

50

ここで、「 n 」は独立実体及び属性状態の数である。(同じことが、図面 37 を参照して更に詳細に説明されている)。

【0323】

[00492] CES の 1 つのみが、拡張 CES と呼ばれ、LSI の組合せを生成する、CES の階層拡張を担うトリガー CES としてカウントされる。(同じことが、図面 37 を参照して更に詳細に説明されている)。

【0324】

[00493] したがって、トリガー CES では、全ての独立実体及びそれらの属性が現実状態であることも生じる。(同じことが、図面 37 を参照して更に詳細に説明されている)。

【0325】

[00494] LSI のそのような集まりはグローバル意図言明に繋がり、これは段落の均等物である。(同じことが、図面 32 及び図面 95 を参照して更に詳細に説明されている)。

【0326】

[00495] 分化のそのような階層は、理論における段落内の段落に拡張することができる。(同じことが、図面 34 及び図面 95 を参照して更に詳細に説明されている)。

【0327】

[00496] NSL 文法に等しい分化の形成を導く原理は、自然言語の分布と多くの類似点を有する。(同じことが、図面 34 及び図面 95 を参照して更に詳細に説明されている)。

【0328】

[00497] 分化原理は、集合論、微積分学、生体系、及び多くの他の科学原理に含まれる原理を発端とする。(同じことが、図面 34 を参照して更に詳細に説明されている)。

【0329】

[00498] 実体の最も未分化又は一般化された形態は全能実体と呼ぶことができる。(同じことが、図面 8 を参照して更に詳細に説明されている)。

【0330】

[00499] 所与の数の分化状態がソリューションエコシステムに存在する場合、それらの分化値は無視され、それらの存在のみが認識され、残るものは分化した形態のみである。(同じことが、図面 34 を参照して更に詳細に説明されている)。

【0331】

[00500] これは以下により示すことができる。

【0332】

[00501] 5 つの状態 A、AB、ABC、ABCD、及びABCDE がある場合。状態 ABCDE における「E」の値が無視される場合、前の状態である ABCD に等しくなる。

【0333】

[00502] 4 番目の状態における「D」の値も無視される場合、ABC に等しくなり、システム内に 3 つの ABC がある。

【0334】

[00503] A まで行き、他のあらゆるものがその存在については認識されるが、その値について無視される場合、いかなる分化も添付されていない状態で 5 つの「A」のみがシステムに残ることになる。(同じことが、図面 34 を参照して更に詳細に説明されている)。

【0335】

[00504] 「A」が、人、車、色を有する車等々のいかなる他の分化もないただの実体として認識される場合、そのような最も一般化された形態は、いかなる分化もない実体カウントのみを残す。(同じことが、図面 34 を参照して更に詳細に説明されている)。

【0336】

[00505] ソリューション設計では、CES へのあらゆる追加は、分化ソリューション

10

20

30

40

50

経路に繋がる制約と同じである。(同じことが、制約としての客室の空き状況が部屋を予約すること又は部屋を予約しないことの何れか一方に繋がる図面 1 0 2 F を参照して更に詳細に説明されている)。

【 0 3 3 7 】

[00506] 複数のそのような連続する又は関連する追加又は制約がある場合、これらは、従来のソリューション構築でルールと呼ばれるものと同じである。

【 0 3 3 8 】

[00507] アルゴリズムは、機械エージェントにより実施されるルールである。(同じことが、室料 5 0 0 及び Wi-Fi が利用可能ではなく、ステータスが利用不可であることが、アルゴリズムとしてデフォルトにより機械エージェントによりあらゆるラグジュアリー室に設定される図面 4 0 及び図面 1 0 2 D を参照して更に詳細に説明されている)。

10

【 0 3 3 9 】

[00508] ユーザインターフェース及びソリューションアーキテクチャが分化レベルを独立実体及び属性の最下レベルとする間、情報技術の基礎はこれらの抽象化のレベルを英字、数字、及び記号、そして二進数、そしてトランジスタ又は記憶装置における電磁力まで拡張する。(同じことが、図面 6 6 を参照して更に詳細に説明されている)。

【 0 3 4 0 】

[00509] 自立独立実体がある。(同じことが、図面 9 を参照して更に詳細に説明されている)。

【 0 3 4 1 】

[00510] 独立実体は、イベントがそれらのレベルで又はそれらの属性のレベルで生じ、それらが一部をなす実体状態組合せ (CES) に変更を生じさせることができる程度まで自立している。(同じことが、顧客実体、年齢、性別、及び ID 証明が顧客情報取得変化単位の変更を生じさせる図面 9、図面 9 8 A、及び図面 9 8 B を参照して更に詳細に説明されている)。

20

【 0 3 4 2 】

[00511] 属性は、独立実体に依存する実体である。(同じことが、室料、Wi-Fi、及び客室ステータスが、独立実体であるホテルの部屋に依存する図面 1 4 及び図面 9 7 A を参照して更に詳細に説明されている)。

【 0 3 4 3 】

[00512] 換言すれば、属性はそれらの存在に関して独立実体の恩恵を受けており、独立実体なしでは存在できない。(同じことが、室料、Wi-Fi、及び客室ステータスが、独立実体であるホテルの部屋に依存する図面 1 4 及び図面 9 7 A を参照して更に詳細に説明されている。ホテルの部屋が削除される場合、Wi-Fi 及びステータスは自動的に存在しない)。

30

【 0 3 4 4 】

[00513] しかしながら、イベントはそれらのレベルで独立して生じ、独立実体及び CES の状態を変えることができる。

【 0 3 4 5 】

[00514] 多くのレベルの属性が存在することができる。(同じことが、2つのレベルの属性が客室タイプ実体に存在する図面 1 8 及び図面 1 0 2 D を参照して更に詳細に説明されている)。

40

【 0 3 4 6 】

[00515] 各独立実体は、数兆もの粒子で構成されるため、高度に分化しており、既に膨大な量の情報を携帯している(情報は単に、実体の分化を表すにすぎない)。(同じことが、ホテルの部屋が、属性としてレート、Wi-Fi、及び客室ステータスを有して十分に分化した独立実体である図面 9 及び図面 9 7 A を参照して更に詳細に説明されている)。

【 0 3 4 7 】

[00516] 独立実体は、潜在又は現実のバイナリ状態で存在することを意味する凍結状態の実体として見られるものとする。

50

【 0 3 4 8 】

[00517] 独立実体は、膨大な数の粒子又は情報を携帯する場合であっても、変更に適さない。

【 0 3 4 9 】

[00518] あらゆるCESは潜在又は現実のバイナリ状態であり、イベントの指示に基づいて状態をいつでも変える準備ができています。(同じことが、GSI CESとしてホテルの部屋の予約が静的形態及び動的形態で存在する図面27、図面98A、及び図面98Bを参照して更に詳細に説明されている)。

【 0 3 5 0 】

[00519] 実体が組合せられる場合、実体 - 状態組合せ(CES)が作成される。(同じことが、図面37を参照して更に詳細に説明されている)。

10

【 0 3 5 1 】

[00520] CESは、既に高度に分化した独立実体の組合せであるため、さらに分化する。(同じことが、図面37を参照して更に詳細に説明されている)。

【 0 3 5 2 】

[00521] あらゆるCESはバイナリ状態である。(同じことが、図面27を参照して更に詳細に説明されている)。

【 0 3 5 3 】

[00522] 情報技術では、全ての状態はデジタル的に表現可能である。

【 0 3 5 4 】

[00523] システム内で、CESが永続的に現実状態又は潜在状態である場合、それは定数と呼ばれる。(同じことが、室料が固定値を有する図面29及び図面99Bを参照して更に詳細に説明されている)。

20

【 0 3 5 5 】

[00524] CESが潜在状態と現実状態との間で振れることができる場合、それは変動状態と呼ばれる。(同じことが、客室の空き状況及び顧客が変数である図面28及び図面99Bを参照して更に詳細に説明されている)。

【 0 3 5 6 】

[00525] CESが潜在から現実になる場合、それは作成されたと見られる。(同じことが、図面12を参照して更に詳細に説明されている)。

30

【 0 3 5 7 】

[00526] CESが現実から潜在になる場合、それは削除されたと見られる。(同じことが、図面12を参照して更に詳細に説明されている)。

【 0 3 5 8 】

[00527] CESのサブ状態が潜在状態と現実状態との間の変化を受ける場合、変更されたと見られる。(同じことが、図面13を参照して更に詳細に説明されている)。

【 0 3 5 9 】

[00528] CES視座が高いほど、サブ状態の全てが手状態のステータスに影響することができる場合、変更及び変動性は無限に向かう傾向を有することができることもある。(同じことが、日数及び稼働状況は部屋宿泊情報に影響し、日数及び稼働状況は、より高い視座においてホテルの部屋を予約するGSIに影響する図面99Bを参照して更に詳細に説明されている)。

40

【 0 3 6 0 】

[00529] 全ての「変化」はエネルギーに依存する。

【 0 3 6 1 】

[00530] 全てのソリューション及び全ての価値あるものは、「制御された変化」から生じる。(同じことが、5つの異なるLSIが、一続きで一緒に接続されて、イベントのフローを制御する図面99Bを参照して更に詳細に説明されている)。

【 0 3 6 2 】

[00531] 制御された変化はエネルギー及び方向 - 方向付きエネルギー - を必要とする。

50

【0363】

[00532] ソリューション設計では、全ての変化は「変化単位」で生じる。(同じことが、顧客情報を取得、客室タイプを選択、宿泊情報を取得、室料を計算、及びホテルの部屋を予約が、全ての変化が生じる変化単位である図面25及び図面99Bを参照して更に詳細に説明されている)。

【0364】

[00533] 全てのソリューションはエージェント固有である。(同じことが、年齢、性別、及びID証明が顧客に固有である図面98Aを参照して更に詳細に説明されている)。

【0365】

[00534] 換言すれば、エージェントが不在の場合、「ソリューション」又は「値」の概念はその意味を失う。

10

【0366】

[00535] あらゆる変化単位には1つ又は複数のエージェントが存在する。(同じことが、機械エージェントが予約ID及び名前を生成し、日数及び稼働状況は人間エージェントにより入力される図面99Bを参照して更に詳細に説明されている)。

【0367】

[00536] 制御される変化の触媒として機能するのはエージェント - CESの部分である。(同じことが、機械エージェントが予約ID及び名前を生成し、日数及び稼働状況は人間エージェントにより入力される図面99Bを参照して更に詳細に説明されている)。

【0368】

20

[00537] ソリューション設計者は、最適化の原理に基づいて変化単位を選ぶ。(同じことが、GSI:ホテルの部屋を予約が5つのLSIを使用して構築され、理論にあるように、同じことは無限状態変化で表現することができる図面26及び図面98Bを参照して更に詳細に説明されている)。

【0369】

[00538] 「変化単位」が大きすぎる場合、変動状態で変化単位に参加している独立実体及び属性は、指数的に増大し、システムを複雑にしすぎる。(同じことが、図面26を参照して更に詳細に説明されている)。

【0370】

[00539] 変化単位が小さすぎる場合、変化単位が増殖することになり、手に負えない冗長性がシステムに忍び込む。(同じことが、図面26を参照して更に詳細に説明されている)。

30

【0371】

[00540] NSLでは、全ての変化単位は、ローカル意図言明(LSI)又はグローバル意図言明(GSI)の何れか一方により表される。(同じことが、GSI:ホテルの部屋を予約が5つのLSIを使用して構築された図面25及び図面98Bを参照して更に詳細に説明されている)。

【0372】

[00541] 意図言明は、自然言語での「規範的言明」に似る - 記述的言明又は疑問的言明とは対照的に。

40

【0373】

[00542] 全ての潜在的な実体状態組合せによりバックアップされない(イベントが生じない場合)全ての意図言明は、「静的意図言明」として見られるべきである。(同じことが、論理想に存在する図面32及び図面98Aを参照して更に詳細に説明されている)。

【0374】

[00543] 換言すれば、静的意図言明は定数であり、作用されない単なる意図のままである。(同じことが、論理層に存在し、作用されない図面32及び図面98Aを参照して更に詳細に説明されている)。

【0375】

[00544] それとは対照的に、動的意図言明は、イベント生じた場合、全ての潜在的な

50

実体状態組合せによりバックアップされる。(同じことが、全てのイベントが生じ、潜在が現実に変換された図面 3 3 及び図面 9 8 B を参照して更に詳細に説明されている)。

【0376】

[00545] 換言すれば、動的意図言明は、作用される言明である。(同じことが、全てのイベントが生じ、潜在が現実に変換された図面 3 3 及び図面 9 8 B を参照して更に詳細に説明されている)。

【0377】

[00546] 静的意図言明は、誰が読んでいるか又は世界で何が生じているかに関係なく、単語がいかなる変化も受けない規範的言明で一杯の本と比較することができる。(同じことが、図面 3 2 及び図面 9 8 A を参照して更に詳細に説明されている)。

10

【0378】

[00547] 動的意図言明は、誰が読んでいるか及び世界で何が生じているかに基づいて単語が単語自体を動的に調整する規範的言明で一杯の本と比較することができる。(同じことが、図面 3 3 及び図面 9 8 B を参照して更に詳細に説明されている)。

【0379】

[00548] ソリューション中の全ての変化はエージェントにより制御され、全ての変化にはエージェントの意図が先行するため、意図言明は常に、任意のトリガー C E S 中に独立実体組合せとして存在する。(同じことが、図面 3 7、図面 9 8 A、及び図面 9 8 B を参照して更に詳細に説明されている)。

【0380】

20

[00549] 意図言明の存在は、黙示されることもあれば、明示的に述べられることもあるが、それらの存在は必然的である。

【0381】

[00550] 意図言明は、実体状態組合せの存在をトリガーするための必須条件ではあるが、十分条件ではない。(同じことが、ホテル予約エージェントの存在を指定するものは何もない図面 3 7 及び図面 9 9 A を参照して更に詳細に説明されている)。

【0382】

[00551] S I はエージェントの意図であり、トリガー C E S はそれらの意図を履行する。(同じことが、最後の属性である料金の捕捉が、ホテルの部屋を予約する意図を履行するトリガー C E S になる図面 3 7 及び図面 9 8 B を参照して更に詳細に説明されている)。

30

【0383】

[00552] 独立実体組合せとしての S I は、人間エージェントへのインターフェースとして機能する主目的に適う。これは、機能的意義を有する必要がない動詞等の記述語を含むことができる。(同じことが、取得、選択、計算、及び予約が、情報値としてのみ機能し、機能に影響しない動詞である図面 2 5、図面 9 8 A、及び図面 9 8 B を参照して更に詳細に説明されている)。

【0384】

[00553] それと共に、独立実体の残りはトリガー C E S の獲得に寄与する。(同じことが、顧客及びその属性がトリガー C E S の履行に役立つ図面 3 7 及び図面 9 8 A を参照して更に詳細に説明されている)。

40

【0385】

[00554] L S I の集まりは G S I に繋がる。(同じことが、一続きに接続された 5 つの L S I の集まりが、ホテルの部屋を予約 G S I の履行に繋がる図面 3 2 及び図面 9 8 A) を参照して更に詳細に説明されている)。

【0386】

[00555] 実体 - 状態組合せ (C E S) は、異なるレベル - 上述したような分化の階層 - における実体の組合せを通して達成される。

【0387】

[00556] 第 1 のレベルの C E S は、種々のレベルにおける独立実体と属性 (依存実体

50

)との組合せから生じる。(同じことが、顧客が、名前、年齢、性別、及びID証明という添付された属性を有する図面98Aを参照して更に詳細に説明されている)。

【0388】

[00557] 第2のレベルのCESは、意図言明内の2つ以上の「第1のレベルのCES」の組合せから生じる。(同じことが、顧客及びホテルは組合せられた独立実体を有する図面98Aを参照して更に詳細に説明されている)。

【0389】

[00558] 第3のレベルのCESは、グローバル意図言明(段落に等しい)内のローカル意図言明(文章に等しい)の組合せから生じる。(同じことが、LSIがCESとして組合せられる図面98Aを参照して更に詳細に説明されている)。

10

【0390】

[00559] 原理上、第4のレベル及びそれよりも上の多くのレベルは、段落を超えるより高レベルの分化に繋がるグローバル意図言明の組合せから生じることができる。

【0391】

[00560] SIは、独立実体として扱われるべきであり、それに関連付けられた属性は、自然言語での「副詞」に等しい。

【0392】

[00561] 他の各独立実体に関連付けられた属性は、自然言語での「形容詞」と等しく扱うことができる。(同じことが、Wi-Fi、レート、及び客室ステータスが形容詞である図面52及び図面97Aを参照して更に詳細に説明されている)。

20

【0393】

[00562] 実体が組合せられる場合、それらの組合せ状態はそれら自体の新興特性を有することになる。これは、それら自体の特性を有し、結合されると、「水」の特性を取得する酸素原子及び水素原子のようである。

【0394】

[00563] 全ての実体は潜在又は現実のバイナリ状態で存在する - 「潜在実体」及び「現実実体」。(同じことが、図面27を参照して更に詳細に説明されている)。

【0395】

[00564] ソリューション設計者は、現実世界から潜在実体を選択し、ソリューションエコシステムの状況において関連がない非潜在実体を除外する。

30

【0396】

[00565] しかしながら、明示的ではないが、ソリューションエコシステムに参加している黙示実体の可能性がある。(同じことが、ホテル予約エージェントが黙示エージェントである図面10及び図面99Bを参照して更に詳細に説明されている)。

【0397】

[00566] 黙示実体とは、所与として解釈され、不確実性が添付されず、生み出すいかなる結果効果も有さない実体である。(同じことが、ホテル予約エージェントが黙示実体である図面10及び図面99Bを参照して更に詳細に説明されている)。

【0398】

[00567] 同様に、ソリューション設計に寄与効果を持たないため無理されるエコシステム内の実体が存在することができる。

40

【0399】

[00568] バイナリ状態での実体の存在は、NSL設計原理の中心である。(同じことが、図面27を参照して更に詳細に説明されている)。

【0400】

[00569] イベント発生時、実体のバイナリ状態が変化するとき、状態遷移が生じる。(同じことが、60としての年齢の値、Mとしての性別の値、及びアドハーとしてのID証明の値がCESの状態を変えている図面30及び図面98Aを参照して更に詳細に説明されている)。

【0401】

50

[00570] CESのサイズは、参加する独立実体及び属性実体に比例して増大する。(同じことが、図面61及び図面99Bを参照して更に詳細に説明されている。顧客名、客室空き状況、稼働状況、及び滞在日数の形態の4つの変数がある。ソリューション設計者はまた、追加の変数として喫煙/非喫煙の好みを捕捉したいことがある)。

【0402】

[00571] NSL設計では、全てのCESは潜在及び現実という離散バイナリ状態でのみ存在することが求められる。実体と一緒に組合せられると、実体の個々のステータスは無視され、組合せステータスのみが重要になる。(同じことが、ホテルの部屋の予約が、全てのLSI、独立実体、及びそれらの属性が全て一緒に組合せられたもので構成されたCESである図面37及び図面99Bを参照して更に詳細に説明されている)。

10

【0403】

[00572] 「イベント」は、生じる場合、関連付けられた空間スタンプ及び時間スタンプを有する。すなわち、イベントは常に空間及び時間を参照する。例：特定のイベントが昨日インド又は米国で発生したか？それらの一方のみが真である。(同じことが、年齢：60、性別：男性、及びID証明：アドラーという属性が、空間及び時間に到着する情報実体である図面98Aを参照して更に詳細に説明されている)。

【0404】

[00573] イベントは、独立実体又はその属性の一方が状態を潜在から現実に及びこの逆に変えて、意図言明内のCESの1つを生じさせる(変化単位)場合、生じたとされるものである。(同じことが、年齢：60、性別：男性、及びID証明：アドラーという属性が、空間及び時間に到着する情報実体である図面37及び図面98Aを参照して更に詳細に説明されている)。

20

【0405】

[00574] 全てのイベントは、イベントが属するSIを含め、1つ又は複数のSIにおけるトリガーCESにより生じる。(同じことが、顧客名の捕捉が、続くLSIにおけるレート及び客室空き状況の捕捉をトリガーする図面37及び図面99Bを参照して更に詳細に説明されている)。

【0406】

[00575] 同様に、トリガーCESは、トリガーCESが属するSI又は別のSI内で1つ又は複数のイベントを生じさせる。これらはイベントの効果と呼ばれる。(同じことが、客室タイプの選択が、表示されるレートを自動的に決める図面99Bを参照して更に詳細に説明されている)。

30

【0407】

[00576] トリガーCESが生じる都度、トリガーCESは、1つ又は複数のイベントを生じさせる前、「経過時間」と呼ばれる特定量の時間を消費する。(同じことが、図面60を参照して更に詳細に説明されている)。

【0408】

[00577] 意図言明内のイベントの空間基準点は、トリガーCESが生じる空間である。(同じことが、図面37を参照して更に詳細に説明されている)。

【0409】

[00578] イベントは、独立実体又は属性レベルで生じる場合、意図言明内の組合せ実体-状態の変化に影響する。(同じことが、名前及び客室空き状況の属性レベルでの捕捉が、CES：顧客情報を取得の変化に影響する図面99Bを参照して更に詳細に説明されている)。

40

【0410】

[00579] 各イベントに関する種々のレベルの不確実性があり得る。

【0411】

[00580] 確率理論を適用して、そのようなイベントに値を割り当てることができる。

【0412】

[00581] 意図言明内の「可能な状態」の数は 2^n であり、ここで、「n」は「バイナ

50

り実体の数」である - すなわち、6つのバイナリ変数又は実体がある場合、64個のCESが存在することになる。(同じことが、図面61を参照して更に詳細に説明されている)。

【0413】

【00582】 全ての独立実体及びそれらの属性が現実状態である状態は「トリガーCES」であり、他の全ての状態は非トリガーCESである。先の例では、64状態はトリガーCESであり、その他の63個は非トリガーCESである。(同じことが、図面37を参照して更に詳細に説明されている)。

【0414】

【00583】 「トリガーCES」は、他の意図言明(同じ段落若しくは他の段落内)又はそれ自体の意図言明においてイベントを生じさせる。(同じことが、図面37を参照して更に詳細に説明されている)。

10

【0415】

【00584】 非トリガーCESは、いかなる結果イベント又は変化も生じさせない。(同じことが、図面37を参照して更に詳細に説明されている)。

【0416】

【00585】 意図言明はニューロンのものであり、トリガーCESは活動電位が蓄積されたときのニューロンの発射のものである。(同じことが、図面37を参照して更に詳細に説明されている)。

【0417】

【00586】 他のSIに影響する意図言明(SI)は関連すると見なされ、CESの圏いに入る。(同じことが、関連するSIと一緒にされる図面95を参照して更に詳細に説明されている)。

20

【0418】

【00587】 プロセスバリアの打ち破りは、関連するSIが新しい拡張CESを生じさせる場合、生じる。

【0419】

【00588】 他のSI及びそれらの構成要素がトリガーCES関係に基づいて組合せられる場合、CESは拡張CESになる。(同じことが、図面98Bを参照して更に詳細に説明されている: a) 顧客情報を取得、b) 顧客情報を取得、客室タイプを選択、c) 顧客情報を取得、客室タイプを選択、客室宿泊情報を取得、..... d) 言明a、b、cは関係に基づいて拡張されている)。

30

【0420】

【00589】 あらゆるイベントは何れかのトリガーCESによって生じる。(同じことが、図面37を参照して更に詳細に説明されている)。

【0421】

【00590】 トリガーCESは原因であり、その結果生じるイベントは結果である。(同じことが、図面37を参照して更に詳細に説明されている)。

【0422】

【00591】 原因及び結果が、拡張CESと呼ばれるCESの形態の分化実体として一緒に組合せられる場合、プロセスは情報に変換される。(同じことが、プロセス自然言語の形態の情報に変換されている図面103を参照して更に詳細に説明されている)。

40

【0423】

【00592】 新しいパラダイム - 拡張CESを含め、独立実体及びCESのみが存在する - では、情報として表現される実体間の分化のみが存在する。

【0424】

【00593】 意味を捕捉するに当たり情報を支配する同じ「分化原理」は、ソリューション又は価値の作成も支配する。(同じことが、プロセスが自然言語の形態の情報に変換される図面59及び図面103を参照して更に詳細に説明されている)。

【0425】

50

【00594】ここでは、ソリューション状態さえも情報のドメイン、ひいてはサーチエンジンのドメインに入る。（同じことが、プロセス自然言語の形態の情報に変換されている図面103を参照して更に詳細に説明されている）。

【0426】

【00595】「エージェント」との関連では、エージェントは大半の他の実体に依存しないが、実体状態組合せによっては好ましいもの及び好ましくないものがある。

【0427】

【00596】拡張CESの場合、グローバルSIに繋がる接続されたSIは、同じ段落に属するものとして扱われる。（同じことが、一続きに接続された5つのLSIの集まりは、ホテルの部屋を予約GSIの履行に繋がる図面63及び図面98Aを参照して更に詳細に説明されている）。

10

【0428】

【00597】全てのSIは、グローバルSIを含め、エージェント固有である。（同じことが、5つ全てのLSIがホテル予約エージェントに固有である図面25及び図面98A）を参照して更に詳細に説明されている）。

【0429】

【00598】段落は、グローバルSIの履行に繋がるローカルSIの集まりとして定義される。（同じことが、一続きに接続された5つのLSIの集まりは、ホテルの部屋を予約GSIの履行に繋がる図面25及び図面98A）を参照して更に詳細に説明されている）。

【0430】

20

【00599】各段落は、1つのみのグローバル意図言明を履行する。（同じことが、一続きに接続された5つのLSIの集まりは、ホテルの部屋を予約GSIの履行に繋がる図面25及び図面98A）を参照して更に詳細に説明されている）。

【0431】

【00600】2つ以上のグローバルSIがある場合、各GSIは異なる段落で対応されるべきである。（同じことが、あらゆる段落が1つのGSIをサポートする図面25及び図面107Hを参照して更に詳細に説明されている）。

【0432】

【00601】NSLでは、グローバルSIを履行するには、グローバルSIの段落に属する全ての独立実体及びそれらの属性が現実状態であるべきことが要求される。（同じことが、全てのイベントが既に生じている図面98Bを参照して更に詳細に説明されている）。

30

【0433】

【00602】NSLが依拠する1つの基本的な基準は、任意のSIを履行するには、その全ての独立実体及びそれらの属性が現実状態であるべきであるというものである。（同じことが、全てのイベントが既に生じている図面98Bを参照して更に詳細に説明されている）。

【0434】

【00603】この原理は、ローカルSI（文章レベル）であれグローバルSI（段落レベル）であれ関係なく等しく適用される。

【0435】

40

【00604】NSLは、「空間及び時間」と同じステータスを与える基本属性として「ソリューション論理」を見なす。

【0436】

【00605】この論理は、エージェントにより定義されエージェントにより作用される。（同じことが、論理がホテル予約エージェントにより作用される図面98Aを参照して更に詳細に説明されている）。

【0437】

【00606】全ての分化経路は、ソリューション設計者により潜在段落として定義される。（同じことが、ホテルの部屋を予約がソリューション設計者により定められる図面98Aに示されている）。

50

【 0 4 3 8 】

【00607】 エージェントが新しい願望を有する場合、それは新しいグローバル意図言明としての資格を満たす。(同じことが、割り当てられた教科教師が割合成績を計算が、3つの異なる段落で対応されている図面107hに示されている)。

【 0 4 3 9 】

【00608】 その意図言明は、設計により、そのサブセットとして潜在段落の1つに属する必要がある。(同じことが、NSLソリューション構造体が示されている図面95に示されている)。

【 0 4 4 0 】

【00609】 グローバルSIが生まれると、そのグローバルSIを履行するために分化経路に定義された潜在段落に即座にラッチされる。(同じことが、潜在段落にラッチされたGSI図面98Aに示されている)。

10

【 0 4 4 1 】

【00610】 互いに関連する全てのグローバルSIは、「関連段落」として表現することができる。(同じことが、3つの関連段落が表されている図面107hに示されている)。

【 0 4 4 2 】

【00611】 関連段落は、同じグローバルSIに仕えている段落であってもよく(従来、同じソリューションに繋がる異なるプロセス)、又はグローバルSIに対してわずかなバリエーションを有する段落であってもよい。

【 0 4 4 3 】

【00612】 関連段落の1つは、グローバル意図言明を終わらせるものであることができる。(同じことが、ホテルの部屋を予約が、段落を終わらせるGSIである図面99Bに示されている)。

20

【 0 4 4 4 】

【00613】 複数の関連段落にわたり共通するCESは、共有実体と呼ばれる。共有は、2つ以上のSIに仕える潜在性を有する実体に関連する。(同じことが、割り当て教科教師が生徒の成績を計算が共有実体である図面56及び図面107hを参照して更に詳細に説明されている)。

【 0 4 4 5 】

【00614】 複数の段落が個々にグローバル意図言明の履行に繋がることのできる場合がある。NSLは、判決方向を通して代替段落間から行うことで選択を解決する。(同じことが、図面49を参照して更に詳細に説明されている)。

30

【 0 4 4 6 】

【00615】 判決方法は、物事が、言明意図の履行に残っている1つのみの段落に微調整されるように、追加の独立実体(制約として機能する)を既存の組合せ実体に追加することに頼る。(同じことが、図面49を参照して更に詳細に説明されている)。

【 0 4 4 7 】

【00616】 代替的には、グローバル意図言明を履行することができる段落がない場合、独立実体の選択的削除(制約の除去として機能する)により、意図言明を出現させることができる。(同じことが、図面49を参照して更に詳細に説明されている)。

40

【 0 4 4 8 】

【00617】 影響が原因イベントにより感じられる前、あらゆるトリガーCESに関連付けられた経過時間がある。(同じことが、図面60を参照して更に詳細に説明されている)。

【 0 4 4 9 】

【00618】 経過時間は長くても又は短くてもよいが、常に例外なく存在する。一般的には、機械エージェントの関連する経過時間は短い - 数分の一秒でカウントされる。人間エージェントの関連する経過時間は長い - 数秒から数時間又は数日まで様々である。(同じことが、図面60を参照して更に詳細に説明されている)。

【 0 4 5 0 】

50

【00619】 NSLは、科学により確立される「世界の仕組み」原理に基づく。

【0451】

【00620】 NSLは、人間エージェントの挙動及び機能を模倣する。

【0452】

【00621】 3層の定義された人間エージェント機能がある。(同じことが、図面44を参照して更に詳細に説明されている)。

【0453】

【00622】 第1の層は、トリガー条件の履行に当たり、SIでイベントを引き起こす物理機能である。(同じことが、名前、客室タイプ、及び稼働状況の入力は物理機能である図面44及び図面99Aを参照して更に詳細に説明されている)。

10

【0454】

【00623】 第2の層は、人間の感覚をなくし、エコシステムの状態を捕捉して、エージェントに常に情報が与えられるようにする情報機能である。(同じことが、何をしたいですか、予約した部屋をキャンセル、サービスを利用が情報実体として機能する図面44及び図面99Aを参照して更に詳細に説明されている)。

【0455】

【00624】 第3の層は、各イベントの発生時、環境を再評価し、フィードバックを物理機能に提供するマインド機能である。(同じことが、図面44を参照して更に詳細に説明されている)。

【0456】

20

【00625】 NSLは、必要に応じて存在する3つ全ての層の段落を提供する。(同じことが、図面44を参照して更に詳細に説明されている)。

【0457】

【00626】 デフォルト層は物理機能層である。(同じことが、図面44を参照して更に詳細に説明されている)。

【0458】

【00627】 情報機能層における実体は、情報潜在性を保持することに成功していた。すなわち、分析を支援し、測定の結果を果し、又は将来の設計の状況で使用することができた。(同じことが、図面44を参照して更に詳細に説明されている)。

【0459】

30

【00628】 マインド機能層は、物理層をガイドし、トリガー条件の確認に役立つ。イベント発生時、関連する意図言明中の現在及び将来のCESを再査定(予測)して更新し、トリガープロセスをガイドする。(同じことが、図面44を参照して更に詳細に説明されている)。

【0460】

【00629】 マインド機能層は、特に、予想されたトリガーCES、トリガーに関連付けられた経過時間、及びトリガーCESに関連付けられた確率に頼る。(同じことが、図面44を参照して更に詳細に説明されている)。

【0461】

【00630】 高度計画及び最適化(Advanced Planning and Optimization)(APO)原理は自然に、NSLにおける「マインド機能層」の使用を通して動作可能にすることができる。(同じことが、図面44を参照して更に詳細に説明されている)。

40

【0462】

【00631】 NSLは、他の内部又は外部ソリューションにアクセスする過程において、これもまた同様の分化原理に頼るアプリケーションプログラミングインターフェース(Application Programming Interfaces)(API)に等しく適用される。(同じことが、図面65を参照して更に詳細に説明されている)。

【0463】

【00632】 NSLの最も重要で飛躍的な側面の1つは、構造体が自然言語のようであるのみならず、いかなる自然言語にも依存しないものであることである。(同じことが、

50

図面 6 8 を参照して更に詳細に説明されている)。

【 0 4 6 4 】

[00633] N S L は分化原理に基づくため、集合分化方法に従っている限り、システムは、使用される自然言語に関係なく、全く同じように認識し動作する。(同じことが、図面 6 8 を参照して更に詳細に説明されている)。

【 0 4 6 5 】

[00634] N S L ソリューション言語は英語に固有ではなく、任意の既知の自然言語で等しく容易に書くことができる。(同じことが、図面 6 8 を参照して更に詳細に説明されている)。

【 0 4 6 6 】

[00635] 必要なのは、N S L 技術フレームワークが、抽出レベルに適切なように実体と対処し、使用される言語に関係なく同じ結果を生成するように分化レベルに従うことだけである。(同じことが、図面 6 8 を参照して更に詳細に説明されている)。

【 0 4 6 7 】

[00636] 自然言語は、人間エージェント間の情報交換の土台を形成する。(同じことが、図面 6 8 を参照して更に詳細に説明されている)。

【 0 4 6 8 】

[00637] 自然言語はまた、ロシア語しか話せない人が英語しか話せない人とコミュニケーションすることができないようなコードの形態でもある。(同じことが、図面 5 2 を参照して更に詳細に説明されている)。

【 0 4 6 9 】

[00638] 同様に、コンピュータとの通信は、長い間、専用コードを含み、タスクの実行にプログラミング言語の訓練を受けた人々が必要とされる種々のプログラミング言語を通して行われてきた。(同じことが、図面 5 2 を参照して更に詳細に説明されている)。

【 0 4 7 0 】

[00639] N S L は、自然言語のような構造体の使用を通して、ソリューション論理をコンピュータに伝える方法を導入することにより新しいパラダイムを作成する。(同じことが、図面 6 6 を参照して更に詳細に説明されている)。

【 0 4 7 1 】

[00640] 自然言語ソリューションは、基本的な技術フレームワークの上に革新的な構造及び方法を使用して、ソリューション論理を機械エージェント(コンピュータ)に自然言語のように効果的に伝える。(同じことが、図面 6 6 を参照して更に詳細に説明されている)。

【 0 4 7 2 】

[00641] N S L での実体関係は、自然言語での発話の部分と同様である。(同じことが、図面 6 7 を参照して更に詳細に説明されている)。

【 0 4 7 3 】

[00642] これは、N S L が媒体に依存せずにソリューション論理をコンピュータに伝達する柔軟性を与える。(同じことが図面 7 2 に示されている)。

【 0 4 7 4 】

[00643] N S L は、文字及び画像認識技法の進歩を利用し、コンピュータに接続されたビデオカメラを用いて、ソリューション論理を手書き形態でコンピュータに直接伝達することができる。(同じことが図面 7 2 に示されている)。

【 0 4 7 5 】

[00644] 同様に、音声認識技法の進歩を利用して、N S L ソリューション論理を音声記録し、コンピュータに直接伝達することができる。(同じことが図面 7 2 に示されている)。

【 0 4 7 6 】

[00645] N S L は手話にも拡張可能である。(同じことが図面 7 2 に示されている)。

【 0 4 7 7 】

10

20

30

40

50

【00646】 NSLはまた、ソリューションの性能を捕捉する動画に基づいてソリューション論理を抽出する能力も有する。(同じことが図面72に示されている)。

【0478】

【00647】 動画に存在する現実実体及び情報実体は、システムにより認識される。(同じことが図面72に示されている)。

【0479】

【00648】 動画で観測される変化単位は「拡張CES」に変換され、それにより、プロセス又はソリューション論理を捕捉する。(同じことが図面72に示されている)。

【0480】

【00649】 NSLは全てのソリューション論理を実体及び実体の関係に縮小するため、任意の識別された実体を個々に又はまとめて暗号的にラップする柔軟性をもって任意の所望のレベルまで「システムセキュリティ」に対処するのに最も自然に適する。

10

【0481】

【00650】 拡大解釈すれば、信頼機関をなくしながら高いセキュリティを探し求めるソリューションのようなブロックチェーンの適用を容易に達成することができる。

【0482】

【00651】 NSLは、特徴的な容易さで「測定フレームワーク」を作成する能力を有する。(同じことが、図面50を参照して更に詳細に説明されている)。

【0483】

【00652】 NSLは、任意の実体又はエコシステムの状態を客観的に特定するのに正しい種類の詳細及び粒度が達成されるように、分化を要求される限度まで推し進める。(同じことが、図面34を参照して更に詳細に説明されている)。

20

【0484】

【00653】 その後、従来、規範と呼ばれる現実の状態の周囲の価値判断を極めて自然に捕捉し、緑及び赤等の色を生成して、選択された物体の状態を人間エージェントに通知することもできる。(同じことが、図面50を参照して更に詳細に説明されている)。

【0485】

【00654】 NSLは、必要に応じて任意の所望の視座においてCESを極めて自然に捕捉する。(同じことが、図面35を参照して更に詳細に説明されている)。

【0486】

30

【00655】 NSLは、エコシステム内の任意のSIの性能ステータスを容易に管理することができる。

【0487】

【00656】 NSLは、尽力及びリソース要件推知を動的に行うことができる。

【0488】

【00657】 全てのSIは、人間エージェントにより所有され駆動されるため、任意の人間エージェントの「理解レベル」を容易に査定する機能で構築することができる。

【0489】

【00658】 各変化単位、すなわち各SIは、PSAサイクルと呼ばれる絶えず存在する「ソリューション」又は「価値」サイクル(ソリューション又は価値の全ライフサイクル)の影響下にある。(同じことが、図面67を参照して更に詳細に説明されている)。

40

【0490】

【00659】 「P」は、システム及びシステム内のエージェントにより捕捉されているSI(変化単位)のレベルに存在する「可能性」を表す。(同じことが、ホテル予約エージェントが、部屋の予約、交通機関若しくはランドリーの予約、又は部屋のキャンセルの何れかの全ての可能性を捕捉している図面99Aを参照して更に詳細に説明されている)。

【0491】

【00660】 「S」は、システム及びエージェントが、利用可能な可能性から正しい種類の選択を行う能力を有する「選択」を表す。(同じことが、ホテル予約エージェントが、顧客が頭の中で選んだ部屋の予約、交通機関若しくはランドリーの予約又は部屋のキャン

50

セルの何れかの全ての可能性を捕捉している図面 9 9 A を参照して更に詳細に説明されている)。

【0492】

[00661] 「A」は、システム及びエージェントが P S A サイクルを完了するのに正しい機能を事項する能力を有する「行動」を表す。(同じことが、ホテル予約エージェントが、顧客が頭の中で選んだ部屋の予約、交通機関若しくはランドリーの予約又は部屋のキャンセルの何れかの全ての可能性を入力した図面 9 9 A を参照して更に詳細に説明されている)。

【0493】

[00662] N S L は、ソリューションエコシステムの全てのレベルに関わるソリューションサイクルのフラクタルのような挙動及びソリューションエコシステム内部の固有の対称性を認識し利用する。(同じことが、図面 6 7 を参照して更に詳細に説明されている)。

10

【0494】

[00663] N S L はソリューション設計に自然言語ベースの手法をとるため、ユーザインターフェースとデータベースとの間の実体差を効果的になくす。(同じことが、図面 6 6 及び図面 9 5 を参照して更に詳細に説明されている)。

【0495】

[00664] 全ての実体は、何れの間人エージェントに関してもユーザインターフェースレベルに存在する。(同じことが、図面 5 1 を参照して更に詳細に説明されている)。

【0496】

20

[00665] 「機械エージェント」機能さえも、システム管理者を含め、いずれのエージェントにもトランスペアレントである。(同じことが、予約 I D、機械エージェント機能がエージェント及びシステム管理者にトランスペアレントである図面 5 1 及び図面 9 8 B を参照して更に詳細に説明されている)。

【0497】

[00666] N S L は、ソリューションエコシステムに存在する全ての実体についてのブラックボックスの存在をなくす。

【0498】

[00667] 実体へのアクセスは、純粹にアクセス特権 - 情報権及び決定権 - に基づいて全ての間人エージェントに提供される。(同じことが、図面 4 5 及び図面 4 6 を参照して更に詳細に説明されている)。

30

【0499】

[00668] N S L 技術フレームワークは、全ての関連する分化実体の特定に当たりソリューション設計者及びユーザを助ける。(同じことが、図面 6 9 を参照して更に詳細に説明されている)。

【0500】

[00669] 次に、それらの分化実体のそれぞれには、技術フレームワークにより支援される情報権及び決定権がタグ付けされる。(同じことが、図面 6 9 を参照して更に詳細に説明されている)。

【0501】

40

[00670] エコシステム内の各 C E S は全ての視座で明確に識別されるため、大半の場合、属性値の適切な割り当てが系統的に駆動される。(同じことが、属性及び属性の値の割り当てが系統的に駆動される図面 6 9 及び図面 9 8 A を参照して更に詳細に説明されている)。

【0502】

[00671] より高い視座における事柄に属性値が割り当てられる場合、技術フレームワークは、いかなる手動介入もなく自動的に、同じ分化分岐下に入る全ての実体と同じ値を割り当てる能力を有する。(同じことが、上位集合生徒下のあらゆるユーザに同じ属性及び値が自動的に与えられる図面 6 9 及び図面 1 0 7 I を参照して更に詳細に説明されている)。

50

【0503】

[00672] NSLは、定められた規格の範囲内でステークホルダが好むように実体を編成する大きな柔軟性をステークホルダに提供する。(同じことが、図面69を参照して更に詳細に説明されている)。

【0504】

[00673] 関連する任意の画面上の任意の場所に実体を移動させることができるようにステークホルダに提供される方法及びツールがある。(同じことが、図面69を参照して更に詳細に説明されている)。

【0505】

[00674] エージェント毎の実体の所有権は、全てのSI及びそれらのSIに参加しているCESが、それらのSIを駆動するエージェントに属することになるという単純な原理によって決まる。(同じことが、図面43を参照して更に詳細に説明されている)。

10

【0506】

[00675] 2つの条件は偶発的にそのような所有権を決める - それらの実体及び実体に提供される特権と物理的に相互作用する能力。(同じことが、図面43、図面45、及び図面46を参照して更に詳細に説明されている)。

【0507】

[00676] 所有権はまた、SIで行動を実行するように参加エージェントに指示する権利を有するエージェントに拡張される。(同じことが、図面43、図面45、及び図面46を参照して更に詳細に説明されている)。

20

【0508】

[00677] ビジネスソリューションはまた、各イベントにおける価値を直接又は間接的に交換する能力も提供する。

【0509】

[00678] キャッシュ等における価値のこの交換は、金融システムで取引の記録を駆動するのと同じ原理により駆動される。

【0510】

[00679] 換言すれば、SIが、1つ又は複数の他のSIにおけるCESによりトリガーされるイベントを経験する都度、又はトリガーCESが1つ又は複数の他のSIにおけるイベントを生じさせる場合、価値は交換される。(同じことが、全体として2つのLSIのトリガーに応じる顧客名、客室空き状況、稼働状況、及び日数の形態の4つの変数がある図面62及び図面99Bに示されている。したがって、ソリューション設計者が情報を価値として定量化するのを助ける)。

30

【0511】

[00680] 到着するイベントによりNSLで表現される因果原理は、1つ又は複数の他のSI中の1つ又は複数のトリガーCESによって生じる。(同じことが、全体として2つのLSIのトリガーに応じる顧客名、客室空き状況、稼働状況、及び日数の形態の4つの変数がある図面99Bに示されている。したがって、ソリューション設計者が情報を価値として定量化するのを助ける)。

【0512】

[00681] 同様に、トリガーCESがSI内で生じる場合、1つ又は複数のSIにおいて1つ又は複数のイベントを生じさせる同じことが。(同じことが、全体として2つのLSIのトリガーに応じる顧客名、客室空き状況、稼働状況、及び日数の形態の4つの変数がある図99Bに示されている。したがって、ソリューション設計者が情報を価値として定量化するのを助ける)。

40

【0513】

[00682] 標準原理に従ってNSLでソリューションエコシステムにおけるCESとイベントとの間に相関も容易に確立される。(同じことが、室料が、日数及び稼働状況属性での実体到着に等しいCESである図面97Bに示されている)。

【0514】

50

【00683】 主語及びシステムとしての自然言語処理（NLP）は、自然言語のコンピュータ理解及びアプリケーションの開発に対処する。

【0515】

【00684】 自然言語原理はNLP及びNSLの両方に共通であるが、NSLのフォーカスは、関わる人間エージェントによる使用のし易さのために、自然言語のような構造体を用いてソリューション論理をコンピュータに直接付与することである。

【0516】

【00685】 NLPとNSLとの間のこの自然な親和性は、NSLが、現在状態と比較してはるかに改良されて、NLPシステム及び方法を作成するのをはるかに容易にする。（同じことが、図面52を参照して更に詳細に説明されている）。

10

【0517】

【00686】 NSLに関連付けられたユーザフレンドリさは、NSLの状況で「ゲーミング技法」を用いて次のレベルに運ぶことができる。

【0518】

【00687】 ゲーミング技法は、アニメーションを通してよりよい視覚的效果を提供するために、ユーザをトレーニングするため、ソリューションをテストするため、ユーザ間により高いインタラクティブ性を生じさせる等のために展開することができる。

【0519】

【00688】 NSLでのユーザインターフェース（UI）はCES中心であり、大きな多様性があるステークホルダのニーズを満たすようにUIを構成するより大きな柔軟性を与えるように設計される。（同じことが、図面51を参照して更に詳細に説明されている）。

20

【0520】

【00689】 NSLの状況でのこれらのユーザインターフェースは、ステークホルダエンゲージメントセンタ（SEC）と呼ばれる。（同じことが、図面51を参照して更に詳細に説明されている）。

【0521】

【00690】 SECは、必要に応じて、必要な場合に各ステークホルダへのCESアクセスに対応しカスタマイズするように、それ自体を自動的に調整する。（同じことが、図面51を参照して更に詳細に説明されている）。

【0522】

30

【00691】 NSLはエージェントの情報権及び決定権に気付いているため、それ自体を自動較正する能力を取得するはステークホルダである。（同じことが、図面51を参照して更に詳細に説明されている）。

【0523】

【00692】 ユーザフレンドリさをビジネスソリューションにおける更に別のレベルに運ぶために、SECはまた、相互作用の性質に基づいてステークホルダへの異なるレベルのアクセスよりサブ構造も提供する。（同じことが、図面51を参照して更に詳細に説明されている）。

【0524】

【00693】 これらのレベルは私のトランザクション、私自身、私のチーム、及び私の組織と呼ばれ、ステークホルダの要件のタイプの通りにCESを包含する。（同じことが、図面51を参照して更に詳細に説明されている）。

40

【0525】

【00694】 動的ミッションコントロールセンタ（MCC）は、ステークホルダがタスクをよりよく実行するために重要なCESへの即時アクセスを提供することにフォーカスされるSECの自然拡張である。（同じことが、図面71を参照して更に詳細に説明されている）。

【0526】

【00695】 動的MCCは、時間、イベント、ステークホルダの関心事、及びステークホルダのクエリの考慮に基づいてCESアクセスをステークホルダに提供する。（同じこと

50

が、図面 7 1 を参照して更に詳細に説明されている)。

【 0 5 2 7 】

[00696] N S L は、現在のシステムよりもはるかに大きな効率でエコシステムでの不確実性に対処する能力を有する。

【 0 5 2 8 】

[00697] N S L では、あらゆることが C E S 中心であるため、それらに関連付けられた不確実性は、エコシステムにおける変化している C E S に基づいて自然発生的に特定することができる。

【 0 5 2 9 】

[00698] 確率は、分析原理及びビッグデータの使用に基づいて動的に各イベントに割り当てることができる。

10

【 0 5 3 0 】

[00699] 実体及びイベント(実体の挙動)に基づく機械エージェントの自己学習は、ここ数十年がそこらで強固な足場を築いてきた。

【 0 5 3 1 】

[00700] システムとしての N S L は、N S L の全ての実体は個々に及び集合的に、世界の仕組み(及びマインドの仕組み)を模倣して、経験ベースの学習の下地を作るため、そのような機械学習に自然に調和する。

【 0 5 3 2 】

[00701] この自然な拡張は、人工知能と呼ばれる人間知能の模倣であり、N S L は、エコシステムにおいて絶えず変わる状況に機械エージェントが適合し応答する肥沃な土地を提供する。

20

【 0 5 3 3 】

[00702] N S L は、主に規範情報(意図言明)に対処し、偶発的に記述的情報(情報機能層における情報)に対処する自然言語での言明の特殊な場合である。

【 0 5 3 4 】

[00703] N S L の最も劇的な成果の 1 つは、情報理論が「ビット」でカウントされる情報を定量化することができたのと全く同じように、「ソリューション」及び「価値」を定量化するその能力である。(同じことが、全体として 2 つの L S I のトリガーに応じる顧客名、客室空き状況、稼働状況、及び日数の形態の 4 つの変数がある図面 6 1 及び図面 9 9 B を参照して更に詳細に説明されている。したがって、ソリューション設計者が情報を価値として定量化するのを助ける)。

30

【 0 5 3 5 】

[00704] あらゆるソリューションでは、自動的に定量化させるのに所与の数の独立及び属性のバイナリ状態(バイナリイベントの数)が現実状態である必要がある。(同じことが、全体として 2 つの L S I のトリガーに応じる顧客名、客室空き状況、稼働状況、及び日数の形態の 4 つの変数がある図面 6 1 及び図面 9 9 B を参照して更に詳細に説明されている。したがって、ソリューション設計者が情報を価値として定量化するのを助ける)。

【 0 5 3 6 】

[00705] 同様に、あらゆるソリューションでは、意図された履行 C E S を生じさせるためには、所与の数のトリガー C E S が生じる必要がある。(同じことが、図面 9 9 B を参照して説明されている)。

40

【 0 5 3 7 】

[00706] N S L では、アプリケーション論理に関する全ての実体は、自然言語のようなフォーマットではユーザインターフェースレベルで存在し、物事を高度にユーザフレンドリにする。(同じことが、図面 6 6 を参照して更に詳細に説明されている)。

【 0 5 3 8 】

[00707] N S L はユーザをソリューション構築、保守、又は改良に参加者させるため、正しい種類のツール及びコンポーネントを用いてユーザをサポートする必要がある。

【 0 5 3 9 】

50

[00708] NSLライブラリは、バリエーションが「段落」以上までの属性関係から生じる独立実体等の有用なソリューションコンポーネントをストックする。(同じことが、図面55及び図面107Aを参照して更に詳細に説明されている)

【0540】

[00709] これらは、必要に応じて使用することができ、インテリジェントサーチ法又はドラッグアンドドロップメカニズムによりサポートすることができる。(同じことが、名前、モバイル、アドレス、年齢、及び電子メールが、あらゆる生徒に再使用可能な属性として添付されている図面107Iに示されている)。

【0541】

[00710] 一般に、NSLは、ライブラリ内の再使用可能コンポーネントを容易に識別し、バージョン変更が必要とされる場合を除き、アプリケーション開発に関して再使用可能コンポーネントのみを使用することにより、冗長性を最小化する。(同じことが、名前、モバイル、住所、年齢、及び電子メールが、あらゆる生徒に再使用可能な属性として添付されている図面53及び図面107Iを参照して更に詳細に説明されている)。

10

【0542】

[00711] NSLライブラリ内のこれらの再使用可能コンポーネントは、キュレータとして機能する適切なステークホルダのサポート機能に基づいて分類、ソートされる(同じことが、図面107Aを参照して更に詳細に説明されている)。

【0543】

[00712] NSLアプリケーション開発者は、殆どはライブラリからのこれらのコンポーネントにアクセスし、必要に応じて分化したソリューションと一緒に組み合わせることができる。(同じことが、図面55及び図面107Aを参照して更に詳細に説明されている)。

20

【0544】

[00713] プログラミング論理のNSLトランスレータがプログラミングコードをNSLアプリケーション論理に変換する場合、全ての既存のコードは、オープンソースプログラミングコードを含め、ユーザにトランスペアレントになる。

【0545】

[00714] アプリケーション論理はもはやプログラミングコードのブラックボックスに隠れていないため、再使用に有用な全てのソリューションコンポーネントを識別する大きな機会が存在する。(同じことが、図面53を参照して更に詳細に説明されている)。

30

【0546】

[00715] SIはニューロンと比較することができ、あらゆるトリガーCESはニューロンの発射と比較することができる。

【0547】

[00716] 行動潜在性は、ニューロンに閾値レベルまで蓄積され、ニューロンを発射させる。

【0548】

[00717] 同様に、独立実体レベルでのバイナリ実体状態は、イベント発生時に現実状態になり続け、トリガーCESの閾値に達し、他のSIにおける変化を生じさせる。(同じことが、図面99Bを参照して更に詳細に説明されている)。

40

【0549】

[00718] これは、人間の脳がいかに機能するかをNSLがいかに模倣することができるかについて劇的な暗示を有する。

【0550】

[00719] 拡大解釈すれば、人口ニューラルネットワーク(ANN)環境に効果的に対処するその能力は非常に高い。(同じことが、図面64を参照して更に詳細に説明されている)。

【0551】

[00720] NSLは、より構造化されるとともにネットワーク化されたソリューション

50

環境に対処する柔軟なフレームワークを提供する。

【0552】

[00721] その意味では、機構モデル及び生物モデルの両方の場合で適用されるのに極めてよく適する。

【0553】

[00722] NSLにおける可能性 - 選択 - 行動サイクルと、タンパク質形成のために作用される「RNAを生じさせるDNA」により選択される可能性として提示されるアミノ酸とを比べることが興味深い。

【0554】

[00723] NSLは、プログラミングコードを消費して、プログラミングコードを自然言語情報フォーマットに変換することが可能である。(同じことが、図面102A~102Dを参照して更に詳細に説明されている)。

10

【0555】

[00724] 同様に、DNAに含まれる情報を抽出し、自然言語サーチのドメインに入ることができるようにNSLに変換する潜在性も保持する。(同じことが、図面73を参照して更に詳細に説明されている)。

【0556】

[00725] NSLは現在、分化原理に基づいて、プログラミング論理に属する情報技術における抽象化レベルへの対処にフォーカスしている。(同じことが、図面66を参照して更に詳細に説明されている)。

20

【0557】

[00726] 分化原理の適用はプログラミング層のみに限定されないため、将来、同じ原理をオペレーティングシステム以下における抽象化レベルにも適用する潜在性がある。

【0558】

[00727] それが生じると、システム内のブラックボックスにある機能の多くは、よりよい理解及び管理のために、ユーザによりトランスペアレントになることができる。

【0559】

[00728] NSLの非依存性はまた、NSLフォーマットの言語に含まれる論理を抽出し、それらのソリューションを新しいものと全く同じように使用する能力を有するため、過去に書かれたプログラムにさえも拡張される。

30

【0560】

[00729] NSLは、任意のプログラミング言語を自然言語のようなNSLフォーマットに変換する能力を有する。(同じことが、図面75に更に詳細に説明されている)。

【0561】

[00730] NSLは、NSLで構築されたソリューションを任意のプログラミング言語コードに変化する能力を有する。(同じことが、図面75に更に詳細に説明されている)。

【0562】

[00731] NSLは、任意のプログラミング言語のソリューションを任意の他のプログラミング言語/自然言語に変換する能力を有する。(同じことが、図面75を参照して更に詳細に説明されている)。

40

【0563】

[00732] NSLは、一連の予め定義されたクエリへの応答を通してソリューション構造体を提供する。(同じことが、図面76を参照して更に詳細に説明されている)。

【0564】

[00733] NSLは、本を読むのと同種のソリューションを見る経験をユーザに提供する。(同じことが、図面77を参照して更に詳細に説明されている)。

【0565】

[00734] NSLは、動詞を関数に自動的に割り当て、関数を自然言語に近い見たいにする。(同じことが、図面78を参照して更に詳細に説明されている)。

【0566】

50

【00735】 任意の既存の実施からの論理は、NSLにより抽出することができる。(同じことが、図面79を参照して更に詳細に説明されている)。

【0567】

【00736】 NSLは、ソリューション論理を基盤非依存にする。(同じことが、図面80を参照して更に詳細に説明されている)。

【0568】

【00737】 NSLは、イベント、情報、空間、時間、及びリソース消費により実体間の距離の識別を通してソリューションを定量化する。(同じことが、図面81を参照して更に詳細に説明されている)。

【0569】

【00738】 ソリューション論理をトランスペアレントにし、既存のソリューションコンポーネントを再使用することにより、NSLは、冗長性をなくすことを通して論理圧縮に役立つ。(同じことが、図面82を参照して更に詳細に説明されている)。

【0570】

【00739】 NSLは、ソリューション論理の圧縮を理由として計算リソースの低減に役立つ。

【0571】

【00740】 NSLの原理はエンジニアリング設計に適用することができる。(同じことが、図面83を参照して更に詳細に説明されている)。

【0572】

【00741】 NSLは、制御された分化にコンテナシステムを採用し、IDをクラス、サブクラス、及びメンバに自動的に割り当てる。(同じことが、図面85を参照して更に詳細に説明されている)。

【0573】

【00742】 NSLは、静的ドキュメントのNSLフォーマットへの変換を提供する。(同じことが、図面86を参照して更に詳細に説明されている)。

【0574】

【00743】 NSLは、通常の画面を超えて、5つの1次性感覚及び追加の感覚を網羅するようにユーザインターフェースを拡張する。(同じことが、図面87を参照して更に詳細に説明されている)。

【0575】

【00744】 NSLは、リソースの使用時間及びアイドル時間の追跡に役立つ。(同じことが、図面89を参照して更に詳細に説明されている)。

【0576】

【00745】 NSLは、フローチャートからソリューション論理を抽出し、ソリューションをNSLフォーマットで再構築する。

【0577】

【00746】 NSLは、ソリューションレベルトランザクションカウントを提供し、例えば、所与のソリューション論理を使用したユーザのカウントを識別する。

【0578】

【00747】 特定の期間中に実体を使用した回数又は特定の時間に使用中の実体の数は、NSLにより捕捉することができる。

【0579】

【00748】 NSLは、ユーザ経験を保存しながら任意の既存のソリューションとシームレスに統合する、自然言語構造体で開発されるAPIを有する。

【0580】

【00749】 NSLは、機械エージェントによるソリューション開発を支援するDLLエンジンを提供する。(同じことが、図面116を参照して更に詳細に説明されている)。

【0581】

【00750】 NSLは、SOPのNSLフォーマットへの変換を提供する。(同じことが

10

20

30

40

50

、図面 1 1 7 を参照して更に詳細に説明されている)。

【 0 5 8 2 】

[00751] 全ての变化単位は連続し、グローバル意図言明を達成する。(同じことが、図面 1 1 8 を参照して更に詳細に説明されている)。

【 0 5 8 3 】

[00752] X O R 演算子に続く代替の選択肢の場合、全ての变化単位は代替の様式で配置されて、グローバル意図言明を達成する。(同じことが、図面 1 1 9 を参照して更に詳細に説明されている)。

【 0 5 8 4 】

[00753] 並列選択肢トリガーの場合、全ての变化単位は複数のエージェントと共に並列に配置されて、グローバル意図言明を達成する。(同じことが、図面 1 2 0 を参照して更に詳細に説明されている)。

10

【 0 5 8 5 】

[00754] 視座を上げるにつれて、より低い視座における变化単位は下位变化単位になる。(同じことが、図面 1 2 1 を参照して更に詳細に説明されている)。

【 0 5 8 6 】

[00755] G S I が複数のエージェントの G S I にわたって広がる場合、エージェントの別の G S I の变化単位をトリガーするあるエージェントの G S I は一時的変化単位になる。(同じことが、図面 1 2 2 を参照して更に詳細に説明されている)。

【 0 5 8 7 】

[00756] 变化単位はそれ自体、組み込みサブ变化単位と呼ばれる複数のローカル意図言明を有する。(同じことが、図面 1 2 3 を参照して更に詳細に説明されている)。

20

【 0 5 8 8 】

[00757] 設計者により設計されたソリューションは、呼び出されたソリューションクラスにあり、イベントが到着すると、開発はトランザクションクラスになる。(同じことが、図面 1 2 4 を参照して更に詳細に説明されている)。

【 0 5 8 9 】

[00758] コンテキストトリプルアイデンティティモデルは、組織レベルから属性の最下位レベルまでのアイデンティティを許容する。(同じことが、図面 1 2 5 を参照して更に詳細に説明されている)。

30

【 0 5 9 0 】

[00759] N S L は、幾つかの基盤の何れかからのソリューションを許容するハブを機能する。(同じことが、図面 1 2 6 を参照して更に詳細に説明されている)。

【 0 5 9 1 】

図面の詳細な説明

[00760] 図面 9 1 は、プログラミングコードを使用せずに、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する方法を示すフローチャートを表す。方法によれば、ステップ 9 1 0 2 において、ユーザから計算デバイスのプロセッサにより、ソリューションを構築するグローバル意図言明を受信し、グローバル意図言明は、自然言語の形態で受信され、潜在状態に設定される。ステップ 9 1 0 4 において、ユーザからプロセッサにより、グローバル意図言明に関連付けられた 1 つ又は複数のローカル意図言明並びに各ローカル意図言明に関連付けられた実体及びエージェントの詳細を受信し、各ローカル意図言明並びに各実体及びエージェントの詳細は、自然言語の形態で受信され、それぞれ潜在状態に設定され、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは、人間エージェント、機械エージェント、及びコンピュータ機能の少なくとも 1 つである。ステップ 9 1 0 6 において、プロセッサにより、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて 1 組の実体 - 状態組合せ (C E S) を形成し、各ローカル意図言明の全ての実体に基づいて形成された C E S は、トリガー実体状態組合せ (トリガー C E S) であり、組中の各 C E S は潜在状

40

50

態である。ステップ 9 1 0 8 において、グローバル意図言明に関連付けられた、受信したローカル意図言明が 1 つのみであると判断し、受信したローカル意図言明のトリガー C E S をソリューション構築の終わりとして識別する。ステップ 9 1 1 0 において、グローバル意図言明に関連付けられた、受信したローカル意図言明が 2 つ以上であると判断し、ユーザからプロセッサにより自然言語の形態で、ローカル意図言明間の関係を受信し、関係は、1 つのローカル意図言明のトリガー C E S が別のローカル意図言明の 1 組の C E S に影響しているか、それともソリューション構築の終わりであることを示す。

【 0 5 9 2 】

[00761] 図面 9 2 は、プログラミングコードを自然言語に変換して、自然言語ベースのコンピュータ実施ソリューションを構築する方法を示すフローチャートを表す。方法によれば、ステップ 9 2 0 2 において、計算デバイスのプロセッサにより、プログラミングコードをパズして、プログラミングコードに存在する記号、キーワード、演算子、及び関数を抽出する。ステップ 9 2 0 4 において、プロセッサにより、データベースに記憶された自然言語データへの抽出された記号、キーワード、演算子及び関数のマッピングに基づいて、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、及び実体のそれぞれに関連付けられた属性を特定し、複数のローカル意図言明のそれぞれ、実体、及び属性を潜在状態に設定し、各ローカル意図言明はソリューションを構築するステップを示し、各実体は対応するローカル意図言明により示されるステップに参加し、属性は各実体の特性を定義し、対応するローカル意図言明の他の実体から各実体を差別化する。ステップ 9 2 0 6 において、プロセッサにより、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて 1 組の実体 - 状態組合せ (C E S) を形成し、各ローカル意図言明の全ての実体に基づいて形成された C E S は、トリガー実体状態組合せ (トリガー C E S) であり、組中の各 C E S は潜在状態である。ステップ 9 2 0 8 において、プロセッサにより、データベースに記憶された自然言語データへの抽出された演算子及び関数のマッピングに基づいて、ローカル意図言明間の関係を特定し、関係は、1 つのローカル意図言明のトリガー C E S が別のローカル意図言明の 1 組の C E S に影響しているか、それともコンピュータ実施ソリューションを構築するグローバル意図言明の終わりであることを示し、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される。ステップ 9 2 1 0 において、ユーザからプロセッサにより、自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定し、エージェントは、人間エージェント、機械エージェント、及びコンピュータ機能の少なくとも 1 つである。

【 0 5 9 3 】

[00762] 図面 9 3 は、自然言語に基づいて構築されたコンピュータ実施ソリューションをプログラミングコードに変換する方法を示すフローチャートを表す。方法によれば、ステップ 9 3 0 2 において、計算デバイスのプロセッサによりコンピュータ実施ソリューションをパズし、コンピュータ実施ソリューションは、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、各実体に関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ (C E S)、1 つ又は複数のトリガー C E S を含み、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、各実体の特性を定義し、対応するローカル意図言明の他の実体から各実体を差別化し、関係は、あるローカル意図言明のトリガー C E S が別のローカル意図言明の 1 組の C E S に影響しているか、それともコンピュータ実施ソリューションを構築するグローバル意図言明の終わりであることを示す。ステップ 9 3 0 4 において、計算デバイスのプロセッサにより自然言語ソリューションコンポーネントを抽出し、自然言語ソリューションコンポーネントは、グローバル意図言明、複数のローカル意図言明、各実体に関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ (C E S)、1 つ又は複数のトリガー C E S を含む。ステップ 9 3 0 6 において、計算デバイスのプロセッサにより、データベースに記憶され

10

20

30

40

50

たプログラミング言語の記号、キーワード、演算子、及び関数への自然言語ソリューションコンポーネントのマッピングに基づいてプログラミング言語コードを自動生成する。

【0594】

【00763】 NSLは、1つ若しくは複数のコンピュータシステムで実施し得、又は1つ若しくは複数のコンピュータシステムを含み得る。図面94は、計算環境9402の一般化された例を示す。計算環境9402は、記載される実施形態の使用又は機能の範囲についてのいかなる限定の示唆も意図しない。

【0595】

【00764】 図面94を参照すると、計算環境9402は少なくとも1つの処理ユニット9404及びメモリ9406を含む。処理ユニット9404はコンピュータ実行可能命令を実行し、現実又は仮想プロセッサであり得る。マルチ処理システムでは、複数の処理ユニットがコンピュータ実行可能命令を実行して、処理力を上げる。メモリ9406は、揮発性メモリ(例えば、レジスタ、キャッシュ、RAM)、不揮発性メモリ(例えば、ROM、EEPROM、フラッシュメモリ等)、又はそれら2つの何らかの組合せであり得る。幾つかの実施形態では、メモリ9406は、記載される技法を実施するフレームワーク9416を記憶する。

【0596】

【00765】 計算環境は追加の特徴を有し得る。例えば、計算環境9402は記憶装置9414、1つ又は複数の入力デバイス9410、1つ又は複数の出力デバイス9412、及び1つ又は複数の通信接続9408を含む。バス、コントローラ、又はネットワーク等の相互接続機構(図示せず)は、計算環境9402のコンポーネントを相互接続する。通常、オペレーティングシステムソフトウェア(図示せず)は、計算環境9402で実行中の他のソフトウェアに動作環境を提供し、計算環境9402のコンポーネントの活動をコーディネートする。

【0597】

【00766】 記憶装置9414は、リムーバブルであってもよく又は非リムーバブルであってもよく、磁気ディスク、磁気テープ若しくはカセット、CD-ROM、CD-RW、DVD、又は情報の記憶に使用し得、計算環境9402内でアクセスし得る任意の他の媒体を含む。幾つかの実施形態では、記憶装置9414はフレームワーク9416の命令を記憶する。

【0598】

【00767】 入力デバイス9410は、キーボード、マウス、ペン、トラックボール、タッチスクリーン、又はゲームコントローラ等のタッチ入力デバイス、音声入力デバイス、スキャンデバイス、デジタルカメラ、又は入力を計算環境9402に提供する別のデバイスであり得る。出力デバイス9412は、ディスプレイ、プリンタ、スピーカ、又は出力を計算環境9402から提供する別のデバイスであり得る。

【0599】

【00768】 通信接続9408は、通信媒体を介して別の計算エンティティと通信できるようにする。通信媒体は、コンピュータ実行可能命令、音声若しくは動画情報、又は他のデータ等の情報を変調データ信号で伝達する。変調データ信号は、特性セットの1つ又は複数を有し、又は情報を信号に符号化するように変更された信号である。限定ではなく例として、通信媒体は、電気、光学、RF、赤外線、音響、又は他の搬送波を用いて実施される有線又は無線技法を含む。

【0600】

【00769】 実施態様は、コンピュータ可読媒体の一般状況で説明し得る。コンピュータ可読媒体は、計算環境内でアクセスし得る任意の利用可能な媒体である。限定ではなく例として、計算環境9402内で、コンピュータ可読媒体は、メモリ9406、記憶装置9414、通信媒体、及び上記の何れかの組合せを含む。

【0601】

【00770】 記載された実施形態を参照して本開示の原理を説明し示したが、そのような

10

20

30

40

50

原理から逸脱せずに、記載された実施形態の構成及び詳細が変更可能であることが認識されよう。本開示の原理を適用し得る多くの可能な自体に鑑みて、特許請求の範囲の趣旨及びその均等物内にあり得るような全ての実施形態を本開示として特許請求する。

【0602】

[00771] 本開示は上記実施形態に関して関連するが、本開示が示された実施形態に限定されないことを当業者ならば認識しよう。本開示は、添付の特許請求の範囲の趣旨及び範囲内で変更及び改変を用いて実施し得る。したがって、説明は、本開示への限定ではなく例示として見なされるべきである。

【0603】

[00772] 当業者により理解されるように、上記の例、実証、及び方法ステップは、汎用又は専用コンピュータ等のプロセッサベースシステムで適したコードにより実装し得る。本技法の異なる実施態様が、本明細書に記載されるステップ幾つかの又は全てを異なる順序で又は略同時に、すなわち並列して実行し得ることに留意されたい。さらに、機能は多種多様なプログラミング言語で実装し得る。そのようなコードは、当業者により理解されるように、メモリチップ、ローカル若しくはリモートハードディスク、光ディスク、又はプロセッサベースシステムによりアクセスされて、記憶されたコードを実行し得る他の媒体等の1つ又は複数の有形機械可読媒体に記憶し得、又は記憶に向けて適合し得る。なお、有形媒体は、命令がプリントされる紙又は別の適した媒体を含み得る。例えば、命令は、紙又は他の媒体の光学スキャンを介して電子的に捕捉され、次に、必要な場合、コンパイル、インタプリタ、又は他の方法で適したように処理され、次にコンピュータメモリに記憶し得る。

【0604】

[00773] 詳細な説明は、当業者が本開示を作製、使用できるようにするのに提示され、特許取得の要件の状況で提供されている。本説明は、現在考えられる、本開示を実行するのに最もよい方法である。本実施形態への種々の変更は、当業者に容易に明らかになり、本開示の一般原理は他の実施形態に適用し得、本開示の幾つかの特徴は、他の特徴の対応する使用なしで使用し得る。したがって、本開示は、示される実施形態に限定されることが意図されず、本明細書に記載される原理及び特徴と一貫する最も広い範囲に従うべきである。

【0605】

[00774] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する方法に関し、方法は、ユーザから計算デバイスのプロセッサにより、ソリューションを構築するグローバル意図言明を受信することであって、グローバル意図言明は、自然言語の形態で受信され、潜在性の状態で設定される、受信することと、ユーザからプロセッサにより、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明並びに各ローカル意図言明に関連付けられた実体及びエージェントの詳細を受信することであって、各ローカル意図言明並びに各実体及びエージェントの詳細は、自然言語の形態で受信され、それぞれ潜在性の状態で設定され、各ローカル意図言明は、ソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは、人間エージェント及び機械エージェントの少なくとも一方である、受信することと、プロセッサにより、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体-状態組合せ(CES)を形成することであって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー-実体状態組合せ(トリガーCES)であり、1組中の各CESは潜在性の状態に設定される、形成することと、グローバル意図言明に関連付けられた、受信したローカル意図言明が1つのみであるとの判断に回答して、受信したローカル意図言明のトリガーCESをソリューションの構築の終了として識別することと、グローバル意図言明に関連付けられた、受信したローカル意図言明が2つ以上であるとの判断に回答して、ユーザからプロセッサにより自然言語の形態で、ローカル意図言明間の関係を受信することであって、関係

10

20

30

40

50

は、あるローカル意図言明のトリガー C E S が、別のローカル意図言明の C E S の組に影響しているか、それともソリューションの構築の終了であることを示す、関係を受信することを含む。

【0606】

[00775] 実施態様例によれば、上記方法は、ローカル意図言明毎に、ユーザからプロセッサにより自然言語の形態で、ローカル意図言明のそれぞれの特性を定義する1つ又は複数の属性を受信することを含む。

【0607】

[00776] 実施態様例によれば、上記方法は、実体毎に、ユーザからプロセッサにより自然言語の形態で、実体のそれぞれの特性を定義し、実体のそれぞれを対応するローカル意図言明の他の実体から区別する1つ又は複数の属性を受信することを含む。

10

【0608】

[00777] 実施態様例によれば、関係を受信することは、グローバル意図言明のローカル意図言明間の複数の別個の関係を受信することを含み、別個の各関係は、ソリューションを構築する別個の経路である。

【0609】

[00778] 実施態様例によれば、上記方法は、各ローカル意図言明の実体毎に、関連するエージェントからプロセッサにより自然言語の形態で、実体のそれぞれに対する値を受信することであって、実体のそれぞれに対する値を受信することは、受信される値に基づいて、実体のそれぞれについての潜在性の状態を現実性の状態に変えるイベントの記録である、値を受信することを含む。

20

【0610】

[00779] 実施態様例によれば、上記方法は、各ローカル意図言明の属性毎に、関連するエージェントからプロセッサにより自然言語の形態で、属性のそれぞれに対する値を受信することであって、属性のそれぞれに対する値を受信することは、受信される値に基づいて、属性のそれぞれについての潜在性の状態を現実性の状態に変えるイベントの記録である、値を受信することを含む。

【0611】

[00780] 実施態様例によれば、上記方法は、各実体の属性について、関連するエージェントからプロセッサにより自然言語の形態で、属性のそれぞれに対する値を受信することであって、属性のそれぞれに対する値を受信することは、受信される値に基づいて、属性のそれぞれについての潜在性の状態を現実性の状態に変えるイベントの記録である、値を受信することを含む。

30

【0612】

[00781] 実施態様例によれば、上記方法は、プロセッサにより、ログイン詳細、バイオメトリック詳細、顔認識技法、及び網膜検出技法の1つに基づいて関連するエージェントを認証することを含む。

【0613】

[00782] 実施態様例によれば、各ローカル意図言明に関連する全ての実体に対する値を受信することは、ローカル意図言明のそれぞれについての潜在性の状態を現実性の状態に変えるイベントの記録である。

40

【0614】

[00783] 実施態様例によれば、各ローカル意図言明に関連する全ての実体に対する値を受信することは、グローバル意図言明についての潜在性の状態を現実性の状態に変えるイベントの記録である。

【0615】

[00784] 実施態様例によれば、機械エージェントは、有線通信チャネル又は無線通信チャネルを介して計算デバイスに通信可能に結合される。

【0616】

[00785] 実施態様例によれば、機械エージェントは、計算デバイスの機能モジュール

50

であり、プロセッサに結合される。

【0617】

[00786] 実施態様例によれば、グローバル意図言明に関連するローカル意図言明は、他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションのローカル意図言明又は他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションのグローバル意図言明である。

【0618】

[00787] 実施態様例によれば、グローバル意図言明のローカル意図言明に関連する実体は、他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションのローカル意図言明又は他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションの実体である。

10

【0619】

[00788] 実施態様例によれば、ローカル意図言明のそれぞれの特性を定義する属性は、他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションの実体又は他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションの属性である。

【0620】

[00789] 実施態様例によれば、実体のそれぞれの特性を定義する属性は、他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションの実体又は他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションの属性である。

20

【0621】

[00790] 実施態様例によれば、ソリューションは、ローカル意図言明を2つ以上の他のローカル意図言明に分割し、対応する実体及びエージェントを各ローカル意図言明に添付すること又は2つ以上のローカル意図言明を結合して1つのローカル意図言明にし、対応する実体及びエージェントをそのローカル意図言明に添付することにより構築される。

【0622】

[00791] 実施態様例によれば、ソリューションは、自然言語を使用して構築されるため、1つ又は複数のサーチエンジンによりサーチ可能である。

【0623】

30

[00792] 実施態様例によれば、上記方法は、各実体にわたる1つ又は複数の情報権又は決定権を実体のそれぞれのローカル意図言明に関連するエージェントに割り当てることと、割り当てられた情報権又は決定権に基づいて実体をエージェントに表示することを含む。

【0624】

[00793] 実施態様例によれば、属性の数が少ないほど、実体のそれぞれは一般化され、属性の数が多きほど、実体間の分化が進む。

【0625】

[00794] 実施態様例によれば、上記方法は、ユーザインターフェースを通して実体を表示して、関連するエージェントにより実体に対する値を受信することを含む。

40

【0626】

[00795] 実施態様例によれば、上記方法は、ローカル意図言明間の1つ又は複数の別個の関係を提供して、実体に対する値を受信することを含み、1つ又は複数の別個の関係を提供することは、組織化されたプロセス、自動プロセス、又は人間エージェント駆動プロセスに基づく。

【0627】

[00796] 実施態様例によれば、実体の1つ若しくは複数の又はローカル意図言明の1つ若しくは複数のCESは、他のコンピュータ実施ソリューションの1つ又は複数のローカル意図言明と共有され、共有実体又はCESは、所与の時点で1つのみのトリガCESに参加して、1つのトリガCESに対応するローカル意図言明について潜在性状態が

50

ら現実性状態に変更し、共有実体又はCESの可用性は、複数のローカル意図言明についての潜在性状態から現実性状態への変更の順序に影響する。

【0628】

[00797] 実施態様例によれば、実体は本質的に物理的であり、実体は時間及び空間に存在する。

【0629】

[00798] 実施態様例によれば、ローカル意図言明を受信すること及び実体の詳細を受信することは、他のコンピュータ実施ソリューションのローカル意図言明及び実体を記憶したデータベース中のライブラリから1つ又は複数のローカル意図言明候補及び1つ又は複数の実体候補を提供することを含む。

10

【0630】

[00799] 実施態様例によれば、上記方法は、プロセッサにより、実体に対する値を受け取る別個の関係の1つを選択し、他の別個の関係をなくすことを含む。

【0631】

[00800] 実施態様例によれば、自然言語は、1つ若しくは複数の母語、1つ若しくは複数の手話、1つ若しくは複数の数字、又はそれらの組合せに基づく。

【0632】

[00801] 実施態様例によれば、自然言語の形態の情報は、手書きベースインターフェース、タッチセンシティブインターフェース、音声ベースインターフェース、画像ベースインターフェース、ビデオベースインターフェース、又は組合せを通して受信され、自然言語の形態で受信される情報は、物体認識技術、文字認識技術、画像認識技術、又はそれらの組合せを使用して解読される。

20

【0633】

[00802] 実施態様例によれば、上記方法は、プロセッサにより、記録すべき残りのイベント数を特定することと、プロセッサにより残りのイベント数を表示することを含む。

【0634】

[00803] 実施態様例によれば、各実体及び各ローカル意図言明のユーザインターフェース上の場所は、ドラッグアンドドロップ機能であって、有線入力デバイス、無線入力デバイス、及びタッチセンシティブインターフェースの少なくとも1つにより実行される、ドラッグアンドドロップ機能又は、実体のそれぞれ若しくはローカル意図言明のそれぞれの場所座標に対応する属性の値の変更により変更可能である。

30

【0635】

[00804] 実施態様例によれば、上記方法は、公開ブロックチェーン技術、秘密ブロックチェーン技術、及びハイブリッドブロックチェーン技術の1つを適用することを含む。

【0636】

[00805] 実施態様例によれば、公開ブロックチェーン技術、秘密ブロックチェーン技術、及びハイブリッドブロックチェーン技術のそれぞれは、対称鍵暗号化技法、非対称鍵暗号化技法、又はそれらの組合せに基づく。

【0637】

[00806] 実施態様例によれば、上記方法は、1つ又は複数の予め定義された規範に基づいて測定フレームワークを実体に割り当てることを含み、測定フレームワークは、実体が潜在性状態から現実性状態の変更される時間期間を示す。

40

【0638】

[00807] 実施態様例によれば、上記方法は、ユーザからプロセッサにより、各ローカル意図言明に関連する情報実体の詳細を受信することであって、各情報実体の詳細は、自然言語の形態で受信され、潜在性状態に設定され、各情報実体は、トリガースCESを形成する実体と異なる、受信することを含む。

【0639】

[00808] 実施態様例によれば、上記方法は、データベース中のライブラリの形態で、別のコンピュータ実施ソリューションの構築で使用するために、グローバル意図言明、ロ

50

ーカル意図言明、実体、及びエージェントの1つ又は複数に関連するデータを記憶することを含む。

【0640】

[00809] 実施態様例によれば、上記方法は、データベース内のライブラリへのクエリベースのアクセスを提供することを含み、クエリベースのアクセスは、再使用に適切な実体の識別のためのサーチ及びクエリベースの技術に依存し、クエリベースのアクセスは、構造化クエリ言語 (SQL) 及びノットオンリー構造化クエリ言語 (NoSQL) の少なくとも一方を含む。

【0641】

[00810] 実施態様例によれば、上記方法は、ドラッグアンドドロップアクセスをデータベース中のライブラリに提供することを含む。

10

【0642】

[00811] 実施態様例によれば、上記方法は、CESの1つ又は複数の過去の挙動及びイベントの1つ又は複数の発生に基づいて、実体の1つ又は複数及びイベントの1つ又は複数に対応する1つ又は複数の不確実性及び確率を提供することを含む。

【0643】

[00812] 実施態様例によれば、上記方法は、会計システム及び金融システムに関する1つ又は複数の実体をコンピュータ実施ソリューションの各ローカル意図言明に添付することにより、会計システム及び金融システムを統合することを含む。

【0644】

20

[00813] 実施態様例によれば、上記方法は、1つ又は複数のアドバンスドプランニングアンドオプティマイゼーション (APO) 機能を実行して、実体の1つ又は複数及びイベントの1つ又は複数に対応する1つ又は複数の不確実性及び確率を提供し、イベントの記録についての値の受信を最適化することを含む。

【0645】

[00814] 実施態様例によれば、上記方法は、機械学習技法及び人工知能技法、実体中心手法であって、実体中心手法は1つ又は複数のイベントの発生を提供する、実体中心手法、並びに各イベントのデータ及び挙動パターンの記憶の少なくとも1つに基づいて、CESの1つ又は複数の過去の挙動及びイベントの1つ又は複数の発生を査定することを含み、機械学習技法は、教師あり学習技法、教師なし学習技法、及び/又は半教師あり学習技法を含み、教師あり学習技法は、サポートベクターマシン、線形回帰、ロジスティック回帰、ナイーブベイズ、線形判別分析、決定木、k最近傍アルゴリズム、及びニューラルネットワークの少なくとも1つを含み、教師なし学習技法は、階層クラスタリング、K平均クラスタリング、K-NN (k最近傍)、及び相関ルールの少なくとも1つを含む。

30

【0646】

[00815] 実施態様例によれば、上記方法はを含む、グローバル意図言明、各ローカル意図言明、及び実体の詳細の1つ又は複数を受信することは、対話式質問表に回答してのものであり、対話式質問表は、コンピュータ実施ソリューションを構築するための構造化フォーマットでの質問を含む。

【0647】

40

[00816] 実施態様例によれば、上記方法は、自然言語文法に基づいて受信したグローバル意図言明、各ローカル意図言明、及び実体の詳細の1つ又は複数を変更することを含み、自然言語文法は、自然言語ライブラリを使用して、修正に適切な1つ又は複数の動詞及び前置詞を選ぶ。

【0648】

[00817] 実施態様例によれば、上記方法は、プロセッサにより、残りの数のイベントに必要な時間及びリソースの量を特定することと、プロセッサにより、必要とされる時間及びリソースの量を表示することとを含む。

【0649】

[00818] 実施態様例によれば、上記方法は、プロセッサにより、潜在状態から現実状

50

態に変化するとき、記録される2つの連続イベント間の最適アイドル時間をデータベースに記憶することと、プロセッサにより、潜在状態から現実状態に変化するとき、記録される2つの連続イベントのそれぞれ間のアイドル時間をリアルタイムで特定することと、プロセッサにより、特定されたアイドル時間と最適アイドル時間との比較に基づいてレポートを準備することを含む。

【0650】

[00819] 実施態様例によれば、上記方法は、プロセッサにより、受信した値をデータベースに記憶された最適値と比較することと、プロセッサにより、比較に基づいて、受信した値が、コンピュータ実施ソリューションの構築に良好、不良、それとも曖昧であるかを判断することを含む。

10

【0651】

[00820] 実施態様例によれば、上記方法は、一意の識別子（ID）を各ローカル意図言明、各実体、及び各エージェントに割り当てることを含む。

【0652】

[00821] 実施態様例によれば、上記方法は、ユーザが、リソースアイドル時間を最適化するように、全ての利用可能な実体の最適な使用を定義できるようにすることと、リアルタイムで、各実体に関連付けられた時間及び空間の属性の監視を通して価値創造活動への実体の参加を追跡することを含む。

【0653】

[00822] 実施態様例によれば、上記方法は、1つ又は複数の結果可能性及び1つ又は複数の機会又は1つ又は複数のリスクの1つ又は複数の価値判断を提供し、1つ又は複数の結果可能性に対する1つ又は複数の確率をエージェントに割り当てることと、1つ又は複数の行動を実行することとであって、それにより、1つ又は複数の実体を改変することにより1つ又は複数の経路に介入して改変することとであって、行動は、関連付けられたエージェントにより望まれる1つ又は複数のリソース最適化原理を駆動する、実行することを含む。

20

【0654】

[00823] 実施態様例によれば、上記方法は、1つ又は複数の別個の経路を選び抜くための1つ又は複数の可能な経路を提供することを含み、1つ又は複数の可能な経路は、データベース中のライブラリからサーチされる、コンピュータ実施ソリューションと同様のソリューションである。

30

【0655】

[00824] 実施態様例によれば、上記方法は、1つ又は複数の教師あり及び/又は1つ又は複数の教師なし機械学習法をデータベース中のライブラリに対して実施することを含み、1つ又は複数の教師あり及び/又は1つ又は複数の教師なし機械学習法の実施は、ソリューション内容を処理するNLPコンポーネント、ANNコンポーネント、及び最近傍コンポーネントを含むDLDEンジンによるものであり、DLDEンジンは、ライブラリの一部であるリッチソリューション内容をマイニングし、トランザクションクラスからトランザクションデータを抽出し、NSLソリューションライブラリの一部であるリッチソリューション内容及びトランザクションクラスから抽出されたトランザクションデータは、DLDEンジンに供給され、DLDEンジンは、種々のレベルでの距離の計算を可能にし、距離計算は、機械学習アルゴリズムを使用してNSL実体値、時間、及び空間を使用している。

40

【0656】

[00825] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する方法に関し、方法は、計算デバイスのプロセッサにより、自然言語の形態でグローバル意図言明を入力するプロンプトをユーザに提供し、ソリューションを構築するグローバル意図言明を潜在状態に設定することと、プロセッサにより、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明と、各ローカル意図言明に関連付けられた実体及びエー

50

ジェントの詳細とを入力するプロンプトをユーザに自然言語の形態で提供し、1つ又は複数のローカル意図言明及び実体のそれぞれを潜在状態に設定することによって、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、提供し設定することと、プロセッサにより、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の全ての可能な組合せに基づいて1組の実体 - 状態組合せ (CES) を形成することによって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されたCESは、トリガー実体状態組合せ (トリガーCES) であり、組中の各CESは潜在状態である、形成することと、グローバル意図言明に関連付けられた、受信されたローカル意図言明が1つのみであるとの判断に回答して、プロセッサにより、受信したローカル意図言明のトリガーCESをソリューションの構築の終わりとして識別することと、グローバル意図言明に関連付けられた、受信されたローカル意図言明が2つ以上であるとの判断に回答して、プロセッサにより、ローカル意図言明間の関係を入力するプロンプトをユーザに自然言語の形態で提供することによって、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともソリューションの構築の終わりであることを示す、提供することを含む。

10

【0657】

[00826] 実施態様例によれば、上記方法は、ローカル意図言明毎に、プロセッサにより、ローカル意図言明のそれぞれの特性を定義する1つ又は複数の属性を自然言語の形態で入力するプロンプトをユーザに提供することを含む。

20

【0658】

[00827] 実施態様例によれば、上記方法は、実体毎に、プロセッサにより、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から各実体を差別化する1つ又は複数の属性を自然言語の形態で入力するプロンプトをユーザに提供することを含む。

【0659】

[00828] 実施態様例によれば、関係についてのプロンプトを提供することは、グローバル意図言明についてのローカル意図言明間の複数の別個の関係をを入力するプロンプトをユーザに提供することを含み、別個の各関係は、ソリューションを構築する別個の経路である。

【0660】

30

[00829] 実施態様例によれば、上記方法は、ローカル意図言明の実体毎に、プロセッサにより、実体のそれぞれに対する価値を自然言語の形態で入力するプロンプトに関連付けられたエージェントに提供することを含み、実体のそれぞれに対する値を受信することは、受信した値に基づいて実体のそれぞれを潜在状態から現実実体に変えるイベントの記録である。

【0661】

[00830] 実施態様例によれば、上記方法は、各ローカル意図言明の属性毎に、プロセッサにより、属性のそれぞれに対する値を自然言語の形態で入力するプロンプトに関連付けられたエージェントに提供することを含み、属性のそれぞれに対する値を受信することは、受信した値に基づいて属性のそれぞれを潜在状態から現実実体に変えるイベントの記録である。

40

【0662】

[00831] 実施態様例によれば、上記方法は、各実体の属性について、プロセッサにより、属性のそれぞれに対する値を自然言語の形態で入力するプロンプトに関連付けられたエージェントに提供することを含み、属性のそれぞれに対する値を受信することは、受信した値に基づいて属性のそれぞれを潜在状態から現実実体に変えるイベントの記録である。

【0663】

[00832] 実施態様例によれば、上記方法は、プロセッサにより、ログイン詳細、バイオメトリック詳細、顔認識技法、及び網膜検出技法の1つに基づいて関連付けられたエージェントを認証することを含む。

50

【0664】

[00833] 実施態様例によれば、上記方法は、ローカル意図言明に関連付けられた実体のそれぞれについてイベントが記録されると、プロセッサにより、ローカル意図言明のそれぞれを潜在状態から現実状態に変えるイベントを記録することを含む。

【0665】

[00834] 実施態様例によれば、上記方法は、各ローカル意図言明を潜在状態から現実状態に変えるイベントが記録されると、プロセッサにより、グローバル意図言明を潜在状態から現実状態に変えるイベントを記録することを含む。

【0666】

[00835] 実施態様例によれば、自然言語は、1つ若しくは複数の母国語、1つ若しくは複数の手話、1つ若しくは複数の記号、1つ若しくは複数の数字、又はそれらの組合せに基づく。

【0667】

[00836] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、動画及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する方法に関し、方法は、計算デバイスのプロセッサにより、動画の画像フレームを処理することであって、それにより、複数の差別化された画像フレームを識別し、分化した各画像フレームは、その内容の違いに基づいて動画の1つ又は複数の先の画像フレームと異なるものとして識別される、処理することと、プロセッサにより、それぞれ動画の直前の画像フレームの内容に関する差別化された各画像フレームの内容の差に基づいて、複数のローカル意図言明を特定し、ローカル意図言明のそれぞれを潜在状態に設定することであって、各ローカル意図言明は、ソリューションを構築するステップを示し、データベースに記憶された自然言語データに関して画像フレームの内容を分析することにより特定される、特定し設定することと、プロセッサにより、データベースに記憶された自然言語データに関して差別化された画像フレームの連続対間の画像フレームの内容を分析することにより、複数のローカル意図言明のそれぞれに関連付けられた実体及び実体のそれぞれに関連付けられた属性を特定することであって、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定することと、プロセッサにより、最初の差別化された画像フレームと最後の差別化された画像フレームの内容の変化に基づいて、コンピュータ実施ソリューションのグローバル意図言明を特定し、グローバル意図言明を潜在状態に設定することであって、グローバル意図言明は、データベースに記憶された自然言語データに関して、最初の差別化された画像フレーム及び最後の差別化された画像フレームの内容を分析することにより特定される、特定し設定することと、プロセッサにより、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の全ての可能な組合せに基づいて1組の実体状態組合せ(CES)を形成することであって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、組中の各CESは潜在状態である、形成することと、ユーザからプロセッサにより自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定することであって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することと、ユーザからプロセッサにより自然言語の形態で、ローカル意図言明間の関係を受信することであって、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであることを示す、受信することを含む。

【0668】

[00837] 実施態様例によれば、画像フレームを処理することは、画像フレームの内容及び画像フレームにわたる内容の変化に基づいて情報を抽出することを含み、情報を抽出することは、物体認識技術、文字認識技術、音声認識技術、又はそれらの組合せに基づく。

【0669】

10

20

30

40

50

【00838】 実施態様例によれば、グローバル意図言明及び複数のローカル意図言明のそれぞれを特定することは、内容の変化に1つ又は複数の動詞をタグ付けすることを含む。

【0670】

【00839】 実施態様例によれば、関係を受信することは、グローバル意図言明についてのローカル意図言明間の複数の別個の関係を受信することを含み、別個の各関係はソリューションを構築する別個の経路である。

【0671】

【00840】 実施態様例によれば、上記方法は、各ローカル意図言明の実体毎に、関連付けられたエージェントからプロセッサにより自然言語の形態で、実体のそれぞれに対する値を受信することを含み、実体のそれぞれに対する値を受信することは、受信した値に基づいて実体のそれぞれを潜在状態から現実状態に変えるイベントの記録である。

10

【0672】

【00841】 実施態様例によれば、上記方法は、各実体の属性毎に、関連付けられたエージェントからプロセッサにより自然言語の形態で、属性のそれぞれに対する値を受信することを含み、属性のそれぞれに対する値を受信することは、受信した値に基づいて属性のそれぞれを潜在状態から現実状態に変えるイベントの記録である。

【0673】

【00842】 実施態様例によれば、上記方法は、プロセッサにより、ログイン詳細、バイオメトリック詳細、顔認識技法、及び網膜検出技法の1つに基づいて関連するエージェントを認証することを含む。

20

【0674】

【00843】 実施態様例によれば、上記方法は、各ローカル意図言明に関連付けられた実体のそれぞれについてイベントが記録されると、プロセッサにより、ローカル意図言明のそれぞれを潜在状態から現実状態に変えるイベントを記録することを含む。

【0675】

【00844】 実施態様例によれば、上記方法は、各ローカル意図言明を潜在状態から現実状態に変えるイベントが記録されると、プロセッサにより、グローバル意図言明を潜在状態から現実状態に変えるイベントを記録することを含む。

【0676】

【00845】 実施態様例によれば、自然言語は、1つ若しくは複数の母国語、1つ若しくは複数の手話、1つ若しくは複数の記号、1つ若しくは複数の数字、又はそれらの組合せに基づく。

30

【0677】

【00846】 本趣旨は、実施態様例によれば、プログラミングコードを使用せずに、標準作業手順書(SOP)ファイル及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する方法に関し、SOPファイルは規範的言明、記述的言明、及びフローチャートを含み、方法は、計算デバイスのプロセッサにより、SOPファイルをパズすることであって、それにより、規範的言明、記述的言明、及びフローチャートを識別する、パズすることと、プロセッサにより、規範的言明のそれぞれについてそれぞれ1つずつ複数のローカル意図言明を特定し、ローカル意図言明のそれぞれを潜在状態に設定することであって、各ローカル意図言明はソリューションを構築するステップを示し、データベースに記憶された自然言語データに関して対応する規範的言明を分析することにより特定される、特定し設定することと、プロセッサにより、データベースに記憶された自然言語データに関して対応する規範的言明に関連付けられた記述的言明を分析することにより、複数のローカル意図言明のそれぞれに関連付けられた実体及び実体のそれぞれに関連付けられた属性を特定し、実体及び属性のそれぞれを潜在状態に設定することであって、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は実体のそれぞれの特徴を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定し設定することと、プロセッサにより、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体 - 状態

40

50

組合せ（CES）を形成することであって、ローカル意図言明のそれぞれの全ての实体に基づいて形成されるCESは、トリガー実体状態組合せ（トリガーCES）であり、1組中の各CESは潜在性の状態に設定される、形成することと、プロセッサにより、フローチャートに基づいてローカル意図言明間の関係を特定することであって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを特定することであって、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される、特定することと、ユーザからプロセッサにより自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定することであって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することを含む。

10

【0678】

【00847】 実施態様例によれば、複数のローカル意図言明を特定すること、実体及び属性を特定すること、並びに関係を特定することのそれぞれは、規範的言明、記述的言明、フローチャートの内容における発話部分を識別することと、内容中の同一指示、特に先行詞による代名詞を解決することと、継承関係をトラバースして、内容中の曖昧性をなくすことと、分析において曖昧性がある場合、内容の1つ又は複数の部分をマークすることと、1つ又は複数の曖昧な部分の曖昧さをなくすユーザ入力を受信することを含む。

【0679】

20

【00848】 実施態様例によれば、実体を特定することは、記述的言明中の名詞句に基づく。

【0680】

【00849】 実施態様例によれば、SOPファイルをパースすることは、分散ドキュメントパースシステムを使用しており、分散ドキュメントパースシステムはparse2Runである。

【0681】

【00850】 実施態様例によれば、parse2Runは、コアパス、参照パス、関係パス、理解パス、プロセスパス、エンリッチメントパス、及びコンプライアンスパスの少なくとも1つを含む。

30

【0682】

【00851】 実施態様例によれば、自然言語は、1つ若しくは複数の母国語、1つ若しくは複数の手話、1つ若しくは複数の記号、1つ若しくは複数の数字、又はそれらの組合せに基づく。

【0683】

【00852】 本趣旨は、実施態様例によれば、プログラミングコードを自然言語に変換して、自然言語ベースのコンピュータ実施ソリューションを構築する方法に関し、自然言語ベースのコンピュータ実施ソリューションは、自然言語ベースの入力を使用して実行可能であり、方法は、計算デバイスのプロセッサにより、プログラミングコードをパースすることであって、それにより、プログラミングコード中の記号、キーワード、演算子、及び関数を抽出する、パースすることと、プロセッサにより、データベースに記憶された自然言語データへの抽出された記号、キーワード、演算子、及び関数のマッピングに基づいて、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、及び実体のそれぞれに関連付けられた属性を特定し、複数のローカル意図言明、実体、及び属性のそれぞれを潜在状態に設定することであって、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定し設定することと、プロセッサにより、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体 - 状態組合せ（CES）を形成することであって、ローカル意図言明のそ

40

50

それぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、1組中の各CESは潜在性の状態に設定される、形成することと、プロセッサにより、データベースに記憶された自然言語データへの抽出された演算子及び関数のマッピングに基づいて、ローカル意図言明間の関係を特定することとであって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを特定し、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される、特定することと、ユーザからプロセッサにより自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定することとであって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することを含む。

10

【0684】

[00853] 実施態様例によれば、関係を受信することは、グローバル意図言明のローカル意図言明間の複数の別個の関係を受信することを含み、別個の各関係は、ソリューションを構築する別個の経路である。

【0685】

[00854] 実施態様例によれば、機械エージェントは、有線通信チャネル又は無線通信チャネルを介して計算デバイスに通信可能に結合される。

【0686】

[00855] 実施態様例によれば、グローバル意図言明のローカル意図言明に関連付けられた実体は、他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションのローカル意図言明又は他のコンピュータ実施ソリューションを借りる別のコンピュータ実施ソリューションの実体である。

20

【0687】

[00856] 実施態様例によれば、プログラミングコードは、バッチオペレーティングシステム、時間共有オペレーティングシステム、分散オペレーティングシステム、ネットワークオペレーティングシステム、及びリアルタイムオペレーティングシステムの少なくとも1つに対するものである。

【00857】

30

【0688】

[00858] 本趣旨は、実施態様例によれば、自然言語に基づいて構築されたコンピュータ実施ソリューションをプログラミングコードに変換する方法に関し、コンピュータ実施ソリューションは自然言語ベースの入力を使用して実行可能であり、方法は、計算デバイスのプロセッサにより、コンピュータ実施ソリューションをパースすることとであって、コンピュータ実施ソリューションは、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ(CES)、1つ又は複数のトリガーCESを含み、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化し、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであることを示す、パースすることと、計算デバイスのプロセッサにより、自然言語ソリューションコンポーネントを抽出することとであって、自然言語ソリューションコンポーネントは、グローバル意図言明、複数のローカル意図言明、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ(CES)、1つ又は複数のトリガーCESを含む、抽出することと、データベースに記憶されたプログラミング言語の記号、キーワード、演算子、及び関数への自然言語ソリューションコンポーネントのマッピングに基づいて、プログラミング言語コードを自動生成することを含む。

40

50

【0689】

【00859】 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、ユーザから、ソリューションを構築するグローバル意図言明を受信することであって、グローバル意図言明は、自然言語の形態で受信され、潜在性の状態で設定される、受信することと、ユーザから、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明並びに各ローカル意図言明に関連付けられた実体及びエージェントの詳細を受信することであって、各ローカル意図言明並びに各実体及びエージェントの詳細は、自然言語の形態で受信され、それぞれ潜在性の状態で設定され、各ローカル意図言明は、ソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは、人間エージェント及び機械エージェントの少なくとも一方である、受信することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体-状態組合せ(CES)を形成することであって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、1組中の各CESは潜在性の状態に設定される、形成することと、グローバル意図言明に関連付けられた、受信したローカル意図言明が1つのみであるとの判断に回答して、受信したローカル意図言明のトリガーCESをソリューションの構築の終了として識別することと、グローバル意図言明に関連付けられた、受信したローカル意図言明が2つ以上であるとの判断に回答して、ユーザから自然言語の形態で、ローカル意図言明間の関係を受信することであって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともソリューションの構築の終了であることを示す、関係を受信することを行うようにプロセッサにより実行可能な命令を含む。

10

20

【0690】

【00860】 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、ユーザから、ソリューションを構築するグローバル意図言明を受信する手段であって、グローバル意図言明は、自然言語の形態で受信され、潜在性の状態で設定される、受信する手段と、ユーザから、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明並びに各ローカル意図言明に関連付けられた実体及びエージェントの詳細を受信する手段であって、各ローカル意図言明並びに各実体及びエージェントの詳細は、自然言語の形態で受信され、それぞれ潜在性の状態で設定され、各ローカル意図言明は、ソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは、人間エージェント及び機械エージェントの少なくとも一方である、受信する手段と、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体-状態組合せ(CES)を形成する手段であって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、1組中の各CESは潜在性の状態に設定される、形成する手段と、グローバル意図言明に関連付けられた、受信したローカル意図言明が1つのみであるとの判断に回答して、受信したローカル意図言明のトリガーCESをソリューションの構築の終了として識別する手段と、グローバル意図言明に関連付けられた、受信したローカル意図言明が2つ以上であるとの判断に回答して、ユーザから自然言語の形態で、ローカル意図言明間の関係を受信する手段であって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともソリューションの構築の終了であることを示す、受信する手段とを含む。

30

40

【0691】

【00861】 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デ

50

デバイスに関し、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、上記方法を実行するようにプロセッサにより実行可能な命令を含む。

【0692】

[00862] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、自然言語の形態でグローバル意図言明を入力するプロンプトをユーザに提供し、ソリューションを構築するグローバル意図言明を潜在状態に設定することと、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明と、各ローカル意図言明に関連付けられた実体及びエージェントの詳細とを入力するプロンプトをユーザに自然言語の形態で提供し、1つ又は複数のローカル意図言明及び実体のそれぞれを潜在状態に設定することと、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、提供し設定することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の全ての可能な組合せに基づいて1組の実体 - 状態組合せ (CES) を形成することと、ローカル意図言明のそれぞれの全ての実体に基づいて形成されたCESは、トリガー実体状態組合せ (トリガーCES) であり、組中の各CESは潜在状態である、形成することと、グローバル意図言明に関連付けられた、受信されたローカル意図言明が1つのみであるとの判断に回答して、受信したローカル意図言明のトリガーCESをソリューションの構築の終わりとして識別することと、グローバル意図言明に関連付けられた、受信されたローカル意図言明が2つ以上であるとの判断に回答して、ローカル意図言明間の関係を入力するプロンプトをユーザに自然言語の形態で提供することと、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともソリューションの構築の終わりであることを示す、提供することとを行うようにプロセッサにより実行可能な命令を含む。

10

20

【0693】

[00863] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、自然言語の形態でグローバル意図言明を入力するプロンプトをユーザに提供し、ソリューションを構築するグローバル意図言明を潜在状態に設定する手段と、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明と、各ローカル意図言明に関連付けられた実体及びエージェントの詳細とを入力するプロンプトをユーザに自然言語の形態で提供し、1つ又は複数のローカル意図言明及び実体のそれぞれを潜在状態に設定する手段と、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、提供し設定する手段と、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の全ての可能な組合せに基づいて1組の実体 - 状態組合せ (CES) を形成する手段と、ローカル意図言明のそれぞれの全ての実体に基づいて形成されたCESは、トリガー実体状態組合せ (トリガーCES) であり、組中の各CESは潜在状態である、形成する手段と、グローバル意図言明に関連付けられた、受信されたローカル意図言明が1つのみであるとの判断に回答して、受信したローカル意図言明のトリガーCESをソリューションの構築の終わりとして識別する手段と、グローバル意図言明に関連付けられた、受信されたローカル意図言明が2つ以上であるとの判断に回答して、ローカル意図言明間の関係を入力するプロンプトをユーザに自然言語の形態で提供する手段と、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともソリューションの構築の終わりであることを示す、提供する手段とを含む。

30

40

【0694】

[00864] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザ

50

により理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、上記方法を実行するようにプロセッサにより実行可能な命令を含む。

【0695】

[00865] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、動画及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、動画の画像フレームを処理することであって、それにより、複数の差別化された画像フレームを識別し、分化した各画像フレームは、その内容の違いに基づいて動画の1つ又は複数の先の画像フレームと異なるものとして識別される、処理することと、それぞれ動画の直前の画像フレームの内容に関する差別化された各画像フレームの内容の差に基づいて、複数のローカル意図言明を特定し、ローカル意図言明のそれぞれを潜在状態に設定することであって、各ローカル意図言明は、ソリューションを構築するステップを示し、データベースに記憶された自然言語データに関して画像フレームの内容を分析することにより特定される、特定し設定することと、データベースに記憶された自然言語データに関して差別化された画像フレームの連続対間の画像フレームの内容を分析することにより、複数のローカル意図言明のそれぞれに関連付けられた実体及び実体のそれぞれに関連付けられた属性を特定することであって、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定することと、最初の差別化された画像フレームと最後の差別化された画像フレームの内容の変化に基づいて、コンピュータ実施ソリューションのグローバル意図言明を特定し、グローバル意図言明を潜在状態に設定することであって、グローバル意図言明は、データベースに記憶された自然言語データに関して、最初の差別化された画像フレーム及び最後の差別化された画像フレームの内容を分析することにより特定される、特定し設定することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の全ての可能な組合せに基づいて1組の実体状態組合せ(CES)を形成することであって、ローカル意図言明のそれぞれの実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、組中の各CESは潜在状態である、形成することと、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定することであって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することと、ユーザから自然言語の形態で、ローカル意図言明間の関係を受信することであって、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであるかを示す、受信することとを行うようにプロセッサにより実行可能な命令を含む。

【0696】

[00866] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、動画及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、計算デバイスは、動画の画像フレームを処理する手段であって、それにより、複数の差別化された画像フレームを識別し、分化した各画像フレームは、その内容の違いに基づいて動画の1つ又は複数の先の画像フレームと異なるものとして識別される、処理する手段と、それぞれ動画の直前の画像フレームの内容に関する差別化された各画像フレームの内容の差に基づいて、複数のローカル意図言明を特定し、ローカル意図言明のそれぞれを潜在状態に設定する手段であって、各ローカル意図言明は、ソリューションを構築するステップを示し、データベースに記憶された自然言語データに関して画像フレームの内容を分析することにより特定される、特定し設定する手段と、データベースに記憶された自然言語データに関して差別化された画像フレームの連続対間の画像フレームの内容を分析することにより、複数のローカル意図言明のそれぞれに関連付けられた実体及び実体のそれぞれに関連付けられた属性を特定する手段であって、各実体は、対応

10

20

30

40

50

するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定する手段と、最初の差別化された画像フレームと最後の差別化された画像フレームの内容の変化に基づいて、コンピュータ実施ソリューションのグローバル意図言明を特定し、グローバル意図言明を潜在状態に設定する手段であって、グローバル意図言明は、データベースに記憶された自然言語データに関して、最初の差別化された画像フレーム及び最後の差別化された画像フレームの内容を分析することにより特定される、特定し設定する手段と、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の全ての可能な組合せに基づいて1組の実体状態組合せ(CES)を形成する手段であって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、組中の各CESは潜在状態である、形成する手段と、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定する手段であって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定する手段と、ユーザから自然言語の形態で、ローカル意図言明間の関係を受信する手段であって、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであることを示す、受信する手段とを含む。

10

【0697】

【00867】 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、動画及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、上記方法を実行するようにプロセッサにより実行可能な命令を含む。

20

【0698】

【00868】 本趣旨は、実施態様例によれば、プログラミングコードを使用せずに、標準作業手順書(SOP)ファイル及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、SOPファイルは規範的言明、記述的言明、及びフローチャートを含み、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、SOPファイルをパズすることであって、それにより、規範的言明、記述的言明、及びフローチャートを識別する、パズすることと、規範的言明のそれぞれについてそれぞれ1つずつ複数のローカル意図言明を特定し、ローカル意図言明のそれぞれを潜在状態に設定することであって、各ローカル意図言明はソリューションを構築するステップを示し、データベースに記憶された自然言語データに関して対応する規範的言明を分析することにより特定される、特定し設定することと、データベースに記憶された自然言語データに関して対応する規範的言明に関連付けられた記述的言明を分析することにより、複数のローカル意図言明のそれぞれに関連付けられた実体及び実体のそれぞれに関連付けられた属性を特定し、実体及び属性のそれぞれを潜在状態に設定することであって、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定し設定することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体-状態組合せ(CES)を形成することであって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、1組中の各CESは潜在性の状態に設定される、形成することと、フローチャートに基づいてローカル意図言明間の関係を特定することであって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであることを特定することであって、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される、特定することと、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を

30

40

50

潜在状態に設定することによって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することを行うようにプロセッサにより実行可能な命令を含む。

【0699】

[00869] 本趣旨は、実施態様例によれば、プログラミングコードを使用せずに、標準作業手順書(SOP)ファイル及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、SOPファイルは規範的言明、記述的言明、及びフローチャートを含み、計算デバイスは、計算デバイスのプロセッサにより、SOPファイルをパースする手段であって、それにより、規範的言明、記述的言明、及びフローチャートを識別する、パースする手段と、規範的言明のそれぞれについてそれぞれ1つずつ複数のローカル意図言明を特定し、ローカル意図言明のそれぞれを潜在状態に設定する手段であって、各ローカル意図言明はソリューションを構築するステップを示し、データベースに記憶された自然言語データに関して対応する規範的言明を分析することにより特定される、特定し設定する手段と、データベースに記憶された自然言語データに関して対応する規範的言明に関連付けられた記述的言明を分析することにより、複数のローカル意図言明のそれぞれに関連付けられた実体及び実体のそれぞれに関連付けられた属性を特定し、実体及び属性のそれぞれを潜在状態に設定する手段であって、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は実体のそれぞれの特徴を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定し設定する手段と、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体-状態組合せ(CES)を形成する手段であって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、1組中の各CESは潜在性の状態に設定される、形成する手段と、フローチャートに基づいてローカル意図言明間の関係を特定する手段であって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを特定する手段であって、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される、特定する手段と、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定する手段であって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信する手段とを含む。

【0700】

[00870] 本趣旨は、実施態様例によれば、プログラミングコードを使用せずに、標準作業手順書(SOP)ファイル及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する計算デバイスに関し、SOPファイルは規範的言明、記述的言明、及びフローチャートを含み、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、上記方法を実行するようにプロセッサにより実行可能な命令を含む。

【0701】

[00871] 本趣旨は、実施態様例によれば、プログラミングコードを自然言語に変換して、自然言語ベースのコンピュータ実施ソリューションを構築する計算デバイスに関し、自然言語ベースのコンピュータ実施ソリューションは、自然言語ベースの入力を使用して実行可能であり、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、プログラミングコードをパースすることによって、それにより、プログラミングコード中の記号、キーワード、演算子、及び関数を抽出する、パースすることと、データベースに記憶された自然言語データへの抽出された記号、キーワード、演算子、及び関数のマッピングに基づいて、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、及び実体のそれぞれに関連付けられた属性を特定し、複数のローカル意図言明、実体、及び属性のそれぞれを潜在状態に設定することによって、各口

10

20

30

40

50

ーカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定し設定することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体 - 状態組合せ (CES) を形成することであって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ (トリガーCES) であり、1組中の各CESは潜在性の状態に設定される、形成することと、データベースに記憶された自然言語データへの抽出された演算子及び関数のマッピングに基づいて、ローカル意図言明間の関係を特定することであって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを特定し、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される、特定することと、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定することであって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することとを行うようにプロセッサにより実行可能な命令を含む。

10

【0702】

[00872] 本趣旨は、実施態様例によれば、プログラミングコードを自然言語に変換して、自然言語ベースのコンピュータ実施ソリューションを構築する計算デバイスに関し、自然言語ベースのコンピュータ実施ソリューションは、自然言語ベースの入力を使用して実行可能であり、計算デバイスは、プログラミングコードをパースする手段であって、それにより、プログラミングコード中の記号、キーワード、演算子、及び関数を抽出する、パースする手段と、データベースに記憶された自然言語データへの抽出された記号、キーワード、演算子、及び関数のマッピングに基づいて、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、及び実体のそれぞれに関連付けられた属性を特定し、複数のローカル意図言明、実体、及び属性のそれぞれを潜在状態に設定する手段であって、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定し設定する手段と、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体 - 状態組合せ (CES) を形成する手段であって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ (トリガーCES) であり、1組中の各CESは潜在性の状態に設定される、形成する手段と、データベースに記憶された自然言語データへの抽出された演算子及び関数のマッピングに基づいて、ローカル意図言明間の関係を特定する手段であって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを特定し、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される、特定する手段と、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定する手段であって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することとを含む。

20

30

40

【0703】

[00873] 本趣旨は、実施態様例によれば、プログラミングコードを自然言語に変換して、自然言語ベースのコンピュータ実施ソリューションを構築する計算デバイスに関し、自然言語ベースのコンピュータ実施ソリューションは、自然言語ベースの入力を使用して実行可能であり、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、上記方法を実行するようにプロセッサにより実行可能である。

50

【0704】

【00874】 本趣旨は、実施態様例によれば、自然言語に基づいて構築されたコンピュータ実施ソリューションをプログラミングコードに変換する計算デバイスに関し、コンピュータ実施ソリューションは自然言語ベースの入力を使用して実行可能であり、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、コンピュータ実施ソリューションをパズすることであって、コンピュータ実施ソリューションは、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ（CES）、1つ又は複数のトリガーCESを含み、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化し、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであることを示す、パズすることと、自然言語ソリューションコンポーネントを抽出することであって、自然言語ソリューションコンポーネントは、グローバル意図言明、複数のローカル意図言明、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ（CES）、1つ又は複数のトリガーCESを含む、抽出することと、データベースに記憶されたプログラミング言語の記号、キーワード、演算子、及び関数への自然言語ソリューションコンポーネントのマッピングに基づいて、プログラミング言語コードを自動生成することを行うようにプロセッサにより実行可能な命令を含む。

10

20

【0705】

【00875】 本趣旨は、実施態様例によれば、自然言語に基づいて構築されたコンピュータ実施ソリューションをプログラミングコードに変換する計算デバイスであって、コンピュータ実施ソリューションは自然言語ベースの入力を使用して実行可能であり、計算デバイスは、コンピュータ実施ソリューションをパズする手段であって、コンピュータ実施ソリューションは、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ（CES）、1つ又は複数のトリガーCESを含み、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化し、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであることを示す、パズする手段と、自然言語ソリューションコンポーネントを抽出する手段であって、自然言語ソリューションコンポーネントは、グローバル意図言明、複数のローカル意図言明、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ（CES）、1つ又は複数のトリガーCESを含む、抽出する手段と、データベースに記憶されたプログラミング言語の記号、キーワード、演算子、及び関数への自然言語ソリューションコンポーネントのマッピングに基づいて、プログラミング言語コードを自動生成する手段とを含む。

30

40

【0706】

【00876】 本趣旨は、実施態様例によれば、自然言語に基づいて構築されたコンピュータ実施ソリューションをプログラミングコードに変換する計算デバイスに関し、コンピュータ実施ソリューションは自然言語ベースの入力を使用して実行可能であり、計算デバイスは、プロセッサと、プロセッサに結合されたメモリとを含み、方法は、上記方法を実行するようにプロセッサにより実行可能である命令を含む。

【0707】

【00877】 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する命令を記憶した非一時的コンピュータ可読媒体に関し、命令は機械実行可能コードを含み、機械

50

実行可能コードは、プロセッサにより実行されると、プロセッサに、ユーザから、ソリューションを構築するグローバル意図言明を受信することであって、グローバル意図言明は、自然言語の形態で受信され、潜在性の状態で設定される、受信することと、ユーザから、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明並びに各ローカル意図言明に関連付けられた実体及びエージェントの詳細を受信することであって、各ローカル意図言明並びに各実体及びエージェントの詳細は、自然言語の形態で受信され、それぞれ潜在性の状態で設定され、各ローカル意図言明は、ソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは、人間エージェント及び機械エージェントの少なくとも一方である、受信することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体 - 状態組合せ (CES) を形成することであって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ (トリガーCES) であり、1組中の各CESは潜在性の状態に設定される、形成することと、グローバル意図言明に関連付けられた、受信したローカル意図言明が1つのみであるとの判断に回答して、受信したローカル意図言明のトリガーCESをソリューションの構築の終了として識別することと、グローバル意図言明に関連付けられた、受信したローカル意図言明が2つ以上であるとの判断に回答して、ユーザから自然言語の形態で、ローカル意図言明間の関係を受信することであって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともソリューションの構築の終了であることを示す、関係を受信することとを行わせる。

10

20

【0708】

[00878] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する命令を記憶した非一時的コンピュータ可読媒体に関し、命令は機械実行可能コードを含み、機械実行可能コードは、プロセッサにより実行されると、プロセッサに、自然言語の形態でグローバル意図言明を入力するプロンプトをユーザに提供し、ソリューションを構築するグローバル意図言明を潜在状態に設定することと、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明と、各ローカル意図言明に関連付けられた実体及びエージェントの詳細とを入力するプロンプトをユーザに自然言語の形態で提供し、1つ又は複数のローカル意図言明及び実体のそれぞれを潜在状態に設定することであって、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、提供し設定することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の全ての可能な組合せに基づいて1組の実体 - 状態組合せ (CES) を形成することであって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されたCESは、トリガー実体状態組合せ (トリガーCES) であり、組中の各CESは潜在状態である、形成することと、グローバル意図言明に関連付けられた、受信されたローカル意図言明が1つのみであるとの判断に回答して、受信したローカル意図言明のトリガーCESをソリューションの構築の終わりとして識別することと、グローバル意図言明に関連付けられた、受信されたローカル意図言明が2つ以上であるとの判断に回答して、ローカル意図言明間の関係を入力するプロンプトをユーザに自然言語の形態で提供することであって、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともソリューションの構築の終わりであることを示す、提供することとを行わせる。

30

40

【0709】

[00879] 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、動画及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する命令を記憶した非一時的コンピュータ可読媒体に関し、命令は機械実行可能コードを含み、機械実行可能コードは、プロセッサにより実行されると、プロセッサに、動画の画像フレームを処理することであって、それにより、複数の差別化された画像フレームを識別

50

し、分化した各画像フレームは、その内容の違いに基づいて動画の1つ又は複数の先の画像フレームと異なるものとして識別される、処理することと、それぞれ動画の直前の画像フレームの内容に関する差別化された各画像フレームの内容の差に基づいて、複数のローカル意図言明を特定し、ローカル意図言明のそれぞれを潜在状態に設定することによって、各ローカル意図言明は、ソリューションを構築するステップを示し、データベースに記憶された自然言語データに関して画像フレームの内容を分析することにより特定される、特定し設定することと、データベースに記憶された自然言語データに関して差別化された画像フレームの連続対間の画像フレームの内容を分析することにより、複数のローカル意図言明のそれぞれに関連付けられた実体及び実体のそれぞれに関連付けられた属性を特定することによって、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定することと、最初の差別化された画像フレームと最後の差別化された画像フレームの内容の変化に基づいて、コンピュータ実施ソリューションのグローバル意図言明を特定し、グローバル意図言明を潜在状態に設定することによって、グローバル意図言明は、データベースに記憶された自然言語データに関して、最初の差別化された画像フレーム及び最後の差別化された画像フレームの内容を分析することにより特定される、特定し設定することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の全ての可能な組合せに基づいて1組の実体状態組合せ(CES)を形成することによって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、組中の各CESは潜在状態である、形成することと、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定することによって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することと、ユーザから自然言語の形態で、ローカル意図言明間の関係を受信することによって、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであることを示す、受信することとを行わせる。

10

20

【0710】

[00880] 本趣旨は、実施態様例によれば、プログラミングコードを使用せずに、標準作業手順書(SOP)ファイル及びユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築する命令を記憶した非一時的コンピュータ可読媒体に関し、SOPファイルは規範的言明、記述的言明、及びフローチャートを含み、命令は機械実行可能コードを含み、機械実行可能コードは、プロセッサにより実行されると、プロセッサに、SOPファイルをパースすることによって、それにより、規範的言明、記述的言明、及びフローチャートを識別する、パースすることと、規範的言明のそれぞれについてそれぞれ1つずつ複数のローカル意図言明を特定し、ローカル意図言明のそれぞれを潜在状態に設定することによって、各ローカル意図言明はソリューションを構築するステップを示し、データベースに記憶された自然言語データに関して対応する規範的言明を分析することにより特定される、特定し設定することと、データベースに記憶された自然言語データに関して対応する規範的言明に関連付けられた記述的言明を分析することにより、複数のローカル意図言明のそれぞれに関連付けられた実体及び実体のそれぞれに関連付けられた属性を特定し、実体及び属性のそれぞれを潜在状態に設定することによって、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定し設定することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体-状態組合せ(CES)を形成することによって、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、1組中の各CESは潜在性の状態に設定される、形成することと、フローチャートに基づいてローカル意図言明間の関係を特定することによって、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明の

30

40

50

C E Sの組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを特定することによって、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される、特定することと、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定することによって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することとを行わせる。

【0711】

[00881] 本趣旨は、実施態様例によれば、プログラミングコードを自然言語に変換して、自然言語ベースのコンピュータ実施ソリューションを構築する命令を記憶した非一時的コンピュータ可読媒体に関し、自然言語ベースのコンピュータ実施ソリューションは、自然言語ベースの入力を使用して実行可能であり、命令は機械実行可能コードを含み、機械実行可能コードは、プロセッサにより実行されると、プロセッサに、プログラミングコードをパズすることによって、それにより、プログラミングコード中の記号、キーワード、演算子、及び関数を抽出する、パズすることと、データベースに記憶された自然言語データへの抽出された記号、キーワード、演算子、及び関数のマッピングに基づいて、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、及び実体のそれぞれに関連付けられた属性を特定し、複数のローカル意図言明、実体、及び属性のそれぞれを潜在状態に設定することによって、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化する、特定し設定することと、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体 - 状態組合せ (C E S) を形成することによって、ローカル意図言明のそれぞれの全ての実体に基づいて形成される C E S は、トリガー実体状態組合せ (トリガー C E S) であり、1組中の各 C E S は潜在性の状態に設定される、形成することと、データベースに記憶された自然言語データへの抽出された演算子及び関数のマッピングに基づいて、ローカル意図言明間の関係を特定することによって、関係は、あるローカル意図言明のトリガー C E S が、別のローカル意図言明の C E S の組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを特定し、関係は、データベースに記憶された自然言語データに関してフローチャートを分析することにより特定される、特定することと、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定することによって、エージェントは人間エージェント及び機械エージェントの少なくとも一方である、受信し設定することとを行わせる。

【0712】

[00882] 本趣旨は、実施態様例によれば、自然言語に基づいて構築されたコンピュータ実施ソリューションをプログラミングコードに変換する命令を記憶した非一時的コンピュータ可読媒体に関し、コンピュータ実施ソリューションは自然言語ベースの入力を使用して実行可能であり、命令は機械実行可能コードを含み、機械実行可能コードは、プロセッサにより実行されると、プロセッサに、コンピュータ実施ソリューションをパズすることによって、コンピュータ実施ソリューションは、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ (C E S) 、 1つ又は複数のトリガー C E S を含み、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化し、関係は、あるローカル意図言明のトリガー C E S が別のローカル意図言明の C E S の組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであることを示す、パズすることと、自然言語ソリューションコンポーネントを抽出することであ

10

20

30

40

50

って、自然言語ソリューションコンポーネントは、グローバル意図言明、複数のローカル意図言明、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ（CES）、1つ又は複数のトリガーCESを含む、抽出することと、データベースに記憶されたプログラミング言語の記号、キーワード、演算子、及び関数への自然言語ソリューションコンポーネントのマッピングに基づいて、プログラミング言語コードを自動生成することとを行わせる。

【0713】

【00883】 本趣旨は、実施態様例によれば、プログラミングコードを使用せず、ユーザにより理解される自然言語を使用してコンピュータ実施ソリューションを構築するシステムに関し、システムは、プロセッサと、プロセッサに結合されたグローバル意図言明モジュール、ローカル意図言明モジュール、CESモジュール、及び関係モジュールとを含み、グローバル意図言明モジュール、ローカル意図言明モジュール、CESモジュール、及び関係モジュールは互いに動作可能に接続され、グローバル意図言明モジュールは、ユーザから、ソリューションを構築するグローバル意図言明を受信するように構成され、グローバル意図言明は、自然言語の形態で受信され、潜在性の状態で設定され、ローカル意図言明モジュールは、ユーザから、グローバル意図言明に関連付けられた1つ又は複数のローカル意図言明並びに各ローカル意図言明に関連付けられた実体及びエージェントの詳細を受信するよう構成され、各ローカル意図言明並びに各実体及びエージェントの詳細は、自然言語の形態で受信され、それぞれ潜在性の状態で設定され、各ローカル意図言明は、ソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、エージェントは、人間エージェント、機械エージェント、及びコンピュータ機能の少なくとも1つであり、CESモジュールは、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体-状態組合せ（CES）を形成するように構成され、ローカル意図言明のそれぞれの実体に基づいて形成されるCESは、トリガー実体状態組合せ（トリガーCES）であり、1組中の各CESは潜在性の状態で設定され、グローバル意図言明に関連付けられた、受信したローカル意図言明が1つのみであるとの判断に応答して、受信したローカル意図言明のトリガーCESをソリューションの構築の終了として識別し、関係モジュールは、ユーザから自然言語の形態で、ローカル意図言明間の関係を受信するように構成され、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともソリューションの構築の終了であることを示す。

【0714】

【00884】 本趣旨は、実施態様例によれば、プログラミングコードを自然言語に変換して、自然言語ベースのコンピュータ実施ソリューションを構築するシステムに関し、自然言語ベースのコンピュータ実施ソリューションは、自然言語ベースの入力を使用して実行可能であり、システムは、プロセッサと、プロセッサに結合されたパーサモジュール、抽出機モジュール、インタプリタモジュール、マッピングモジュール、グローバル意図言明モジュール、ローカル意図言明モジュール、CESモジュール、及び関係モジュールとを含み、パーサモジュール、抽出機モジュール、インタプリタモジュール、マッピングモジュール、グローバル意図言明モジュール、ローカル意図言明モジュール、CESモジュール、及び関係モジュールは、互いに動作可能に接続され、パーサモジュールは、プログラミングコードをパズして、プログラミングコード中の記号、キーワード、演算子、及び関数を抽出するように構成され、抽出機モジュール、インタプリタモジュール、マッピングモジュール、グローバル意図言明モジュール、及び/又はローカル意図言明モジュールは、データベースに記憶された自然言語データへの抽出された記号、キーワード、演算子、及び関数のマッピングに基づいて、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、及び実体のそれぞれに関連付けられた属性を特定し、複数のローカル意図言明、実体、及び属性のそれぞれを潜在状態に設定するように構成され、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を

定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化し、CESモジュールは、ローカル意図言明毎に、ローカル意図言明のそれぞれの実体の可能な全ての組合せに基づいて1組の実体-状態組合せ(CES)を形成するように構成され、ローカル意図言明のそれぞれの全ての実体に基づいて形成されるCESは、トリガー実体状態組合せ(トリガーCES)であり、1組中の各CESは潜在性の状態に設定され、関係モジュールは、データベースに記憶された自然言語データへの抽出された演算子及び関数のマッピングに基づいて、ローカル意図言明間の関係を特定するように構成され、関係は、あるローカル意図言明のトリガーCESが、別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築のグローバル意図言明の終わりであるかを特定し、関係は、データベースに記憶された自然言語データに関してフローチャート进行分析することにより特定され、ローカル意図言明モジュールは、ユーザから自然言語の形態で、複数のローカル意図言明のそれぞれに関連付けられたエージェントの詳細を受信し、エージェントの詳細を潜在状態に設定するように構成され、エージェントは人間エージェント及び機械エージェントの少なくとも一方である。

10

【0715】

[00885] 本趣旨は、実施態様例によれば、自然言語に基づいて構築されたコンピュータ実施ソリューションをプログラミングコードに変換するシステムであって、コンピュータ実施ソリューションは自然言語ベースの入力を使用して実行可能であり、システムは、プロセッサと、プロセッサに結合されたソリューションパーサモジュール、ソリューション抽出機モジュール、自動生成器モジュール、及びマッピングモジュールとを含み、ソリューションパーサモジュール、ソリューション抽出機モジュール、自動生成器モジュール、及びマッピングモジュールは、互いに動作可能に接続され、ソリューションパーサモジュールは、コンピュータ実施ソリューションをパズするように構成され、コンピュータ実施ソリューションは、複数のローカル意図言明、複数のローカル意図言明のそれぞれに関連付けられた実体、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ(CES)、1つ又は複数のトリガーCESを含み、各ローカル意図言明はソリューションを構築するステップを示し、各実体は、対応するローカル意図言明により示されるステップに参加し、属性は、実体のそれぞれの特性を定義し、対応するローカル意図言明の他の実体から実体のそれぞれを差別化し、関係は、あるローカル意図言明のトリガーCESが別のローカル意図言明のCESの組に影響しているか、それともコンピュータ実施ソリューションの構築の終わりであることを示し、ソリューション抽出機モジュールは、自然言語ソリューションコンポーネントを抽出するように構成され、自然言語ソリューションコンポーネントは、グローバル意図言明、複数のローカル意図言明、実体のそれぞれに関連付けられた属性、複数のローカル意図言明間の関係、実体状態組合せ(CES)、1つ又は複数のトリガーCESを含み、自動生成器モジュールは、データベースに記憶されたプログラミング言語の記号、キーワード、演算子、及び関数への自然言語ソリューションコンポーネントのマッピングに基づいて、プログラミング言語コードを自動生成するように構成される。

20

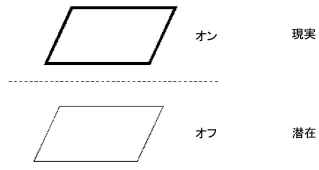
30

40

50

【図面】

【図 1】



【図 2】

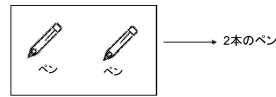


10

【図 3】



【図 4】



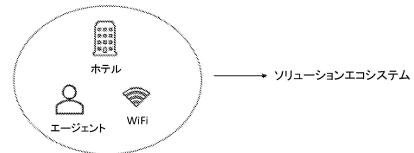
20

【図 5】

部屋を予約

部屋が空いているかを確認。顧客情報を取得。客室宿泊情報を取得。
 トランスポート情報を取得。ランドリー情報を取得。価格を取得。部屋を予約。

【図 6】



30

40

50

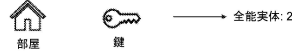
【 図 7 】

部屋を予約

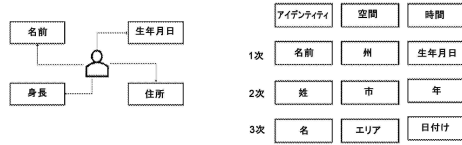
部屋が空いているかをチェック。顧客情報を取得。客室宿泊情報を取得。交通機関情報を取得。
ランドリー情報を取得。場所を取得。ホテルの部屋を予約。

備考：太字及び下線は現実状態を表す。NSLでは、赤色である。

【 図 8 】

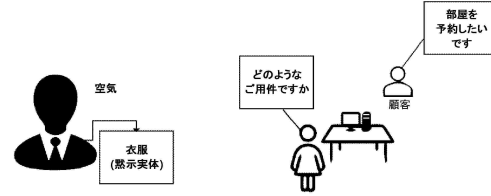


【 図 9 】



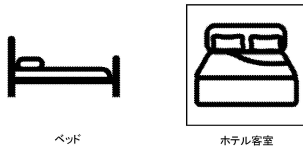
【 図 1 0 】

10



【 図 1 1 】

凍結実体:



【 図 1 2 】

20



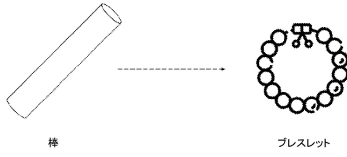
30

40

50

【 図 1 3 】

実体の変更:



【 図 1 4 】



	アイデンティティ	空間	時間	金額
1次	部屋番号	テランガーナ州	日付で	価格
2次		ハイデラバード市	時刻で	
3次		ジュペリー ヘルズ		

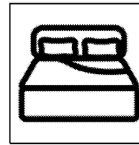
【 図 1 5 】



	アイデンティティ	空間	時間
1次	部屋番号	テランガーナ州	日付で
2次		ハイデラバード市	時刻で
3次		ジュペリー ヘルズ	

【 図 1 6 】

例: 全てのホテル客室は以下の標準属性を有することになる。価格、Wi-Fi



【 図 1 7 】



	アイデンティティ	空間	時間
1次	TS 9999	テランガーナ州	購入日
2次	マルチ	ハイデラバード市	初回サービス 利用日
3次	型式:1234		

【 図 1 8 】



	アイデンティティ	空間
1次	ホリデイン	テランガーナ州
2次	ブロック5	ハイデラバード市
3次	部屋番号: 510	金融街

10

20

30

40

50

【図 19】



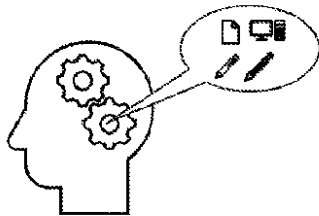
【図 20】



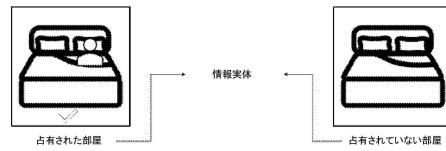
10

【図 21】

Drawing 21:



【図 22】



20

【図 23】

12月

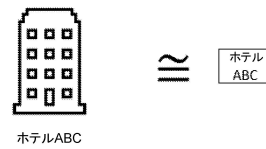
日	月	火	水	木	金	土
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

例:
12月18日顧客からの
徴収

徴収総額 = 2000インドルピー



【図 24】

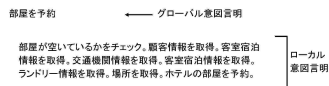


30

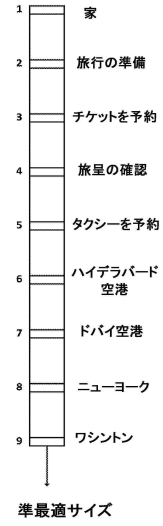
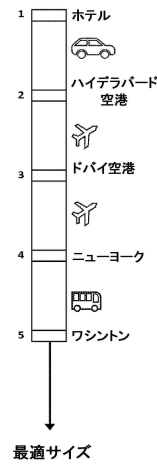
40

50

【 図 2 5 】



【 図 2 6 】



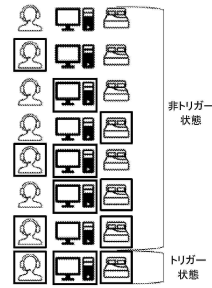
10

20

【 図 2 7 】



【 図 2 8 】



30

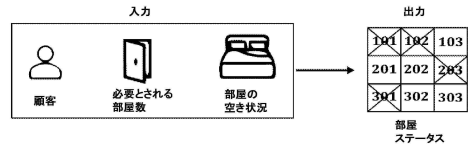
40

50

【 図 29 】



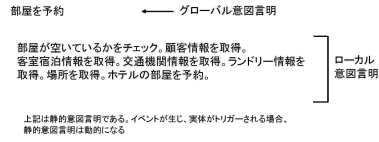
【 図 30 】



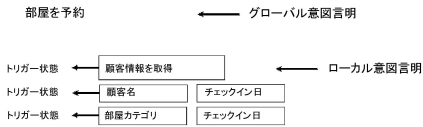
【 図 31 】



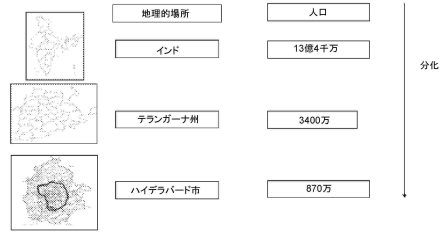
【 図 32 】



【 図 33 】



【 図 34 】



10

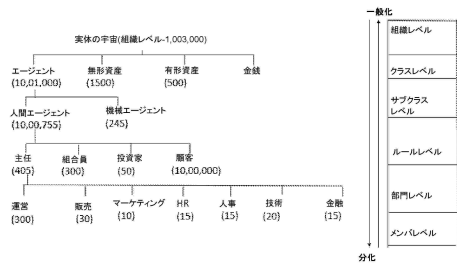
20

30

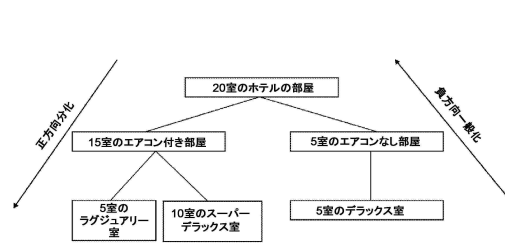
40

50

【図 3 5】

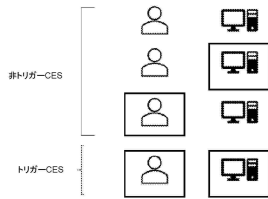


【図 3 6】

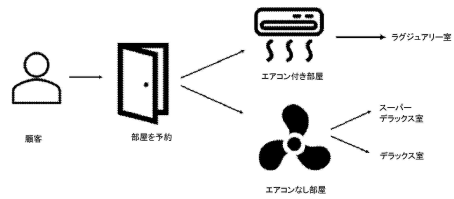


10

【図 3 7】

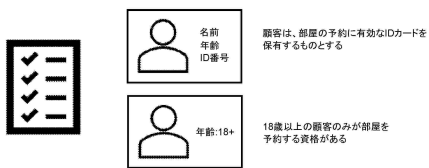


【図 3 8】

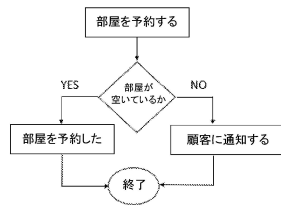


20

【図 3 9】



【図 4 0】



30

40

50

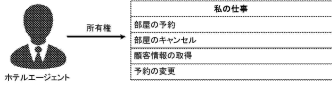
【 図 4 1 】



【 図 4 2 】



【 図 4 3 】



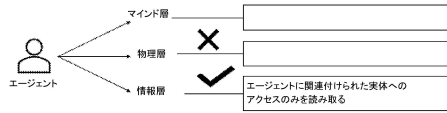
【 図 4 4 】

物理的機能	部屋の予約 部屋の空き状況のチェック、 顧客情報の取得、 客室宿泊情報を取得、 価格の取得、部屋の予約									
マインド機能	過去の部屋の空き状況									
情報機能	<table border="1"> <tr> <td>迎えている部屋</td> <td>デラックス</td> <td>Wi-Fiホスト、LED TV、 サイズ:200平方フィート</td> </tr> <tr> <td>ダブルデラックス</td> <td>スーパーデラックス</td> <td>フリーWi-Fi、LCD TV、 サイズ:250平方フィート</td> </tr> <tr> <td>ラグジュアリー</td> <td>ラグジュアリー</td> <td>フリーWi-Fi、65インチLCD TV、 サイズ:300平方フィート</td> </tr> </table>	迎えている部屋	デラックス	Wi-Fiホスト、LED TV、 サイズ:200平方フィート	ダブルデラックス	スーパーデラックス	フリーWi-Fi、LCD TV、 サイズ:250平方フィート	ラグジュアリー	ラグジュアリー	フリーWi-Fi、65インチLCD TV、 サイズ:300平方フィート
迎えている部屋	デラックス	Wi-Fiホスト、LED TV、 サイズ:200平方フィート								
ダブルデラックス	スーパーデラックス	フリーWi-Fi、LCD TV、 サイズ:250平方フィート								
ラグジュアリー	ラグジュアリー	フリーWi-Fi、65インチLCD TV、 サイズ:300平方フィート								

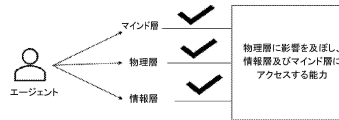
10

20

【 図 4 5 】



【 図 4 6 】



30

40

50

【 図 4 7 】



【 図 4 8 】

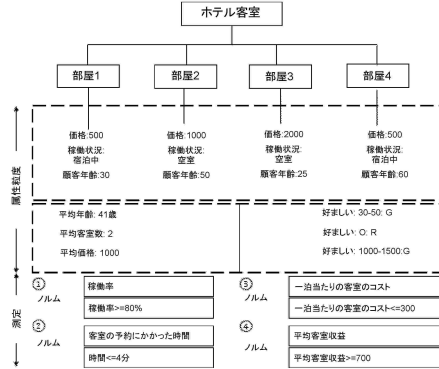


【 図 4 9 】

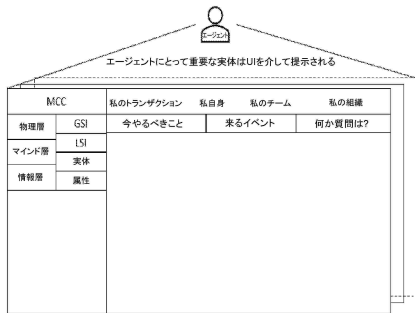
部屋を予約 ← グローバル意図説明

段階I- 部屋が空いているかチェック。顧客情報を取得。客室宿泊情報を取得。場所を取得。ホテル客室を予約
段階II- 部屋が空いているかチェック。顧客情報を取得。客室宿泊情報を取得。交通機関情報を取得。ランドリー情報を取得。場所を取得。ホテル客室を予約

【 図 5 0 】



【 図 5 1 】



【 図 5 2 】



10

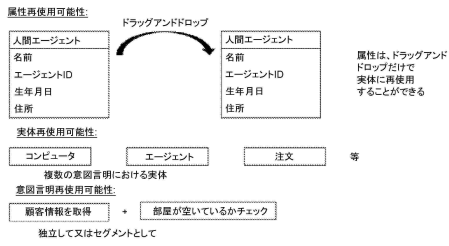
20

30

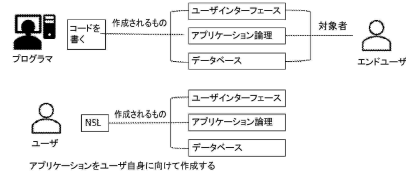
40

50

【図 5 3】

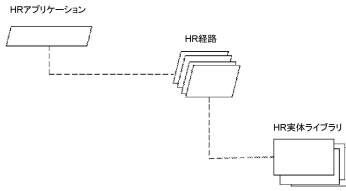


【図 5 4】

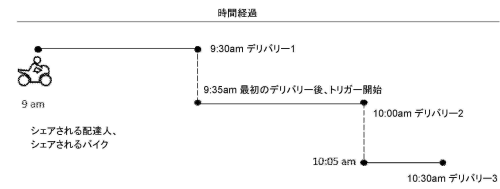


10

【図 5 5】

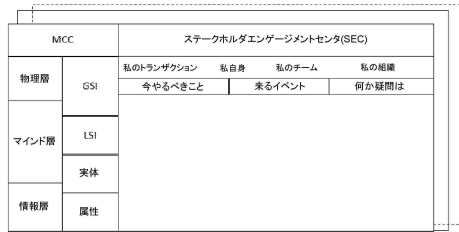


【図 5 6】



20

【図 5 7】



【図 5 8】

Javaコード:

パッケージ com.Hotel 予約

```

Public class fare{
    public int fare calculator
    (int rate, int days, int s)
    {
        if (s==1)
        {
            return days * rate;
        }
        if (s==2)
        {
            return days * rate * 2;
        }
    }
}

```

自然言語:

Fare calculator:
 If occupancy = 1, fare is equal
 To number of days* rate
 If occupancy = 2, fare is equal
 To number of days * rate *2

30

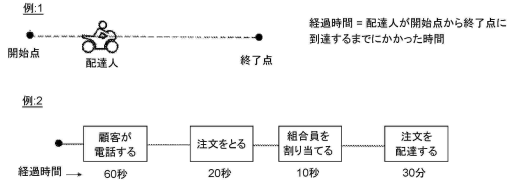
40

50

【 図 5 9 】

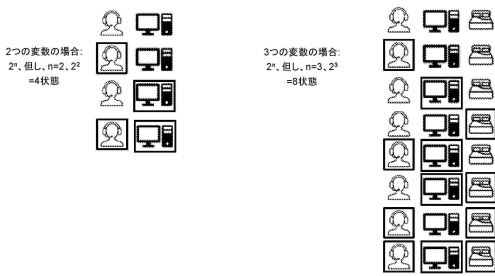
情報 - サチンは夕食を準備した - は、意味を伝達し、
 - サチンは夕食を準備したかもしれない - は意味を伝達し、
 規範情報 - サチンは夕食を準備する
 価値 - サチンは夕食を準備する

【 図 6 0 】

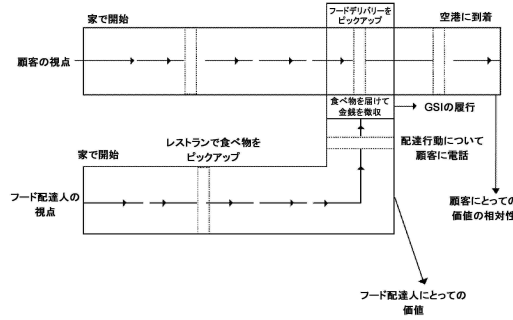


10

【 図 6 1 】

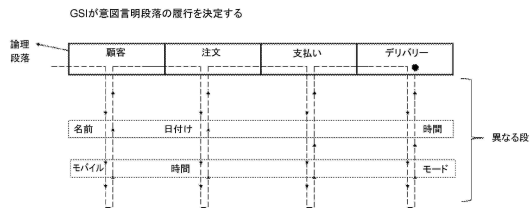


【 図 6 2 】

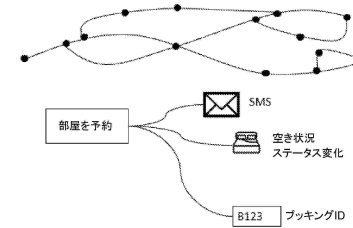


20

【 図 6 3 】



【 図 6 4 】

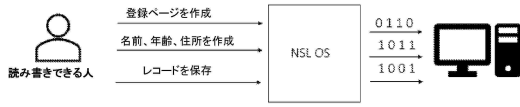


30

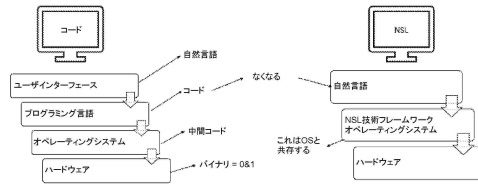
40

50

【 図 6 5 】

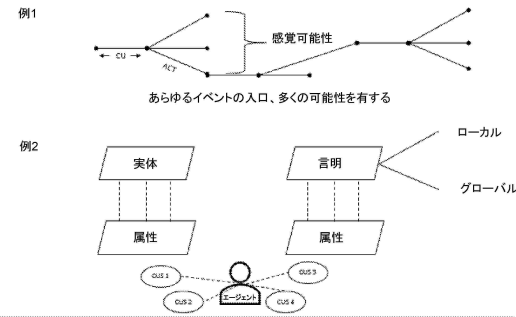


【 図 6 6 】



10

【 図 6 7 】



【 図 6 8 】

ホテルの部屋を予約
 顧客情報を取得、部屋タイププラグジュアリーを選択、部屋タイプデラックスを選択、部屋タイプスーパーデラックスを選択。
 顧客情報を取得、サービスランスポートタイプ1を選択、サービスランスポートタイプ2を選択。
 サービスランスポートタイプ1を選択、サービスランドリータイプ1を選択、サービスランドリータイプ2を選択。
 サービスランドリータイプ1を選択、交通機関情報を取得、ランドリー情報を取得、室料を計算、トランスポート料を計算、ランドリー料金を計算、部屋をキャンセル、ホテルの部屋を予約。

実体	顧客	実体	スーパーデラックス室	実体	客室宿泊情報
1次属性	名前: ラジウ	1次属性	名前: ラジウ 料金の空き状況: 空室	1次属性	日数: 2 稼働 状況: 2
実体	室料	実体	予約確認		
1次属性	日数: "レート" 稼働状況: 10,000	1次属性	ID 番号: ラジウ	料金: 10,000	日数: "レート" 稼働状況: 10,000

ホテルの部屋を予約
 顧客情報を取得、部屋タイププラグジュアリーを選択、部屋タイプデラックスを選択、部屋タイプスーパーデラックスを選択。
 顧客情報を取得、サービスランスポートタイプ1を選択、サービスランスポートタイプ2を選択。
 サービスランスポートタイプ1を選択、サービスランドリータイプ1を選択、サービスランドリータイプ2を選択。
 サービスランドリータイプ1を選択、交通機関情報を取得、ランドリー情報を取得、室料を計算、トランスポート料を計算、ランドリー料金を計算、部屋をキャンセル、ホテルの部屋を予約。

名前	ラジウ	名前	ラジウ	名前	ラジウ
属性	ラジウ	属性	ラジウ	属性	ラジウ

ホテルの部屋を予約
 顧客情報を取得、部屋タイププラグジュアリーを選択、部屋タイプデラックスを選択、部屋タイプスーパーデラックスを選択。
 顧客情報を取得、サービスランスポートタイプ1を選択、サービスランスポートタイプ2を選択。
 サービスランスポートタイプ1を選択、サービスランドリータイプ1を選択、サービスランドリータイプ2を選択。
 サービスランドリータイプ1を選択、交通機関情報を取得、ランドリー情報を取得、室料を計算、トランスポート料を計算、ランドリー料金を計算、部屋をキャンセル、ホテルの部屋を予約。

名前	ラジウ	名前	ラジウ	名前	ラジウ
属性	ラジウ	属性	ラジウ	属性	ラジウ

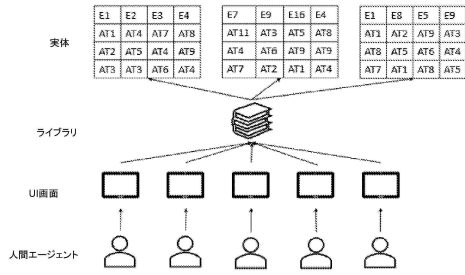
20

30

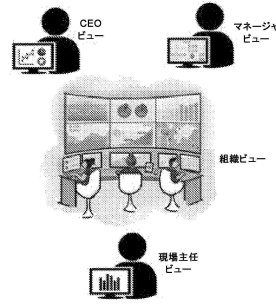
40

50

【図 69】

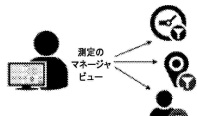


【図 70】



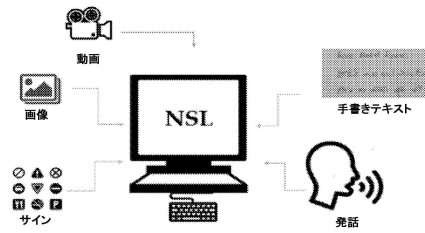
10

【図 71】



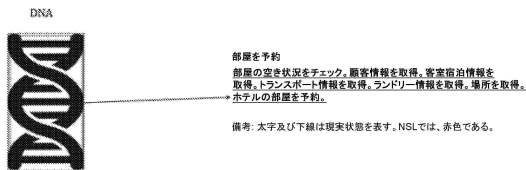
あらゆるステークホルダは、種々のフィルタ(時間フィルタ、空間フィルタ、エージェントフィルタ等)を使用して重要な実体のみを動的に処理することができる

【図 72】



20

【図 73】



【図 74】

ホテルの部屋を予約

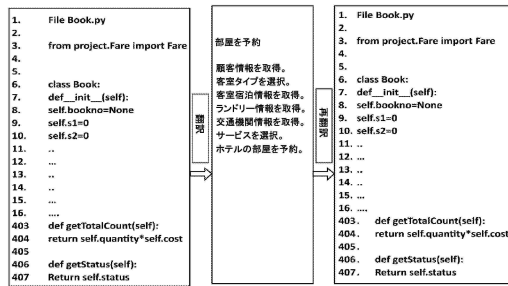
部屋の空き状況をチェック。顧客情報を取得。
客室宿泊情報を取得。トランスポート情報を取得。ランドリー情報を取得。
価格を取得。ホテルの部屋を予約。

30

40

50

【 図 7 5 】



【 図 7 6 】



10

【 図 7 7 】

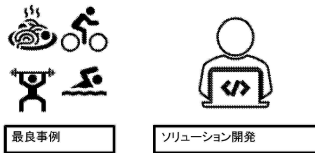
- 自然言語ソリューション: 部屋を予約。
- 自然言語: ホテルの部屋を予約。
- 自然言語ソリューション: 支払いをする。
- 自然言語: 請求書の支払いをする。

【 図 7 8 】

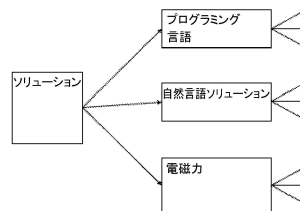
- LSI:
コールセンター主任が顧客に _____
(示唆される動詞: 電話する、郵送する、確認する...)
- LSI:
顧客情報を _____
(示唆される動詞: 取得する、更新する、チェックする、確認する...)

20

【 図 7 9 】



【 図 8 0 】

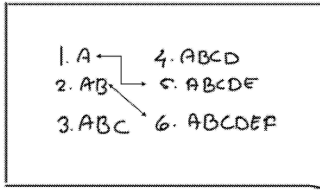


30

40

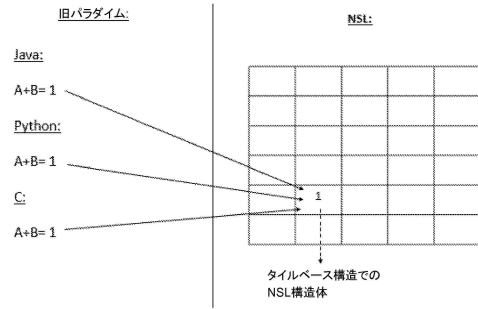
50

【 図 8 1 】



関係距離を測定

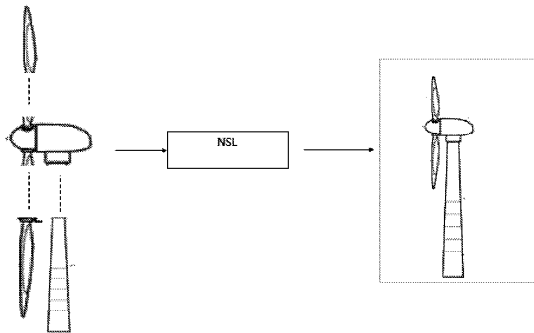
【 図 8 2 】



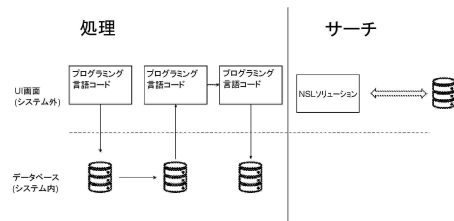
10

【 図 8 3 】

Drawing 83:



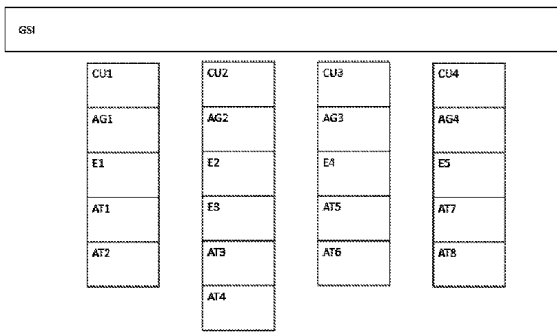
【 図 8 4 】



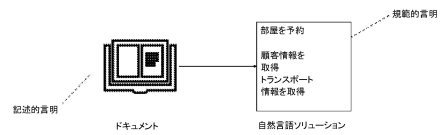
20

【 図 8 5 】

Drawing 85:



【 図 8 6 】

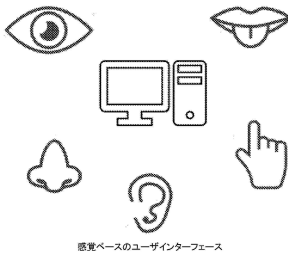


30

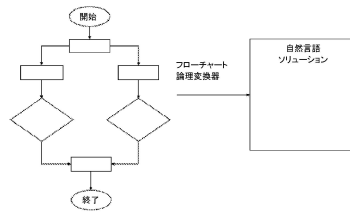
40

50

【図 87】

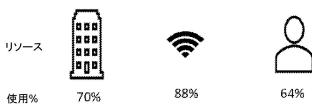


【図 88】



10

【図 89】



【図 90】

ホテルの部屋を予約。
顧客情報を取得、客室タイプを選択、客室宿泊情報を取得、室料を計算、ホテルの部屋を予約。

MSLE

実体	顧客			
1次属性	名前(フリガナ)			

顧客名

ホテルの部屋を予約。
顧客情報を取得、客室タイプスーパーデラックスを選択、客室宿泊情報を取得、室料を計算、ホテルの部屋を予約。

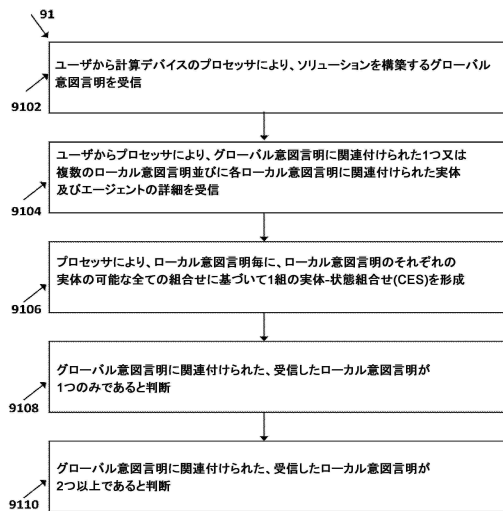
実体	スーパーデラックス室			
1次属性	レート(500)	部屋の空き状況		

客室レートは選択された客室タイプに基づき(定数)である

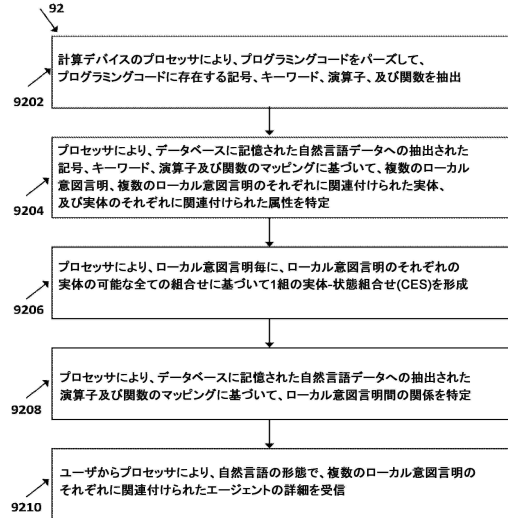
TSLE

20

【図 91】



【図 92】

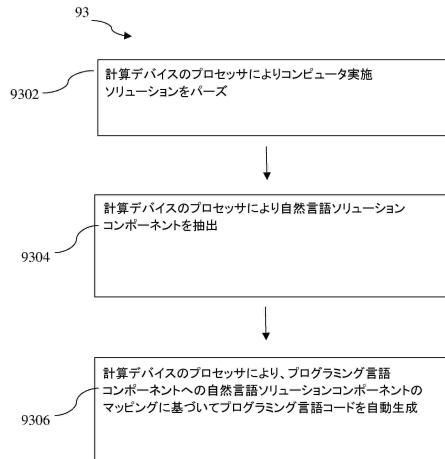


30

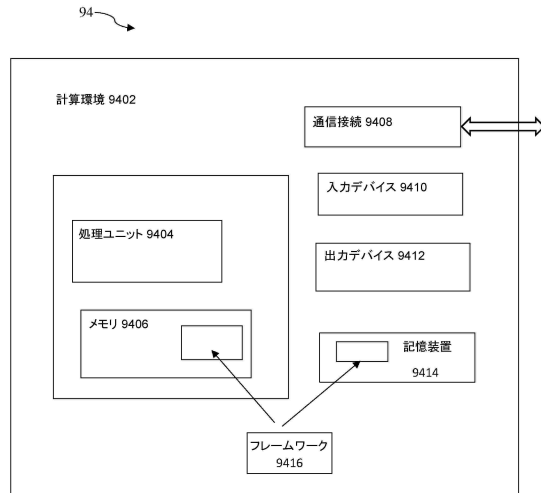
40

50

【図 9 3】



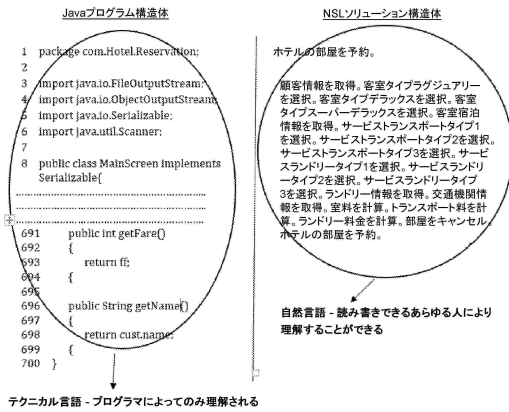
【図 9 4】



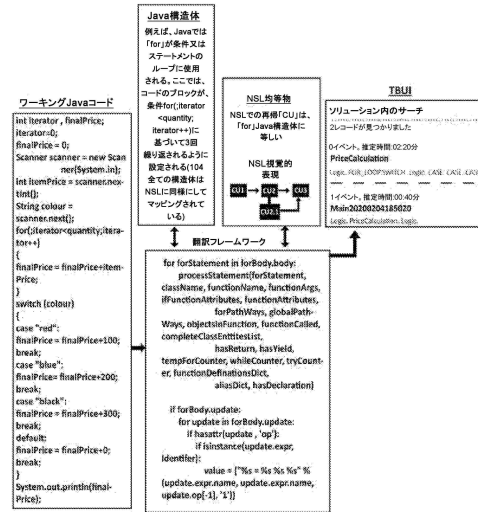
10

20

【図 9 5】



【図 9 6】

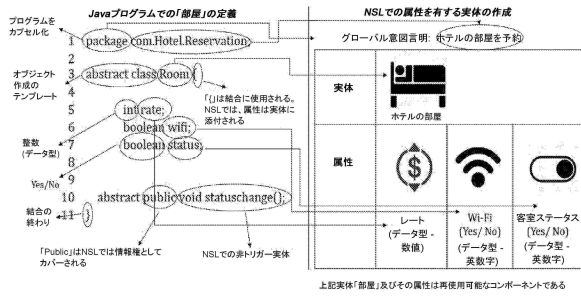


30

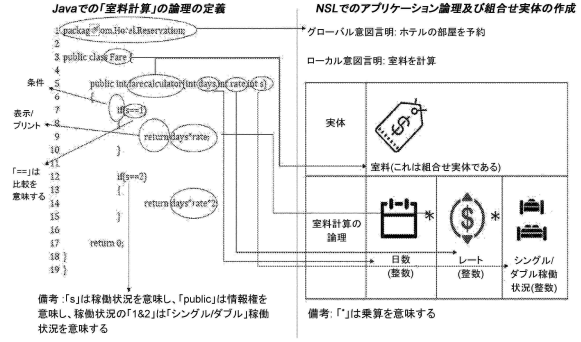
40

50

【図 9 7 A】

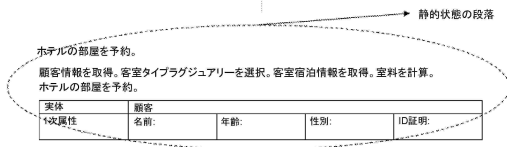


【図 9 7 B】

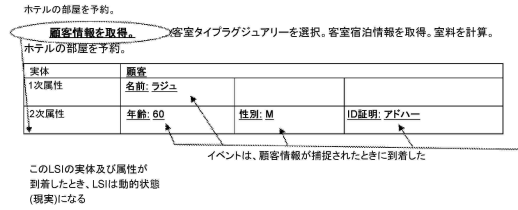


10

【図 9 8 A】



【図 9 8 B】



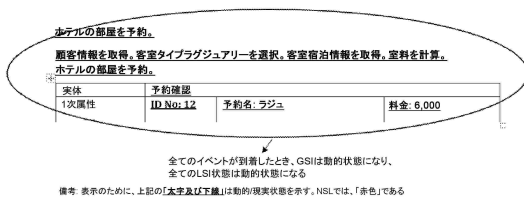
20

30

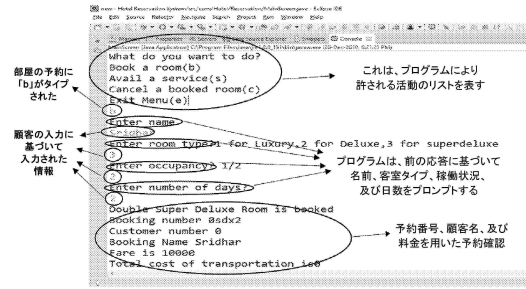
40

50

【 図 9 8 C 】

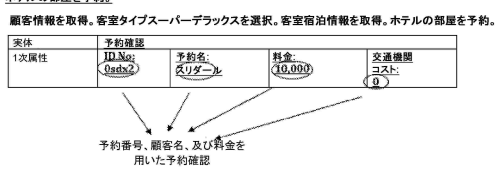
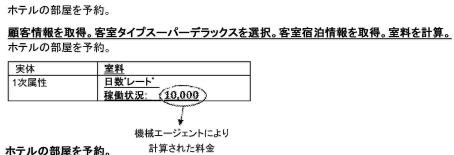
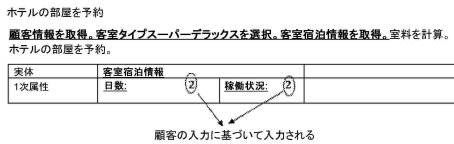
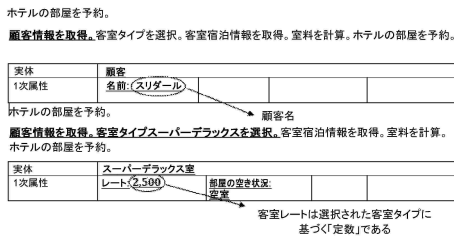


【 図 9 9 A 】



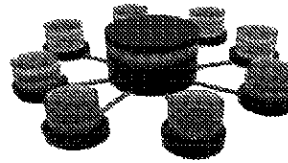
10

【 図 9 9 B 】



【 図 1 0 0 A 】

Drawing 100A



20

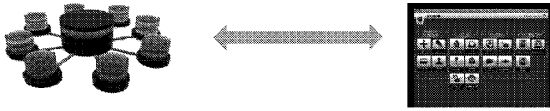
30

40

50

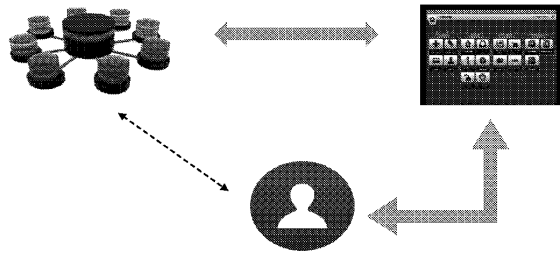
【☒ 100B】

Drawing 100B



【☒ 100C】

Drawing 100C



10

【☒ 101】

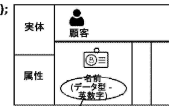
明細	Javaプログラム	NSL
アプリケーションの開発に求められる専門家	Javaプログラマ、UI/UX開発者、データベースプログラマ	読み書きできるあらゆる人
アプリケーション開発時間	約160時間	2時間
ソリューションレーンング時間	約120時間	4時間
アプリケーション構築コスト	約16,000米ドル	約200米ドル
ソリューション論理	システム内	システム外

【☒ 102A】

```

55 System.out.println("What do you want to do?");
56 System.out.println("Book a room(b)");
57 Scanner in = new Scanner(System.in);
58 ch = in.next().charAt(0);

```



Javaでは、これは実体の値を1つのみの英字に制限している

NSLでは、Charはデータ型により制限される。属性として実体に添付される。英字の長さを「1」に制限する場合、それはNSLでのCharに等しい

20

【☒ 102B】

```

5 public class Customer {
6     int no;
7     <String name;
8     String bookingno;
9     int type,id,d;
10    boolean status;

```



Javaでは、文字列は、数字又は英字の何れか一方であることができるデータ型である

NSLでは、文字列はデータ型により表される。データ型「英数字」の添付は「文字列」を表す

【☒ 102C】

```

15 char ch='y';
16 double ff;
17 String name;

```



ダブルは、実体「料金計算機」に小数点以下の数を許容するデータ型として使用される

室料の場合、データ型は属性として添付され、小数点以下の数を許容する数値に設定される

30

40

50

【 102D 】

```

25 for(i=0;3,i++)
26 {
27     ly(i)=new Luxury();
28     ly(i).set(500,false,false);
29 }

```

実体	客室タイプ
1次属性	ラグジュアリー
2次属性	客室数:3

「For」ループは、ここでは、3室のラグジュアリー室がホテルで利用可能であるとデータベースを更新するのに使用されている

「For」ループは、ここでは、3室のラグジュアリー室がホテルで利用可能であるとデータベースを更新するのに使用されている

【 102E 】

```

74 while(true)
75 {
76     int is=0,js=0,ks=0;
77     int id=0,jd=0,kd=0;
78     System.out.println("what do you want to do?");
79     System.out.println("Book a room(b)");
80     System.out.println("Cancel a booked room(c)");
81     System.out.println("Cancel a booked room(c)");
82 }

```

「While」ループは、ここでは、プログラムが提供される利用可能な選択肢のリストを用いてUIを更新するために使用されている。例: 部屋の予約、利用可能なサービス等。活動が完了した後、UIは、「while」ループを使用して無限の時間にわたりコードの同じセクションを使用して再びフラッシュする

ホテルの部屋を予約。
顧客情報を取得。部屋タイプデラックスを選択。
客室宿泊情報を取得。
室料を計算。ホテルの部屋を予約。

「NSL」では、ソリューションの構造体は、あらゆるイベント(各顧客)が段落を有することである

10

【 102F 】

```

95 if(ly[i].getStatus()!=false)
96 {
97     ily-i;
98     flag1=1;
99     break;
100 }
101 else
102     flag1=0;
103 }
104

```

「if」条件は、ここでは、ラグジュアリー室の空き状況のステータスを取得するのに使用されている。ここでは、flag1=1は、部屋が空室であることを示し、flag1=0は部屋が宿泊中であることを示す

NSLでは、部屋が空室である場合、文章は現実になり、段落中の次の文章をトリガーする。「Else」段落は放棄される

ホテルの部屋を予約。
顧客情報を取得。部屋タイプデラックスを選択。
客室宿泊情報を取得。
室料を計算。ホテルの部屋を予約。

【 102G 】

```

95 if(ly[i].getStatus()!=false)
96 {
97     ily-i;
98     flag1=1;
99     break;
100 }
101 else
102     flag1=0;
103 }
104

```

「Break」はここでは、Javaにおいて「if」条件が真である場合、「else」ループを終わらせるために使用されている

ホテルの部屋を予約
顧客情報を取得。部屋タイプスーパーデラックスを選択。
客室宿泊情報を取得。
室料を計算。ホテルの部屋を予約。

ホテルの部屋を予約
顧客情報を取得。部屋タイプラグジュアリーを選択。
客室宿泊情報を取得。
室料を計算。ホテルの部屋を予約。

NSLでは、客室タイプが「スーパーデラックス」として選択される場合、「ラグジュアリー室」の段落はトリガーされない

20

30

40

50

【 図 1 0 3 】

```

Class Book:
  Public void BookLuxury(Customer c, Luxury ly,int type,int ily,int ld,int d)
  { Fare f=new FareC
  If(ld==1)
  System.out.println("Single Luxury Room is booked");
  Ff=f.farecalculator(d,ly.rate,ld);
  Bookno=ily+"lx1";
  BookDisplay(ff,c.name,bookno);
  }
  public void BookDisplay(int ff,String name,String b)
  {
  System.out.println("Booking number"+bookno);
  System.out.println("Customer number"+cust.no);
  System.out.println("Booking Name"+name);
  System.out.println("Fare is"+ff);
  }
}

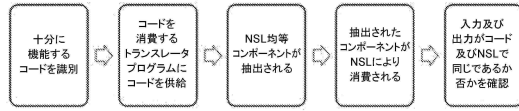
```

NSL表現
 LSI: ラグジュアリー室を予約

実体	客室	客室	顧客	客室	室料
属性	タイプ: レート:	稼働状況	日数:	予約ID:	公式: 日数'稼働状況' レート

プロセスはコードに
 変換される。
 NSLでは、Javaコードの
 各コンポーネントは、
 以下に示されるものと
 同等の情報に
 変換されている

【 図 1 0 4 】



10

【 図 1 0 5 】

```

1. File Book.py
2.
3. from project.Fare import Fare
4. :::
5. ....
6. class Book:
7. def __init__(self):
8. self.bookno=None
9. self.s1=0
10. self.s2=0
11. ...
12. ...
13. ...
14. ...
15. ...
16. ...
403. def getTotalCost(self):
404. return self.quantity*self.cost
405.
406. def getStatus(self):
407. Return self.status

```



ホテルの部屋を予約。
 顧客情報を取得。
 客室タイプを選択。
 客室宿泊情報を取得。
 ランドリー情報を取得。
 交通機関情報を取得。
 サービスを選択。
 ホテルの部屋を予約。

【 図 1 0 6 】

以下の表は、少数のJava構造体及びそれらの対応するNSL均等物を示す

#	JAVA構造体	NSL均等物
1	Javaクラス及びクラス構造体	独立実体
2	クラスメンバ	属性
3	Javaメソッド呼び出し、実行	ドライバ(CD、CES)
4	開発/実行時	潜在/現実
5	Java定数	定数
6	種々の範囲レベルでのJava変数	変数
7	Java数及び数値定数	数
8	Javaキーワード、演算子	単語、演算子
9	単独なJavaプログラミングブロック	基本CU
10	プログラムブロック実行	トランザクションCU
11	ループ構造体	再帰CU
12	条件構造体(if、switch等)	代替CU
13	Java Tokens	ID割り当て
14	JavaフルライフサイクルID	状況的ID
15	フレームワーク、ライブラリ、及びJDK	ブロックボックス

20

30

40

50

【図 107A】

```
public static volatile Scanner scanner;
ArrayList<Teacher> teachers;
ArrayList<Subjects> subjects;
ArrayList<Student> students;
ArrayList<Mark> marks;
```

ここでは、「Static」キーワードが、更なる再使用のために、教師、教科、生徒、成績、等の実体及びその値を凍結するのに使用されている

実体	生徒		
1次属性	年齢	性別	ID NO

NSLでは、「定数」である

【図 107B】

```
switch (option)
case 1:
Subjects subject = new Subjects();
subject.addSubject();
school.subjects.add(subject);
break;
case 2:
Teacher teacher = new Teacher();
teacher.addTeacher();
school.teachers.add(teacher);
break;
case 3:
school.AssignSubjectToTeacher();
break;
```

Case及びSwitchは、値が満たされた場合、あるオブジェクトから別のオブジェクトに移動するためにUIレベルで使用される

教科の詳細を取得。教師情報を取得。教科を教師に割り当てる

NSLでは、2番目の変更単位は、1番目の変更単位が完了した場合のみ、トリガーされる

10

【図 107C】

```
Case8:
school.calculatePercentageOfStudent();
break;
Case 9:
school.junkFunction();
break;
Case 10:
System.out.println("Exiting...\n Have a nice day.");
System.exit(0);
break;
default:
System.out.print("Invalid option!!!\n");
continue;
```

教科の詳細を取得。教師情報を取得。教科を教師に割り当てる。生徒の成績を計算

NSLでは、ソリューションの構造体はあらゆるイベント(各顧客)が役差を有するようものである

「Continue」はここでは、プログラムが提供する利用可能な選択肢のリストを用いてUIを更新するのに使用されている。例、各教科の生徒の成績を入力する。活動が完了した後、UIは、次の生徒の成績を入力するのにコードの同じセクションを使用して再びリフレッシュする

【図 107D】

```
class PersonaiInto
{
String name;
String mobile;
String email;
transien short age;
Address address;
}
```

「Transient」キーワードは、分化に属性が選択的に必要とされる場合、使用される

NSLでは、これは「UI」レベルで使用され、UIは属性として実体添付され、オン/オフ切り替え可能である

20

30

40

50

【 107E 】

```

class Subjects
{
String name;
int pass_mark;
public long max_mark;
int std;
public void addSubject()
{
}
}

```

longキーワードは、数値型として変数を宣言するのに使用される。long値は、-263から263-1までの範囲の64ビット整数を保持することができる

実体	教科		
1次属性	名称	合格成績	最高成績
2次属性		データ型: 整数 サイズ=50	

NSLでは、属性の大きな数値は、実体作成中にサイズを定義することにより捕捉される

【 107F 】

```

class PersonallInfo
{
String name;
String mobile;
String email;
transient short age;
Address address;
}

```

shortキーワードは、数値型として変数を宣言するのに使用される。short値は16ビット整数を保持することができる

実体	生徒		
1次属性	名前	モバイル	年齢
2次属性			データ型: 整数 サイズ=10

NSLでは、属性の大きな数値は、実体作成中にサイズを定義することにより捕捉される

10

【 107G 】

```

public void jakkFunction()
{
byte b;
int j=10;
do
{
j=j-5;
}while(j>=0);
}

```

データ型である。変数毎に1バイトを割り振る

実体	生徒
1次属性	名前
2次属性	データ型: 整数 サイズ=10

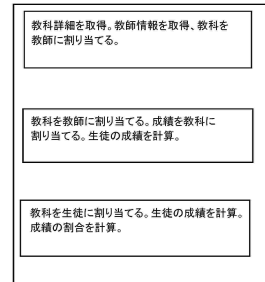
NSLでは、属性の数値は、実体作成中にサイズを定義することにより捕捉される

【 107H 】

```

while(true)
{
school.showOptions();
option = scanner.nextInt();
switch (option)
{
case 2:
Teacher teacher = new Teacher();
teacher.addTeacher();
school.teachers.add(teacher);
break;
default:
System.out.print("Invalid option!!!\n");
continue;
}
}

```



switch caseが定義される場合、使用され、ユーザが、switch内部の何れのcaseにも列記されていない値を入力する場合、プログラムはデフォルトセグメントを実行する。

NSLでは、あらゆるシナリオは別個の段落を有し、高度に分化する。

20

30

40

50

【 107 I 】

```

class Student extends BaseCallInfo
{
    public Student()
    {
        super();
    }
}

```

「super」キーワードは、親実体のプロパティ(生徒観情報)を呼び出すのに使用される

実体	属性
生徒1	学籍番号 名前 モバイル 住所 年齢 電子メール
生徒2	学籍番号 名前 モバイル 住所 年齢 電子メール

値なしで全ての属性をドラッグする。標準属性の容易な複製に使用される

【 107 J 】

```

class Mark
{
    Subjects subject;
    Student student;
    private float mark_scored;

    protected void assignMark(Student student, Subjects subject, float mark_scored)
    {
        this.student = student;
        this.subject = subject;
        this.mark_scored = mark_scored;
    }

    protected float getMarkScored()
    {
        return mark_scored;
    }
}

```

「Protected」キーワードはここでは、決定権を有する教師により教科の生徒に入力された合格成績を凍結するのに使用され、生徒詳細が段落で呼び出される場合は常に、生徒のその教科について、同じ値が使用されることになる

実体	英語教科
属性	合格成績 45

ここでは、合格成績は、成績決定権を通して英語という名前の実体で凍結され、情報として次のレベルに渡される

NSLでは、異なる相座レベルに決定権があり、決定権に基づいて、属性の値はその実体で凍結される

10

【 107 K 】

```

{
    this.student = student;
    this.subject = subject;
    this.mark_scored = mark_scored;
}

```

NSLでは、あらゆる実体は属性と共に作成され、実体が移動/使用される場合は常に、属性も実体と共に移動する

実体	生徒	
属性	名前	スリダール
2次属性	年齢	15
3次属性	モバイル	9849366452

生徒という名前の実体の属性が定義され、この実体で使用される場合は常に、作成中に定義された属性の全てを有することになる

「This」キーワードは、生徒/教科/成績の詳細を生徒/教科/成績の値が定義されたコードの先の部分から呼び出すのに使用される

【 107 L 】

```

// Only one thread can send a message
// @synchronized
synchronized(eeendes)
{
    // synchronizing the send object
    sender.send(msg);
}

```

Synchronizedキーワードは、複数のユーザによる情報の同時使用を凍結するのに使用される。情報は、一度に1人のみユーザにより使用可能である

NSLでは、実体が段落で使用される場合、段落が完了するまで、その実体の詳細は別の段落で使用することができない

20

30

40

50

【 図 1 0 7 M 】

```

)
public strictfp float calculatePercentageOfStudent ()
{
    System.out.println(" Enter role number of the student:");
    int role_number = scanner.nextInt();
    Student student = getStudent(role_number);
    float total = 0;
    float subject_total = 0;
    for (int i=0;i<marks.size();i++)
    {
        if(marks.get(i).student==student)
        {
            total = total + marks.get(i).getMarkScored();
            subject_total = subject_total+
marks.get(i).subject.max_mark;
        }
    }
    float percentage = (total/subject_total) * 100;
    System.out.println("Percentage of
"+student.name+"="+percentage);
    return percentage;
}

```

実体		総合成績割合
属性	名前	スリダール
2次属性	得られた成績	955
3次属性	総合成績	1000
4次属性	成績の%	95.5%

成績の属性%は、成績割合実体では小数点後、1つの値に凍結される

Strick floating point(fp)キーワードは、このプログラムにおいて、小数点後の値の精度を標準化するのに使用されている。例、成績の割合は、小数点後に1つのみの値を有するべきである。例、95.5%

【 図 1 0 7 N 】

```

final int total_marks=100;

System.out.println("Enter role number of the student:");
int role_number=scanner.nextInt();
Student student= getStudent(role_number);

float total= 0;
float subject_total= 0;
for(int i=0;i<marks.size();i++)
{
    if(marks.get(i).student==student)
    {
        total=total+marks.get(i).getMarkScored();
        subject_total=subject_total+marks.get(i).subject.max_mark;
    }
}

```

実体		数学での総合成績割合
属性	名前	スリダール
2次属性	得られた成績	85
3次属性	総合成績	100

Finalキーワードは、このプログラムでは、属性、総合成績の値を固定するのに使用されている

総合成績属性は100に固定され、変更することができない

NSLでは、属性の値は実体において固定され、再使用される場合は常に、変更することができない。最後のレベルの分化を表す

10

20

【 図 1 0 8 A 】

【 図 1 0 8 B 】

ホテルの部屋を予約

顧客情報を取得。客室タイプを選択。客室宿泊情報を取得。室料を計算。ホテルの部屋を予約。

実体	顧客
1次属性	名前: ガイブーク

顧客名

ホテルの部屋を予約

顧客情報を取得。客室タイプスーパーデラックスを選択。客室宿泊情報を取得。室料を計算。ホテルの部屋を予約。

実体	ラグジュアリー室
1次属性	レート: 1,000 客室空き状況: 空き

客室レートは、選択された客室のタイプに基づく「定数」である

顧客情報を取得。客室タイプスーパーデラックスを選択。客室宿泊情報を取得。室料を計算。ホテルの部屋を予約。

実体	客室宿泊情報
1次属性	日数: 3 稼働状況: 2

顧客の入力に基づいて入力される

顧客情報を取得。客室タイプスーパーデラックスを選択。客室宿泊情報を取得。室料を計算。ホテルの部屋を予約。

実体	室料
1次属性	日数'レート': 3,000

料金は機械エージェントにより計算される

ホテルの部屋を予約

顧客情報を取得。客室タイプスーパーデラックスを選択。客室宿泊情報を取得。室料を計算。ホテルの部屋を予約。

実体	予約確認
1次属性	ID No: 012 予約名: ガイブーク 料金: 3,000 交通機関コスト: 0

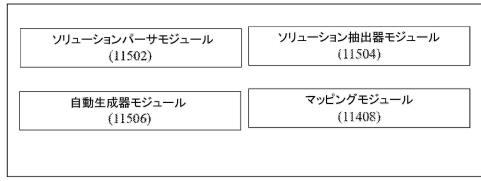
予約番号、顧客名、及び料金を用いた予約確認

30

40

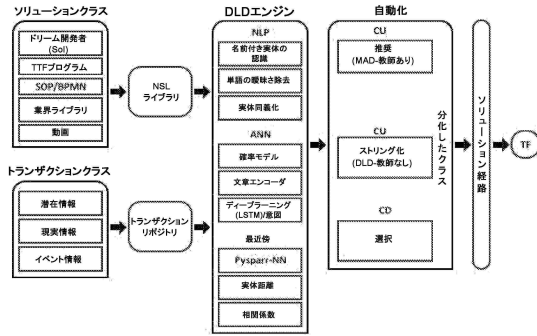
50

【 図 1 1 5 】



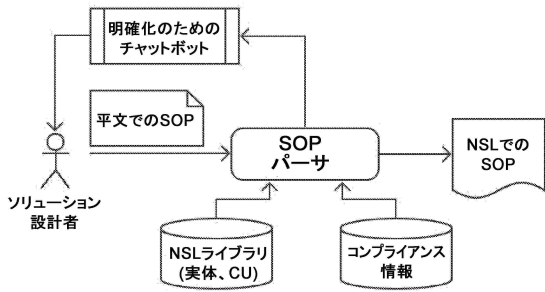
【 図 1 1 6 】

DLD情報フロー

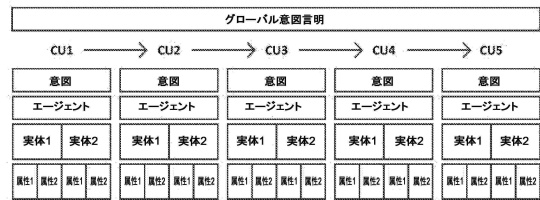


10

【 図 1 1 7 】



【 図 1 1 8 】



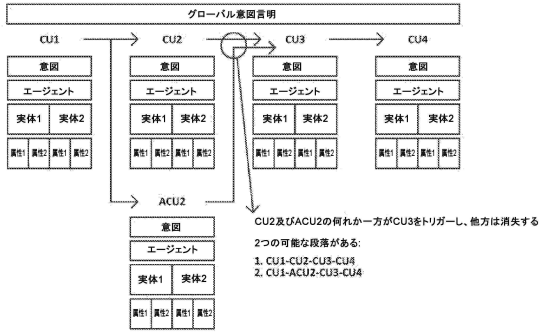
20

30

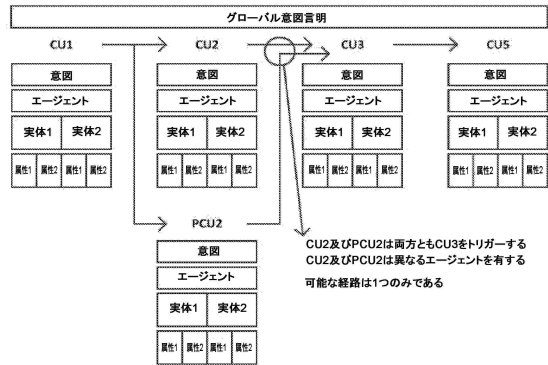
40

50

【 図 1 1 9 】

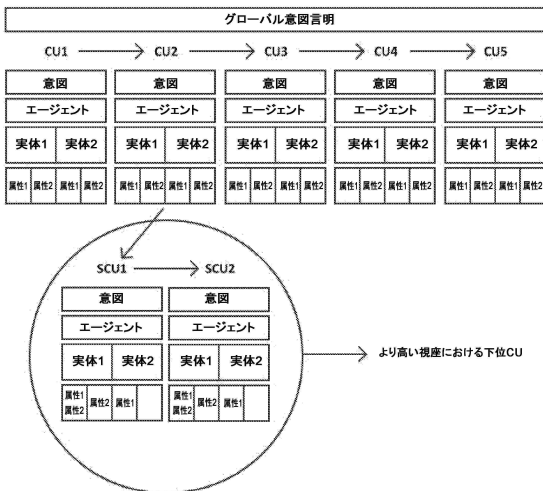


【 図 1 2 0 】

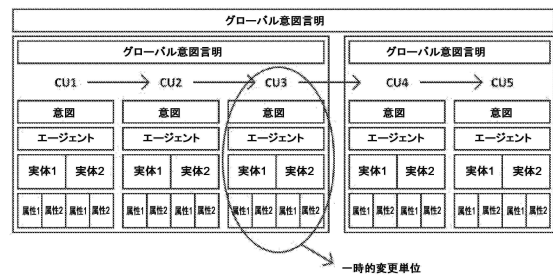


10

【 図 1 2 1 】



【 図 1 2 2 】



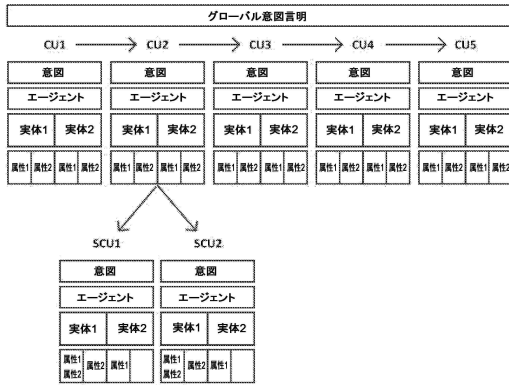
20

30

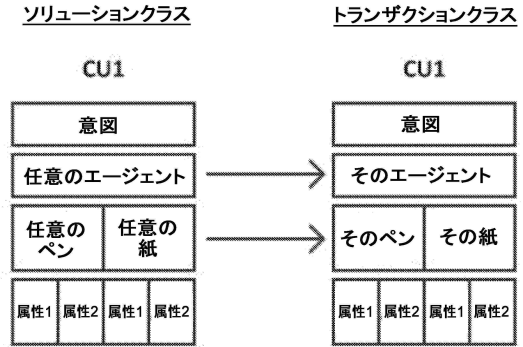
40

50

【 図 1 2 3 】



【 図 1 2 4 】



10

【 図 1 2 5 】



【 図 1 2 6 】



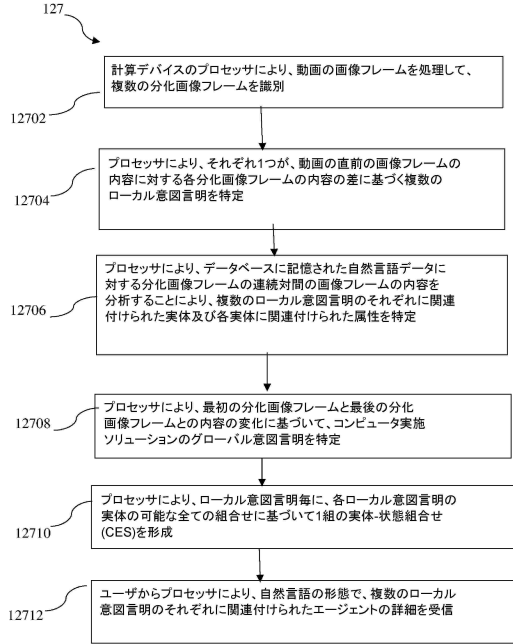
20

30

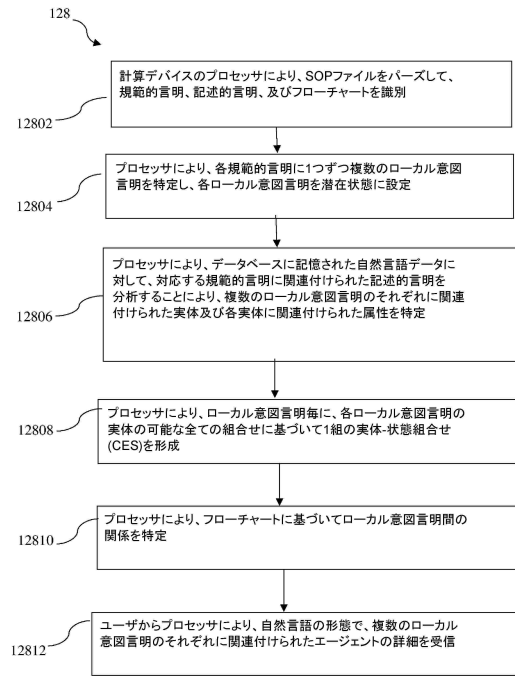
40

50

【図 127】



【図 128】



10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

インド(IN)

早期審査対象出願

(72)発明者 バイラジュ, ラーマリンガ ラジュ

インド国, ハイデラバード 500033, ジュビリー ヒルズ, ロード ナンバー . 62, プロ
ット ナンバー . 1242

審査官 武田 広太郎

(56)参考文献 特開2002-182913(JP, A)

特開2009-181550(JP, A)

欧州特許出願公開第01832975(EP, A1)

(58)調査した分野 (Int.Cl., DB名)

G06F 8/00

G06F 40/56