(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2020年6月25日(25.06.2020)

WIPO | PCT

(10) 国際公開番号
WO 2020/129752 A1

(51) 国際特許分類:
G06F 8/34 (2018.01) *G06F 16/36* (2019.01)

(21) 国際出願番号： PCT/JP2019/048281

(22) 国際出願日： 2019年12月10日(10.12.2019)

(25) 国際出願の言語： 日本語

(26) 国際公開の言語： 日本語

(30) 優先権データ：
特願 2018-236532 2018年12月18日(18.12.2018) JP

(71) 出願人: 日本電信電話株式会社 (**NIPPON TELEGRAPH AND TELEPHONE CORPORATION**) [JP/JP]; 〒1008116 東京都千代田区大手町一丁目５番１号 Tokyo (JP).

(72) 発明者: 孫 晶玉(**Sun, Jingyu**); 〒1808585 東京都武蔵野市緑町３丁目９－１１ ＮＴＴ 知的財産センタ内 Tokyo (JP). 山崎 育生(**YAMASAKI, Ikuo**); 〒1808585 東京都武蔵野市緑町３丁目９－１１ ＮＴＴ 知的財産センタ内 Tokyo (JP). 片山 翔子(**KATAYAMA, Shoko**); 〒1808585 東京都武蔵野市緑町３丁目９－１１ ＮＴＴ 知的財産センタ内 Tokyo (JP). 大森 誓治(**OMORI, Seiji**); 〒1808585 東京都武蔵野市緑町３丁目９－１１ ＮＴＴ 知的財産センタ内 Tokyo (JP).

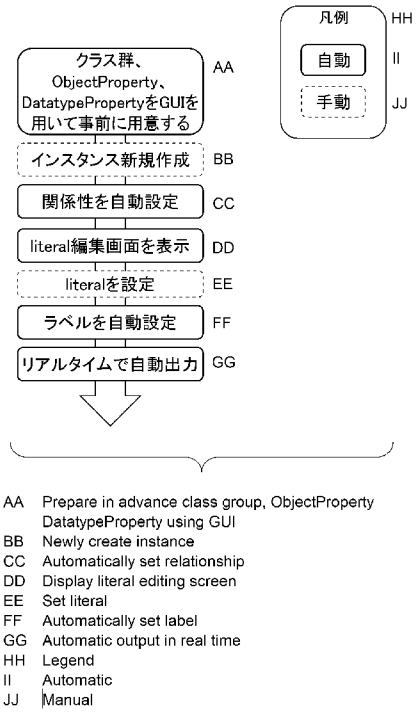
(74) 代理人: 蔵田 昌俊, 外 (**KURATA, Masatoshi** et al.); 〒1050014 東京都港区芝三丁目２３番１号 セレスティン芝三井ビルディング１１階 鈴榮特許綜合事務所内 Tokyo (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,

(54) **Title:** ONTOLOGY CREATION DEVICE, METHOD, AND PROGRAM

(54) 発明の名称： オントロジー作成装置、方法およびプログラム

[図1]



(57) **Abstract:** An ontology creation device according to an embodiment comprises: a first selection unit that accepts an input operation for selecting a class on the basis of information indicating definitions of candidates for classes in an ontology; an acquisition unit that, on the basis of information relating to definitions of properties indicating connection relations between classes as an object and classes as a subject, acquires candidates for a class as the subject when the selected class is used as the object; a second selection unit that accepts an input operation for, on the basis of the candidates, selecting instances belonging to the class as the subject when the selected class is used as the object; a relation setting unit that sets connection relations between instances belonging to the selected class and the selected instances; and an output unit that creates and outputs an ontology indicating the selected class and instances, and the set connection relations.

WO 2020/129752 A1

[続葉有]

SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

一 国際調査報告(条約第21条(3))

(57) 要約:一実施形態に係るオントロジー作成装置は、オントロジーのクラスの候補の定義を表す情報に基づいてクラスを選択するための入力操作を受け付ける第1の選択部と、目的語としてのクラスおよび主語としてのクラスの接続関係を示すプロパティの定義を表す情報に基づいて、選択されたクラスを目的語としたときの主語としてのクラスの候補を取得する取得部と、当該候補に基づいて、選択されたクラスを目的語としたときの主語としてのクラスに属するインスタンスを選択するための入力操作を受け付ける第2の選択部と、選択されたクラスに属するインスタンスと選択されたインスタンスとの間の接続関係を設定する関係設定部と、選択されたクラス、インスタンスおよび設定された接続関係をそれぞれ示すオントロジーを作成して出力する出力部とを有する。

明　細　書

発明の名称 ： オントロジー作成装置、方法およびプログラム

技術分野

[0001] 本発明の実施形態は、オントロジー作成装置、方法およびプログラムに関する。

背景技術

[0002] 近年、オントロジー(Ontology)はWEB分野だけでなく、IoT(Internet of Things)を含む様々な分野で意味および体系を表現可能であるツール（tool）として活躍している。

ところが、オントロジーが記述される言語（RDF(Resource Description Framework)、JSON-LD…）は手動により作成されることが困難な上、この言語が作成される際に、オントロジーが運用されるアプリケーションプログラム（application program）（アプリケーション、又はアプリ）に必要な制約(restriction)が適切に設けられることができない。

[0003] ここで、IoTにおけるオントロジー構築について説明する。図１４は、IoTにおけるオントロジー構築の一例を説明する図である。

IoTデータ（data）の意味および体系を表現する手段として、オントロジーが利用されるIoTプラットフォーム（platform）がある。

[0004] 例えば、データ提供者としての各事業者に共通して設けられてアプリとの間でデータ交換を行なう共通プラットフォーム(PF)と他規格PF/Deviceとの間に変換プロクキ（Proxy）が設けられる。この変換プロクキは、各規格に共通するデータ構造が記述されたオントロジーデータを用いて、共通プラットフォーム(PF)と他規格PF/Deviceとの間でデータ交換を行なう。

[0005] ここでは、IoTデバイス（Device）はサービス（Service）経由でデータの送受信を行なうと仮定する。RESTfulインタフェース（interface）を有する異種規格では、サービスの下にDataPointが定義される。

また、複雑なIoTデバイスでは、「ConsistsOf」フィールド（field）が利

用されて、子デバイスと子サービスとが表現されることによって、階層関係が定義される。

[0006] 例えば、クラス（class）「Device」とクラス「Service」との間はObjectProperty「hasService」で関連付けられ、クラス「Service」とクラス「OutputDataPoint」との間はObjectProperty「hasOutputDataPoint」で関連付けられ、クラス「Service」とクラス「InputDataPoint」との間はObjectProperty「hasInputDataPoint」で関連付けられる。

また、クラス「Service」とリテラル（literal）「API（Application Programming Interface（アプリケーションプログラミングインタフェース））」との間はDatatypeProperty「Protocol:hasAPI」で関連付けられる。ObjectPropertyは、クラス間の関係と、クラスに属するインスタンス（Instance）間の関係とを示す。

[0007] オントロジーを用いて相互連携が行なわれるためには、連携が望まれる他規格PF/Deviceのデータが形式化される。

この形式化が行なわれるには、下記の制約（１）～（３）のように一定的な制約がある。

制約（１）：クラスはThing/Device/Service/DataPointなどに限られる。

制約（２）：クラスの間のObjectPropertyは一定である。

例えば、{Device, hasService, Service}は適切であるが、{Device, hasInputDataPoint, InputDataPoint}は適切でない。

制約（３）：各クラスに付与され得るDatatypePropertyは一定である。

例えば、{Service, hasAPI, API}は適切であるが、{Device, hasAPI, API}は適切でない。

[0008] 従来のオントロジー構築プロセス（process）では、例えば、オントロジー構築対象世界から（１）直感的理解が表出され、この結果をもとに（２）直感的理解が分析され、この分析結果をもとに、（３）オントロジーが形式化される（例えば非特許文献１，２参照）。このようなプロセスは属人性が高いため、形式化されたオントロジーの正確性は保証されにくい。

[0009] ここで、（２）直感的理解の分析と、（３）オントロジーの形式化と、を支援するOntoClean方法論の手順を説明する。

まず、正規化前のオントロジーに対して、まず、それぞれの概念について概念タイプを表すメタプロパティ（meta-property）が判定される。

次に、メタプロパティ間の制約が定義されているので、この制約に従ってDL(Description Logic)推論による一貫性が検証される。

そして、発見された矛盾点が修正され、矛盾が生じなくなるまで上記の判定および検証が繰り返される。

先行技術文献

非特許文献

[0010] 非特許文献1：上田　俊夫（Toshio UEDA）、池田　満（Mitsuru IKEDA）、概念化アスペクト：オントロジー構築の手掛かり（Conceptualizational Aspect: A Lead on Ontology Constructing）、人工知能学会（The Japanese Society for Artificial Intelligence）研究会資料、SIG-SWO-A602-03

非特許文献2：古崎　晃司（Kouji Kozaki）、溝口　理一郎（Riichiro Mizoguchi）、オントロジー構築ツールの現状（A Present State of Ontology Development Tools）

発明の概要

発明が解決しようとする課題

[0011] 従来の構築プロセスの課題について、以下の（１）～（３）にて説明する。

（１）直感的理解の表出で、概念のメタプロパティが判定されるには、客観的な観点から様々な状況を想定して考察する能力が要求される。この能力の習得は容易ではない。

（２）直感的理解が分析されるためには、高度なオントロジーリテラシー（literacy）が要求される。

[0012] （３）オントロジーの形式化では、直感的表現について曖昧な表現、又は

誤っている表現などを修正するには、記述論理などの推論による支援が必要である。（メタプロパティ間の制約ルールを事前定義）

矛盾点、又は誤った記述などは形式化の後に行なわれるため、作成時の防止策はない。

[0013] 次に、従来のオントロジー構築ツールと仕様手順について説明する。

従来のオントロジー構築ツールは、例えばDevice/Serviceなどのクラス、Instance、ObjectProperty、DatatypePropertyを新規作成する。次に、オントロジー構築ツールは、クラスの説明文書であるラベル（label）を追加する。例えばDeviceクラスのラベルは「データを発生し、コマンド（command）を受け取る機器」などに定義され得る。

そして、クラスとクラスの間と、クラスとインスタンスの間との関係性をプロパティを用いて定義し、結果を出力する。このような手順では、上記の課題が存在する。

[0014] この発明は、上記事情に着目してなされたもので、その目的とするところは、データ交換に係るオントロジーの作成を簡素化し、誤作成を防止することができるようにしたオントロジー作成装置、方法およびプログラムを提供することにある。

課題を解決するための手段

[0015] 上記目的を達成するために、この発明の一実施形態に係るオントロジー作成装置の第１の態様は、オントロジーのクラスの候補の定義を表す情報が格納される第１の格納装置と、前記第１の格納装置に格納される定義を表す情報に基づいて行われる、前記オントロジーのクラスの候補からクラスを選択するための入力操作を受け付ける第１の選択部と、目的語としてのクラスおよび主語としてのクラスの接続関係を示すプロパティの定義を表す情報が格納される第２の格納装置と、前記第２の格納装置に格納される定義を表す情報に基づいて、前記第１の選択部により選択されたクラスを目的語としたときの、当該目的語に対する主語としてのクラスの候補を取得する取得部と、前記取得部により取得された主語としてのクラスの候補に基づいて行われる

、前記第１の選択部により選択されたクラスを目的語としたときの、当該目的語に対する主語としてのクラスに属するインスタンスを選択するための入力操作を受け付ける第２の選択部と、前記第１の選択部により選択されたオントロジーのクラスに属するインスタンス、および前記第２の選択部により選択されたインスタンスの間の接続関係を設定する関係設定部と、前記第１の選択部により選択されたオントロジーのクラス、前記第２の選択部により選択されたインスタンス、および前記関係設定部により設定された接続関係をそれぞれ示すオントロジーを作成して出力する出力部と、を備えるようにしたものである。

[0016] この発明のオントロジー作成装置の第２の態様は、第１の態様において、クラスおよびリテラルの関係を示すプロパティの候補の定義を表す情報が格納される第３の格納装置と、前記第３の格納装置に格納されたプロパティの候補の定義を表す情報に基づいて、前記第１の選択部により選択されたクラスに属するインスタンスと前記第２の選択部により選択されたインスタンスとを含むインスタンス群が属するクラスに関係するリテラルを取得する第２の取得部と、前記第２の取得部により取得されたリテラルの編集に係る入力操作を受け付ける編集部と、をさらに備えるようにしたものである。

[0017] この発明のオントロジー作成装置の第３の態様は、第１の態様において、前記出力部は、前記第１の選択部により選択されたオントロジーのクラスと当該クラスに属するインスタンスとの関係を示す情報、前記第１の選択部により選択されたオントロジーのクラスに属するインスタンスと前記第２の選択部により選択されたインスタンスとを含むインスタンス群に属するインスタンス間の関係を示す情報、および前記インスタンス群に属するインスタンスとリテラルとの関係を示す情報の少なくとも１つを生成し、前記生成した情報をオントロジーに変換して出力するようにしたものである。

[0018] この発明のオントロジー作成装置の第４の態様は、第２の態様において、前記第２の取得部により取得されるリテラルは、前記インスタンス群に属するインスタンスのデータタイプであるようにしたものである。

[0019] 本発明の一実施形態に係る、オントロジー作成装置が行うオントロジー作成方法の一つの態様は、第１の格納装置に格納される情報である、オントロジーのクラスの候補の定義を表す情報に基づいて、クラスを入力操作により選択し、第２の格納装置に格納される情報である、目的語としてのクラスおよび主語としてのクラスの接続関係を示すプロパティの定義を表す情報に基づいて、前記選択されたクラスを目的語としたときの、当該目的語に対する主語としてのクラスの候補を取得し、前記取得された主語としてのクラスの候補に基づいて、前記選択されたクラスを目的語としたときの、当該目的語に対する主語としてのクラスに属するインスタンスを入力操作により選択し、前記選択されたオントロジーのクラスに属するインスタンス、および前記選択されたインスタンスの間の接続関係を設定し、前記選択されたオントロジーのクラス、前記選択されたインスタンス、および前記設定された接続関係をそれぞれ示すオントロジーを作成して出力するようにしたものである。

[0020] 本発明の一実施形態に係るオントロジー作成処理プログラムの一つの態様は、第１乃至第４の態様のいずれか１つにおけるオントロジー作成装置の前記各部としてプロセッサを機能させるものである。

発明の効果

[0021] この発明の一実施形態に係るオントロジー作成装置の第１の態様によれば、オントロジーのクラスの候補の定義を表す情報に基づいて、クラスを入力操作により選択し、目的語としてのクラスおよび主語としてのクラスの接続関係を示すプロパティの定義を示す情報に基づいて、選択されたクラスを目的語としたときの、目的語に対する主語としてのクラスの候補を取得し、取得された候補から、選択されたクラスを目的語としたときの主語としてのクラスに属するインスタンスを入力操作により選択し、選択されたクラス、インスタンスの間の接続関係を設定し、選択されたクラス、インスタンスおよび設定された接続関係をそれぞれ示すオントロジーを作成して出力するので、オントロジーの作成を簡素化し、誤作成を防止することができる。

[0022] この発明の一実施形態に係るオントロジー作成装置の第２および第４の態

様によれば、クラスおよびリテラルの関係を示すプロパティの候補の定義を表す情報に基づいて、インスタンス群が属するクラスに関係するリテラルを取得して、このリテラルの編集を行えるので、ユーザは、編集できるリテラルの誤認識を防止することができる。

[0023] この発明の一実施形態に係るオントロジー作成装置の第３の態様によれば、クラスとインスタンスとの関係を示す情報、インスタンス間の関係を示す情報、およびインスタンスとリテラルとの関係を示す情報をオントロジーに変換して出力するので、適切に選択または編集されたクラス、インスタンス、リテラルに基づくオントロジーを出力することができる。

[0024] すなわち、本発明の各態様によれば、データ交換に係るオントロジーの作成を簡素化し、誤作成を防止することが可能になる。

図面の簡単な説明

[0025] [図1]図１は、本発明の一実施形態に係るオントロジー作成装置による手順の一例を示す図である。

[図2]図２は、制約付きオントロジー作成GUIの画面構成の一例を示す図である。

[図3]図３は、本発明の一実施形態に係るオントロジー作成GUIシステム（system）の構成例を示す図である。

[図4]図４は、インスタンスの新規作成に係るワークフロー（workflow）の一例を示す図である。

[図5]図５は、DatatypeProperty設定サブフロー（subflow）の一例を示す図である。

[図6]図６は、owl出力サブフローの一例を示す図である。

[図7]図７は、オントロジーの作成の一例を説明する図である。

[図8]図８は、作成され得るインスタンスに制約が設けられたGUI画面の一例を説明する図である。

[図9]図９は、インスタンス間の関係に制約が設けられたGUI画面の一例を説明する図である。

[図10]図１０は、インスタンスのDatatypePropertyに制約が設けられたGUI画面の一例を説明する図である。

[図11]図１１は、インスタンスのDatatypePropertyに制約が設けされたGUI画面の一例を説明する図である。

[図12]図１２は、オントロジーの出力の一例を説明する図である。

[図13]図１３は、本発明の一実施形態に係るオントロジー作成GUIシステムのハードウエア（hardware）構成の一例を示すブロック図である。

[図14]図１４は、IoTにおけるオントロジー構築の一例を説明する図である。

発明を実施するための形態

[0026] 以下、図面を参照しながら、この発明に係わる一実施形態を説明する。

本発明の一実施形態に係るオントロジー作成装置では、GUI（グラフィカルユーザインタフェース（Graphical User Interface））を用いて、オントロジーを作成する際に、オントロジー運用アプリに必要な制約を設けると共に、オントロジーの作成を簡素化する。

[0027] 第１に、作成され得るクラスがテンプレート（template）として用意されることによって、想定外であるクラスのインスタンスの作成が防がれる。

第２に、目的語インスタンスが選択されてから、設計上許容できる主語インスタンスを本来設計のクラス間の関係（述語）を用いて自動的に選定してアクティブ（active）にし、この選定された結果の中から主語インスタンスをユーザが選択する。これにより、プロパティが誤作成されることが防がれる。

[0028] 第３に、各インスタンスに対して作成され得るデータタイププロパティ（DatatypeProperty）を、システムが、本来の設計である、クラスとプロパティとの関係を自動的に参照して選別し、ユーザにより編集可能な状態にする。これにより、属性が誤作成されることも防がれる。

[0029] 上記により、オントロジーにおける文法的な制約だけでなく、実運用における必要な意味的な制約が設けられるので、オントロジーの作成が簡素化された上、誤作成の防止を可能とする。

また、エンドユーザ（end user）にオントロジー作成を任せる場面、およびオントロジーに関する知識をユーザが十分に有していない場面などに対応でき、オントロジーを利用するアプリの推進が加速され得る。

[0030] 次に、GUIが用いられた、制約付きオントロジーの作成の手順について説明する。図１は、本発明の一実施形態に係るオントロジー作成装置による手順の一例を示す図である。

本発明の一実施形態に係るオントロジー作成装置は、従来のオントロジー構築プロセスとは異なり、オントロジー作成時にGUIを用いて制約を掛け、オントロジーが誤作成されること、およびオントロジーと設計とが矛盾することが有効に防がれる。

[0031] 例えば、従来の手順では、クラス群、インスタンス、ObjectProperty、およびDatatypePropertyを新規作成し、ラベルおよびインスタンスが所属するクラスの関係性と、literal(API)とを設定し、整合性を検証した上で出力する。この手順は、全てユーザによる操作（手動）による手順である。

[0032] これに対し、本発明の一実施形態では、クラス群、ObjectProperty、およびDatatypePropertyがGUIにより事前に用意される。ここでは、設計上の限られたクラスだけ用意される。そして、ユーザ操作によりインスタンスが新規作成される。

そして、オントロジー作成装置は、インスタンスが所属するクラスの関係性を自動で設定し、インスタンスが所属するクラスのDatatypePropertyを継承し、対応のliteral編集画面を自動で表示する。

そして、ユーザ操作によりliteralが設定され、オントロジー作成装置は、ラベルを自動で設定し、このラベルをリアルタイム（real time）で自動的に出力する。

[0033] 次に、制約付きオントロジー作成GUIの概要を説明する。

まず、制約付きオントロジー作成GUIの画面構成について説明する。図２は、制約付きオントロジー作成GUIの画面構成の一例を示す図である。

図２に示された例では、GUIにより事前に用意されたクラス（Device, Serv

ice, InputDataPoint, OutputDataPoint, SimpleDataType）が表示される。

この表示されたクラスが組み合わせられすることでオントロジー図が作成され得る。

この作成されたオントロジー図からOWL(Web Ontology Language)、およびJSONが自動で作成される。

[0034] 次に、制約付きオントロジー作成GUIシステム（オントロジー作成装置）の構成について説明する。

図３は、本発明の一実施形態に係るオントロジー作成GUIシステムの構成例を示す図である。

図３に示されるオントロジー作成GUIシステム１０の機能は、プログラムを実行するＣＰＵ（Central Processing Unit）等のプロセッサ、ＲＡＭ（Random Access Memory）、ＲＯＭ（Read Only Memory）等の記憶媒体、キーボード（keyboard）等の入力装置、およびディスプレイ（display）等の表示装置が用いられることで実現され得る。

[0035] オントロジー作成GUIシステム１０は、クラス選択部１１（A, クラス選択部と表記することがある）、ObjectProperty判定部１２（B, ObjectProperty判定部と表記することがある）、関係性設定部１３（C, 関係性設定部と表記することがある）を有する。

さらに、オントロジー作成GUIシステム１０は、DatatypeProperty判定部１４（D, DatatypeProperty判定部と表記することがある）、literal設定部１５（E, literal設定部と表記することがある）、およびowl出力部１６（F, owl出力部と表記することがある）を有する。

[0036] クラス選択部１１は、クラス定義格納部（格納部（格納装置）Ｘと表記することがある）１１ａを有する。ObjectProperty判定部１２は、ObjectProperty情報格納部（格納部Ｙと表記することがある）１２ａを有する。関係性設定部１３は、関係性情報格納部（格納部Ｚと表記することがある）１３ａを有する。

DatatypeProperty判定部１４は、DatatypeProperty情報格納部（格納部Uと

表記することがある）１４ａを有する。literal設定部１５は、literal情報格納部（格納部Vと表記することがある）１５ａを有する。owl出力部１６は、json/owl情報格納部（格納部Wと表記することがある）１６ａを有する。

[0037] 格納部X, Y, Uに格納される情報は事前に設定される。

格納部Xには、以下の設計クラス群が予め格納される。設計クラス群は、オントロジーのクラスの候補の定義を表す情報である。

設計クラス群{C_1, C_2, …, C_n}：{Device, Service, InputDataPoint, OutputDataPoint, SimpleDataType}

[0038] 格納部Yには、目的語としてのクラスと主語としてのクラスとの間の接続関係を示すプロパティの定義を表す情報である設計ObjectProperty群、例えば下記の（１）～（４）が予め格納される。

設計ObjectProperty群{OP_1, OP_2, …, OP_n}：

（１） OP_1:hasService:(主語)Device→(目的語)Service

（２） OP_2:hasInputDataPoint:(主語)Service→(目的語)InputDataPoint

（３） OP_3:hasOutputDataPoint:(主語)Service→(目的語)OutputDataPoint

（４） OP_3:hasDatatype:(主語)InputDataPoint→(目的語)SimpleDataType|

(主語)InputDataPoint→(目的語)SimpleDataType

[0039] 格納部Uには、プロパティの候補の定義を表す情報である設計DatatypeProperty群、例えば下記の（１）～（５）が予め格納される。このプロパティは、クラスとリテラルとの関係を示す。

設計DatatypeProperty群{DP_1, DP_2, …, DP_n}：

（１） DP_1:rdfs:about:(主語)すべてのクラス→literal

（２） DP_2:hasAPI:主語Service| InputDataPoint|OutputDataPoint→literal

（３） DP_3:storeTimeseries:(主語)Service| OutputDataPoint→literal

（４） DP_4:onDemand:(主語)OutputDataPoint→literal

（５） DP_5:onDemandTS:(主語)OutputDataPoint→literal

[0040] 次に、インスタンスの新規作成に係るワークフローについて説明する。図４は、インスタンスの新規作成に係るワークフローの一例を示す図である。

まず、クラス選択部１１は、新規作成されるインスタンスが所属するクラスC_{select}を、GUI画面（図２に示されたような作成用画面）における右側のエリア（area）に表示される設計クラス群{C_1, C_2, …, C_n}からユーザの操作に従って選択し、この選択した結果を格納部Xに格納する（Ｓ１１）。

クラス選択部１１は、Ｓ１１で選択されたC_{select}のインスタンス$I_{created}$を、ユーザによる入力操作または自動により、GUI画面における左側のエリアに新規作成する（Ｓ１２）。

[0041] ObjectProperty判定部１２は、格納部Yに格納された設計ObjectProperty群{OP_1, OP_2,…, OP_n}から、Ｓ１１で選択されたC_{select}が目的語(Object)であるObjectProperty群OPS_{select}を取得する（Ｓ１３）。

[0042] ObjectProperty判定部１２は、Ｓ１３で取得したOPS_{select}を構成する各ObjectPropertyで定義される、主語（Subject）であるクラス群SS_{select}を取得する（Ｓ１４）。この取得結果は、Ｓ１１で選択されたクラスを目的語としたときの、当該目的語に対する主語として設定可能であるクラスの候補である。

[0043] そして、関係性設定部１３は、Ｓ１３で取得されたクラス群SS_{select}に属するインスタンスがGUI画面における左側のエリアに存在するか否かを判断する（Ｓ１５）。このインスタンスが存在すれば（Ｓ１５のＹｅｓ）、関係性設定部１３は、GUI画面における左側のエリアに表示されるインスタンスであって、Ｓ１３で取得されたクラス群SS_{select}に属するインスタンス{$Parent_1$, $Parent_1$, …, $Parent_n$}を全てハイライト（highlight）にして表示させ、インスタンス間を、ユーザの操作に従って連結可能な状態にする。

[0044] GUI画面で、ユーザの入力操作により所望のインスタンス$Parent_{select}$が選択されると（Ｓ１７）、関係性設定部１３は、この選択されたインスタンス$Parent_{select}$と、Ｓ１２で作成されたインスタンス$I_{created}$とを連結して、この連結を

示す線分をGUI画面に描画する。関係性設定部１３は、連結された各インスタンスに、これらのインスタンス間の関係性を示すプロパティ（ObjectProperty）OP_{select}(OP_{select}∈OPS_{select})を付与し、このプロパティを格納部Zに格納する（Ｓ１８）。

Ｓ１８の後、またはＳ１５でインスタンスが存在しなければ（Ｓ１５のＮｏ）、DatatypeProperty設定サブフロー(Ｓ２０：Ｓ２１～Ｓ２３)、およびowl生成サブフロー(Ｓ３０：Ｓ３１～Ｓ３５)が実行されて処理が終了する。

[0045] 次に、DatatypeProperty設定サブフローについて説明する。図５は、DatatypeProperty設定サブフローの一例を示す図である。

DatatypeProperty設定サブフローが開始されると、DatatypeProperty判定部１４は、Ｓ１２で作成されたインスタンス$I_{editing}$が属するクラスに属するインスタンス、およびＳ１７で選択されたインスタンスに関わるDatatypeProperty群を、格納部Uに格納される設計DatatypeProperty群{DP_1, DP_2, …, DP_n}から取得する（Ｓ２１）。この取得結果は、Ｓ１１で選択されたクラスに属するインスタンスとＳ１７で選択されたインスタンスとにそれぞれ関係するリテラルに対応する。

[0046] DatatypeProperty判定部１４は、Ｓ２１で取得されたDatatypeProperty群で定義されるliteralの編集windowをGUI画面上に表示する（Ｓ２２）。この画面上で、ユーザは入力操作によりliteralを編集できる。

[0047] literal設定部１５は、Ｓ１２で作成された$I_{editing}$と、上記の編集されたliteralと、このliteralに対応するDatatypePropertyである$DP_{editing}$とが関連付けられた情報($I_{editing}$, $DP_{editing}$, literal)を格納部Vに格納する（Ｓ２３）。これによりDatatypeProperty設定サブフローが終了する。

[0048] 次に、owl出力サブフローについて説明する。図６は、owl出力サブフローの一例を示す図である。

owl出力サブフローが開始されると、owl出力部１６は、格納部X, Y, Z, U, Vから、GUI画面における左側のエリアに表示される、各種クラスに属するインスタンス群{I_1, I_2, …, I_n}を取得する（Ｓ３１）。このインスタンス群は

、Ｓ１１で取得されたクラスに属するインスタンスと、Ｓ１７で選択されたインスタンスとを含む。

[0049] owl出力部１６は、Ｓ３１で取得されたインスタンス群{I_1, I_2, …, I_n}を構成する各インスタンスに対して、当該インスタンスと当該インスタンスが属するクラスとの関係性を示す情報(I_i, rafs:subClassof, C_i)を作成する（Ｓ３２）。

[0050] owl出力部１６は、Ｓ３１で取得されたインスタンス群{I_1, I_2, …, I_n}を構成する各インスタンス間の関係性を示す情報(I_i, OP_W, I_j)を作成する（Ｓ３３）。この作成結果は、Ｓ１１で選択されたクラスに属するインスタンスとＳ１７で選択されたインスタンスとを含むインスタンス群に属する、２つのインスタンス間の関係を示す。

[0051] owl出力部１６は、Ｓ３１で取得されたインスタンス群{I_1, I_2, …, I_n}を構成する各インスタンスに対して、当該インスタンスとliteralとの関係性を示す情報(I_i, DP_W, I_j)を作成する（Ｓ３４）。この作成結果は、上記のインスタンス群に属するインスタンスとリテラルとの関係を示す。

owl出力部１６は、Ｓ３２からＳ３４で作成した各情報をowlフォーマット（format）に変換して格納部Wに格納して外部に出力する（Ｓ３５）。これによりowl出力サブフローが終了する。

[0052] 次に、IoT相互連携のためのデータ形式化により作成されるオントロジーの作成について説明する。図７は、オントロジーの作成の一例を説明する図である。

図７に示された例では、インスタンス「getData」および「ManageDev」がクラス「Service」に属し、インスタンス「温度」，「湿度」および「CO2」がクラス「OutputDataPoint」に属し、インスタンス「Interval」および「Switch」がクラス「InputDataPoint」に属する。

[0053] 図７に示された例では、インスタンス「EnvSensor」は、ObjectProperty「hasService」によりインスタンス「getData」および「ManageDev」に関連付けられる。

インスタンス「getData」は、ObjectProperty「hasOutputDataPoint」によりインスタンス「温度」，「湿度」および「CO2」に関連付けられる。

インスタンス「ManageDev」は、ObjectProperty「hasInputDataPoint」によりインスタンス「Interval」および「Switch」に関連付けられる。

[0054] 図７に示された例では、インスタンス「EnvSensor」は、DatatypeProperty「is-a」によりliteral「環境センサー（sensor）」と関連付けられる。

インスタンス「ManageDev」は、DatatypeProperty「hasAPI」によりliteral「Sensor/Service1」と関連付けられる。

インスタンス「getData」は、DatatypeProperty「hasAPI」によりliteral「Sensor/Service2」と関連付けられる。

[0055] 図７に示された例では、インスタンス「温度」は、DatatypeProperty「hasAPI」によりliteral「DATA:arg(argument)1」と関連付けられる。

インスタンス「湿度」は、DatatypeProperty「hasAPI」によりliteral「DATA:arg2」と関連付けられる。

インスタンス「CO2」は、DatatypeProperty「hasAPI」によりliteral「DATA:arg3」と関連付けられる。

インスタンス「Interval」は、DatatypeProperty「hasAPI」によりliteral「DATA:arg4」と関連付けられる。

インスタンス「Switch」は、DatatypeProperty「hasAPI」によりliteral「DATA:arg5」と関連付けられる。

インスタンス「温度」，「湿度」，「CO2」，「Interval」および「Switch」は、DatatypeProperty「hasDatatype」によりliteral「float」と関連付けられる。

[0056] 次に、GUI画面における、作成され得きるインスタンスに制約が設けられた例を説明する。図８は、作成され得るインスタンスに制約が設けられたGUI画面の一例を説明する図である。

図８に示された例では、ユーザは、GUI画面における右側のエリアに用意された候補である、クラスの候補「Device」，「Service」，「InputDataPoint

」，OutputDataPoint」および「SimpleDataType」を利用して、GUI画面における左側のエリアにインスタンスを作成（描画）することができる。図８に示された例では、下記の（１）～（７）が作成される。

[0057] （１） クラス「Device」に属するインスタンス「EnvSensor」

（２） インスタンス「EnvSensor」の下層のインスタンスであって、クラス「Service」に属するインスタンス「getData」および「ManageDev」

（３） インスタンス「getData」の下層のインスタンスであって、クラス「OutputDataPoint」に属するインスタンス「温度」，「湿度」および「CO2」

（４） インスタンス「温度」および「湿度」の下層のインスタンスであって、クラス「SimpleDataType」に属するインスタンス「float」

（５） インスタンス「CO2」の下層のインスタンスであって、クラス「SimpleDataType」に属するインスタンス「integer」

（６） インスタンス「ManageDev」の下層のインスタンスであって、クラス「InputDataPoint」に属するインスタンス「Interval」および「Switch」

（７） インスタンス「Interval」および「Switch」の下層のインスタンスであって、クラス「SimpleDataType」に属するインスタンス「Binary」

[0058] 次に、GUI画面における、インスタンス間の関係に制約が設けられた例を説明する。図９は、インスタンス間の関係に制約が設けられたGUI画面の一例を説明する図である。

インスタンスの間の関係性を示す線分は、設計ObjectProperty群に従った関係性を示す線分としてのみ、GUI画面上で描画され得る。一方で、設計ObjectProperty群に従わない関係性を示す線分はGUI画面上で描画されることはできず、関係性設定部１３により該当の関係性が設定されることもできない。

[0059] 図９に示された例では、GUI画面上におけるクラス「Device」に属するインスタンスと、クラス「OutputDataPoint」に属するインスタンスとの間に、線分が描画されることはできない。

[0060] また、図９に示された例では、GUI画面上におけるクラス「Service」に属

するインスタンスと、クラス「InputDataPoint」に属するインスタンスとの間には、関係性を示す１本の線分が描画され得るが、同じインスタンス間に２本以上の線分は描画され得ない。

[0061] 次に、インスタンスのDatatypePropertyに制約が設けられた例を説明する。図１０および図１１は、インスタンスのDatatypePropertyに制約が設けられたGUI画面の一例を説明する図である。

図１０に示された例では、クラス「Device」および「SimpleDataType」に係るノードでは、インスタンスで設定可能であるDatatypePropertyは、「NAME:名前変更」および「DELETE:削除」に制限される。

[0062] また、図１１に示された例では、クラス「Service」，「InputDataPoint」および「OutputDataPoint」に係るノード（node）では、インスタンスで設定可能であるDatatypePropertyは、「NAME:名前変更」，「DELETE:削除」，「hasAPI」および「保管状態（storeTimeseries（時系列データの保管）、onDemand（アプリからの要求によりデータを取得）およびonDemandTS（アプリからの要求により時系列データを取得））」に制限される。

[0063] 次に、オントロジーの出力の例について説明する。図１２は、オントロジーの出力の一例を説明する図である。

上記図７に示された、EnvSensorのセンサーにかかるクラス、インスタンスが、図８に示されるようにGUI画面上に描画されたときは、図１２に示されるように、CO2のセンサーと、温度のセンサーと、ON/OFFスイッチ（switch）を有するLightデバイスのオントロジーであるOWLファイル（file）が、owl出力部１６により作成される。

[0064] 図１３は、本発明の一実施形態に係るオントロジー作成GUIシステムのハードウエア構成の一例を示すブロック図である。

図１３に示された例では、上記の実施形態に係るオントロジー作成GUIシステム１０は、例えばサーバコンピュータ（server computer）またはパーソナルコンピュータ（personal computer）により構成され、ＣＰＵ等のハードウエアプロセッサ（hardware processor）１１１Ａを有する。そして、このハ

ードウエアプロセッサ１１１Ａに対し、プログラムメモリ（program memory）１１１Ｂ、データメモリ（data memory）１１２、入出力インタフェース１１３及び通信インタフェース１１４が、バス（bus）１２０を介して接続される。

[0065] 通信インタフェース１１４は、例えば１つ以上の無線の通信インタフェースユニットを含んでおり、通信ネットワークＮＷとの間で情報の送受信を可能にする。無線インタフェースとしては、例えば無線ＬＡＮ（Local Area Network）などの小電力無線データ通信規格が採用されたインタフェースが使用される。

[0066] 入出力インタフェース１１３には、オントロジー作成GUIシステム１０に付設される、オペレータ（operator）用の入力デバイス３０および出力デバイス４０が接続される。

入出力インタフェース１１３は、キーボード、タッチパネル（touch panel）、タッチパッド（touchpad）、マウス（mouse）等の入力デバイス３０を通じてオペレータにより入力された操作データを取り込むとともに、出力データを液晶または有機ＥＬ（Electro Luminescence）等が用いられた表示デバイスを含む出力デバイス４０へ出力して表示させる処理を行なう。なお、入力デバイス３０および出力デバイス４０には、オントロジー作成GUIシステム１０に内蔵されたデバイスが使用されてもよく、また、ネットワーク（network）ＮＷを介してオントロジー作成GUIシステム１０と通信可能である他の情報端末の入力デバイスおよび出力デバイスが使用されてもよい。

[0067] プログラムメモリ１１１Ｂは、非一時的な有形の記憶媒体として、例えば、ＨＤＤ（Hard Disk Drive）またはＳＳＤ（Solid State Drive）等の随時書込みおよび読出しが可能な不揮発性メモリ（non-volatile memory）と、ＲＯＭ（Read Only Memory）等の不揮発性メモリとが組み合わせて使用されたもので、一実施形態に係る各種制御処理を実行する為に必要なプログラムが格納されている。

[0068] データメモリ１１２は、有形の記憶媒体として、例えば、上記の不揮発性

メモリと、ＲＡＭ（Random Access Memory）等の揮発性メモリ（volatile memory）とが組み合わせて使用されたもので、各種処理が行なわれる過程で取得および作成された各種データが記憶される為に用いられる。

[0069] 本発明の一実施形態に係るオントロジー作成GUIシステム１０は、ソフトウエア（software）による処理機能部として、図３などに示されるクラス選択部１１、ObjectProperty判定部１２、関係性設定部１３、DatatypeProperty判定部１４、literal設定部１５およびowl出力部１６を有するデータ処理装置として構成され得る。

[0070] クラス定義格納部１１ａ、ObjectProperty情報格納部１２ａ、関係性情報格納部１３ａ、DatatypeProperty情報格納部１４ａ、literal情報格納部１５ａおよびjson/owl情報格納部１６ａは、図１３に示されたデータメモリ１１２が用いられることで構成され得る。ただし、これらの格納部はオントロジー作成GUIシステム１０内に必須の構成ではなく、例えば、ＵＳＢ（Universal Serial Bus）メモリなどの外付け記憶媒体、又はクラウド（cloud）に配置されたデータベースサーバ（database server）等の記憶装置に設けられたものであってもよい。

[0071] 上記のクラス選択部１１、ObjectProperty判定部１２、関係性設定部１３、DatatypeProperty判定部１４、literal設定部１５およびowl出力部１６の各部における処理機能部は、いずれも、プログラムメモリ１１１Ｂに格納されたプログラムを上記ハードウエアプロセッサ１１１Ａにより読み出させて実行させることにより実現され得る。なお、これらの処理機能部の一部または全部は、特定用途向け集積回路（ＡＳＩＣ（Application Specific Integrated Circuit））またはＦＰＧＡ（Field-Programmable Gate Array）などの集積回路を含む、他の多様な形式によって実現されてもよい。

[0072] 以上のように、本発明の一実施形態に係るオントロジー作成装置は、オントロジーのクラスの候補の定義を表す情報に基づいてクラスを入力操作により選択し、目的語としてのクラスおよび主語としてのクラスの接続関係を示すプロパティの定義を表す情報に基づいて、選択されたクラスを目的語とし

たときの、当該目的語に対する主語としてのクラスの候補を取得し、取得された候補に基づいて、選択されたクラスを目的語としたときの主語としてのクラスに属するインスタンスを入力操作により選択し、選択されたクラスに属するインスタンス、選択されたインスタンスの間の接続関係を設定し、選択されたクラス、選択されたインスタンス、および設定された接続関係をそれぞれ示すオントロジーを作成して出力するので、オントロジーの作成を簡素化し、誤作成を防止することができる。

[0073] つまり、IoTシステムにおいてデータ提供者の追加、又はセンサデバイスの変更が生じ、オントロジーの更新が頻繁に行なわれる際に、オントロジーに精通したユーザでなくても、IoTシステムの管理者が、オントロジーの文法上の制約、又は実運用において必要な制約が予め設定されているGUIを用いることにより、比較的短時間で正しく適切なオントロジーの作成が可能になる。

[0074] また、各実施形態に記載した手法は、計算機（コンピュータ）に実行させることができるプログラム（ソフトウエア手段）として、例えば磁気ディスク（フロッピー（登録商標）ディスク（Floppy disk）、ハードディスク等）、光ディスク（ＣＤ－ＲＯＭ、ＤＶＤ、ＭＯ等）、半導体メモリ（ＲＯＭ、ＲＡＭ、フラッシュメモリ（Flash memory）等）等の記録媒体に格納し、また通信媒体により伝送して頒布することもできる。なお、媒体側に格納されるプログラムには、計算機に実行させるソフトウエア手段（実行プログラムのみならずテーブル、データ構造も含む）を計算機内に構成させる設定プログラムをも含む。本装置を実現する計算機は、記録媒体に記録されたプログラムを読み込み、また場合により設定プログラムによりソフトウエア手段を構築し、このソフトウエア手段によって動作が制御されることにより上述した処理を実行する。なお、本明細書でいう記録媒体は、頒布用に限らず、計算機内部あるいはネットワークを介して接続される機器に設けられた磁気ディスク、半導体メモリ等の記憶媒体を含むものである。

[0075] なお、本発明は、上記実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。また、各実

施形態は適宜組み合わせて実施してもよく、その場合組み合わせた効果が得られる。更に、上記実施形態には種々の発明が含まれており、開示される複数の構成要件から選択された組み合わせにより種々の発明が抽出され得る。例えば、実施形態に示される全構成要件からいくつかの構成要件が削除されても、課題が解決でき、効果が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

符号の説明

[0076] １０…オントロジー作成GUIシステム

１１…クラス選択部

１１ａ…クラス定義格納部

１２…ObjectProperty判定部

１２ａ…ObjectProperty情報格納部

１３…関係性設定部

１３ａ…関係性情報格納部

１４…DatatypeProperty判定部

１４ａ…DatatypeProperty情報格納部

１５…literal設定部

１５ａ…literal情報格納部

１６…owl出力部

１６ａ…json/owl情報格納部

請求の範囲

[請求項1] オントロジーのクラスの候補の定義を表す情報が格納される第１の格納装置と、

前記第１の格納装置に格納される定義を表す情報に基づいて行われる、前記オントロジーのクラスの候補からクラスを選択するための入力操作を受け付ける第１の選択部と、

目的語としてのクラスおよび主語としてのクラスの接続関係を示すプロパティの定義を表す情報が格納される第２の格納装置と、

前記第２の格納装置に格納される定義を表す情報に基づいて、前記第１の選択部により選択されたクラスを目的語としたときの、当該目的語に対する主語としてのクラスの候補を取得する取得部と、

前記取得部により取得された主語としてのクラスの候補に基づいて行われる、前記第１の選択部により選択されたクラスを目的語としたときの、当該目的語に対する主語としてのクラスに属するインスタンスを選択するための入力操作を受け付ける第２の選択部と、

前記第１の選択部により選択されたオントロジーのクラスに属するインスタンス、および前記第２の選択部により選択されたインスタンスの間の接続関係を設定する関係設定部と、

前記第１の選択部により選択されたオントロジーのクラス、前記第２の選択部により選択されたインスタンス、および前記関係設定部により設定された接続関係をそれぞれ示すオントロジーを作成して出力する出力部と、

を備えたオントロジー作成装置。

[請求項2] クラスおよびリテラルの関係を示すプロパティの候補の定義を表す情報が格納される第３の格納装置と、

前記第３の格納装置に格納されたプロパティの候補の定義を表す情報に基づいて、前記第１の選択部により選択されたクラスに属するインスタンスと前記第２の選択部により選択されたインスタンスとを含

むインスタンス群が属するクラスに関係するリテラルを取得する第２の取得部と、

前記第２の取得部により取得されたリテラルの編集に係る入力操作を受け付ける編集部と、

をさらに備える、

請求項１に記載のオントロジー作成装置。

[請求項3] 前記出力部は、

前記第１の選択部により選択されたオントロジーのクラスと当該クラスに属するインスタンスとの関係を示す情報、前記第１の選択部により選択されたオントロジーのクラスに属するインスタンスと前記第２の選択部により選択されたインスタンスとを含むインスタンス群に属するインスタンス間の関係を示す情報、および前記インスタンス群に属するインスタンスとリテラルとの関係を示す情報の少なくとも１つを生成し、前記生成した情報をオントロジーに変換して出力する、

請求項１に記載のオントロジー作成装置。

[請求項4] 前記第２の取得部により取得されるリテラルは、前記インスタンス群に属するインスタンスのデータタイプである、

請求項２に記載のオントロジー作成装置。

[請求項5] オントロジー作成装置が行なうオントロジー作成方法であって、

第１の格納装置に格納される情報である、オントロジーのクラスの候補の定義を表す情報に基づいて、クラスを入力操作により選択し、

第２の格納装置に格納される情報である、目的語としてのクラスおよび主語としてのクラスの接続関係を示すプロパティの定義を表す情報に基づいて、前記選択されたクラスを目的語としたときの、当該目的語に対する主語としてのクラスの候補を取得し、

前記取得された主語としてのクラスの候補に基づいて、前記選択されたクラスを目的語としたときの、当該目的語に対する主語としての

クラスに属するインスタンスを入力操作により選択し、

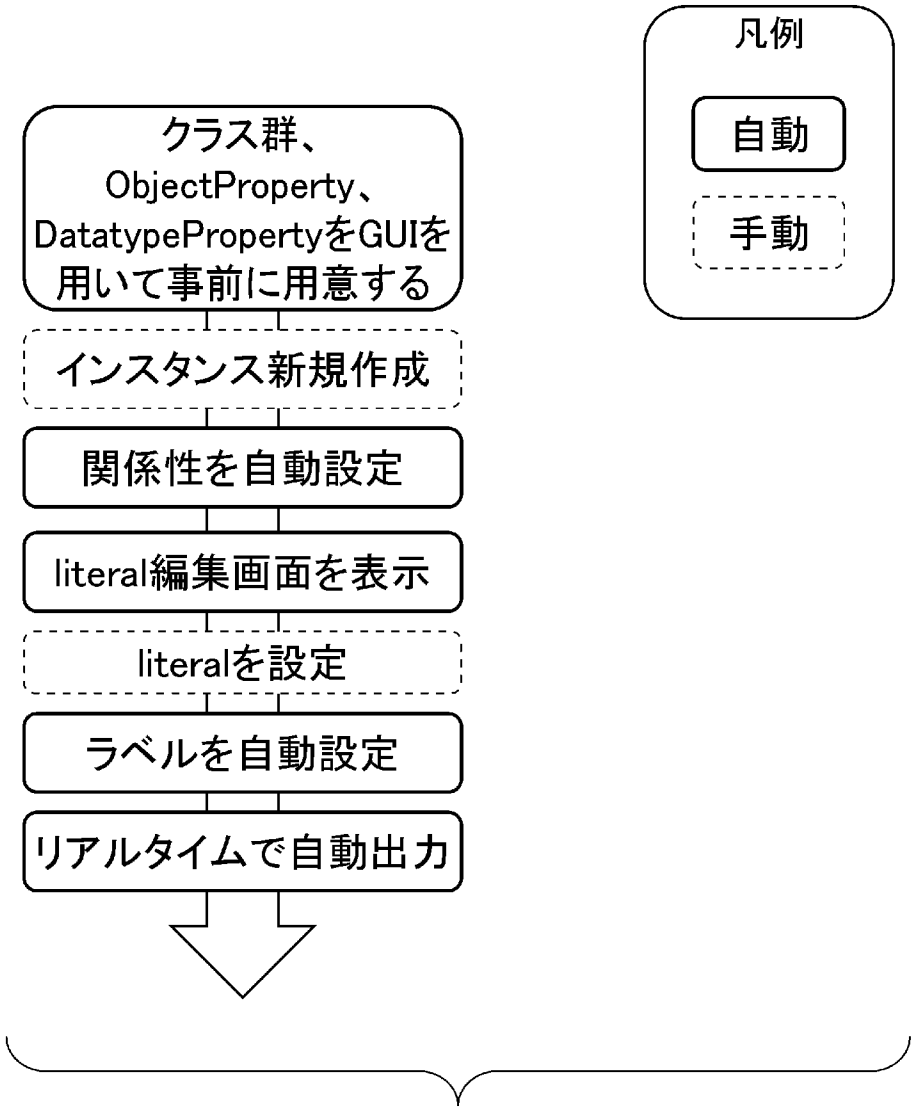
前記選択されたオントロジーのクラスに属するインスタンス、および前記選択されたインスタンスの間の接続関係を設定し、

前記選択されたオントロジーのクラス、前記選択されたインスタンス、および前記設定された接続関係をそれぞれ示すオントロジーを作成して出力する、

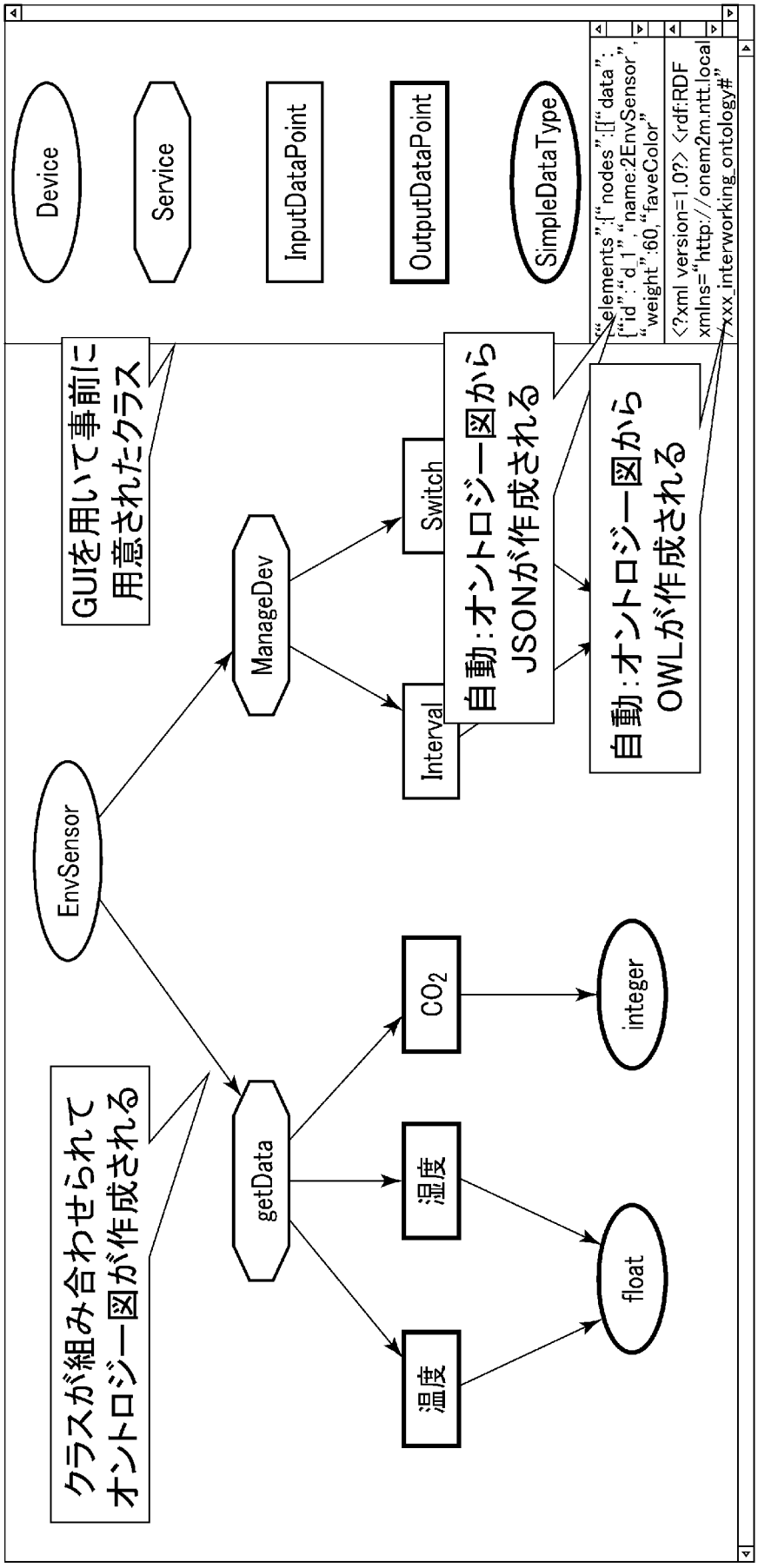
オントロジー作成方法。

[請求項6] 請求項１乃至４のいずれか１項に記載のオントロジー作成装置の前記各部としてプロセッサを機能させるオントロジー作成処理プログラム。

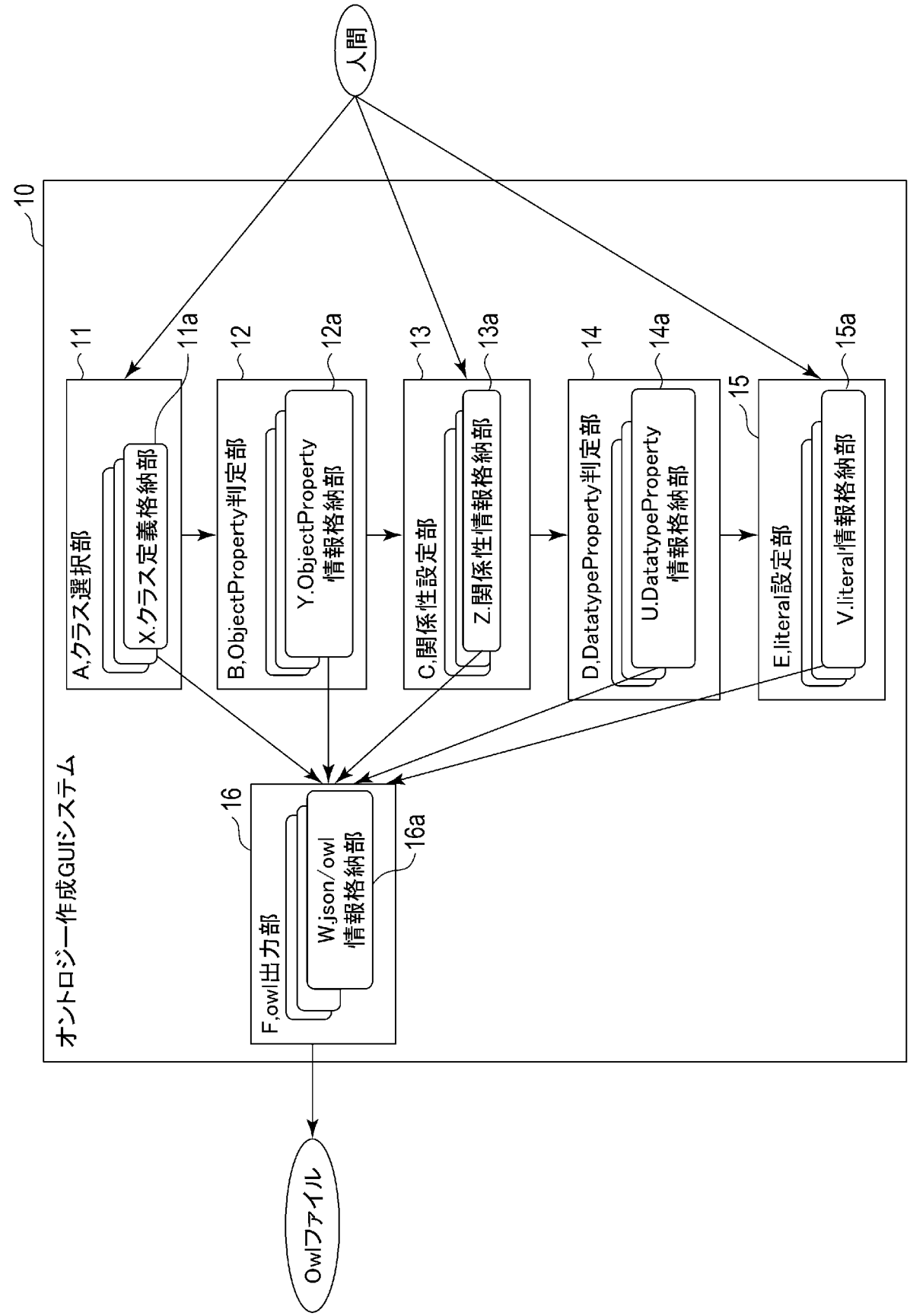
[図1]


凡例
自動
手動
クラス群、
ObjectProperty、
DatatypePropertyをGUIを
用いて事前に用意する
インスタンス新規作成
関係性を自動設定
literal編集画面を表示
literalを設定
ラベルを自動設定
リアルタイムで自動出力

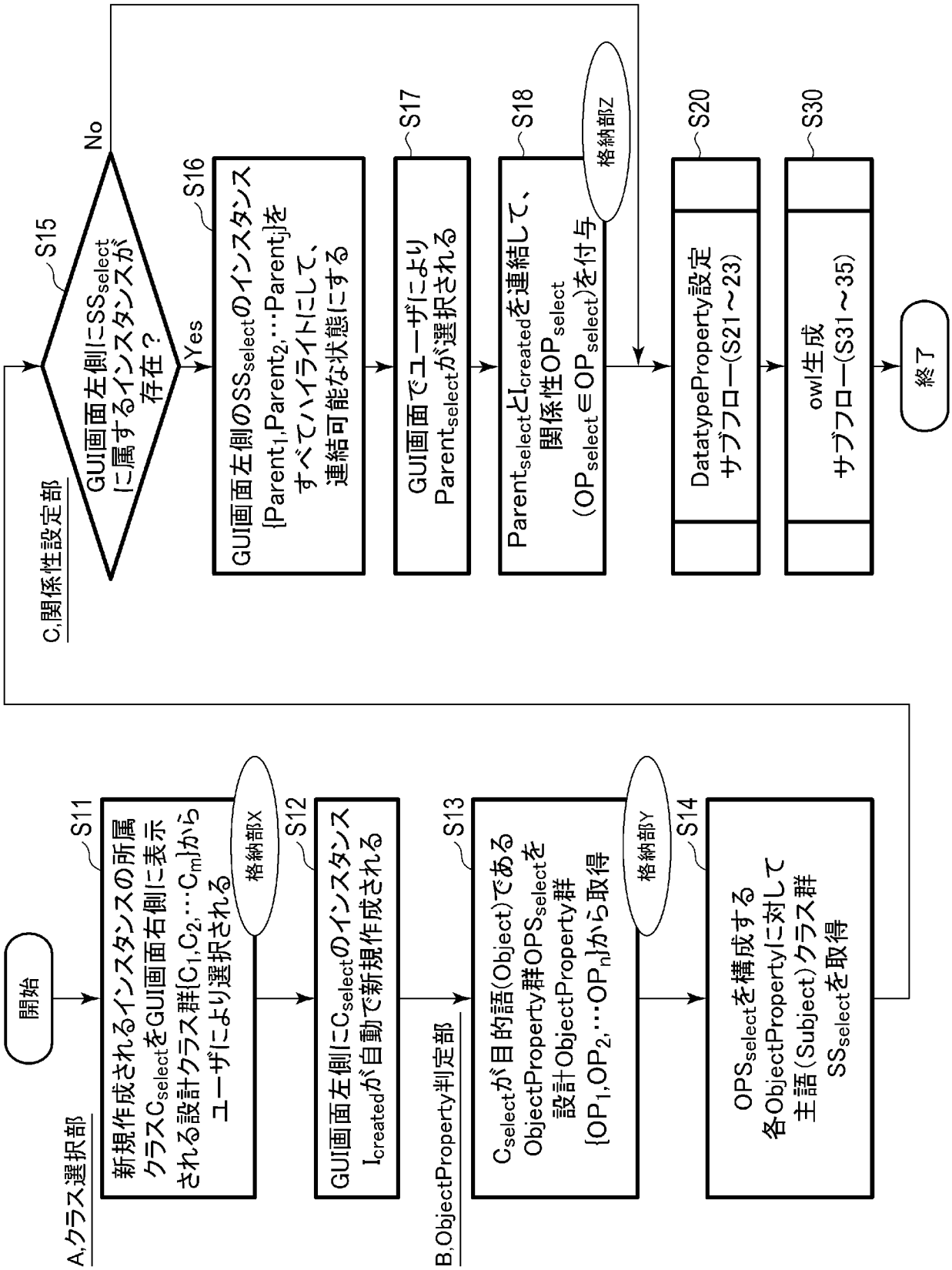
[図2]


Device
Service
InputDataPoint
OutputDataPoint
SimpleDataType
GUIを用いて事前に
用意されたクラス
ManageDev
Switch
Interval
自動:オントロジー図から
JSONが作成される
自動:オントロジー図から
OWLが作成される
EnvSensor
クラスが組み合わせられて
オントロジー図が作成される
getData
CO2
湿度
温度
integer
float
{"elements":[{"nodes":[{"data":
{"id":"d_1","name:2EnvSensor",
"weight":60,"faveColor"
<?xml version=1.0?> <rdf:RDF
xmlns="http://onem2m.ntt.local
/xxx_interworking_ontology#"

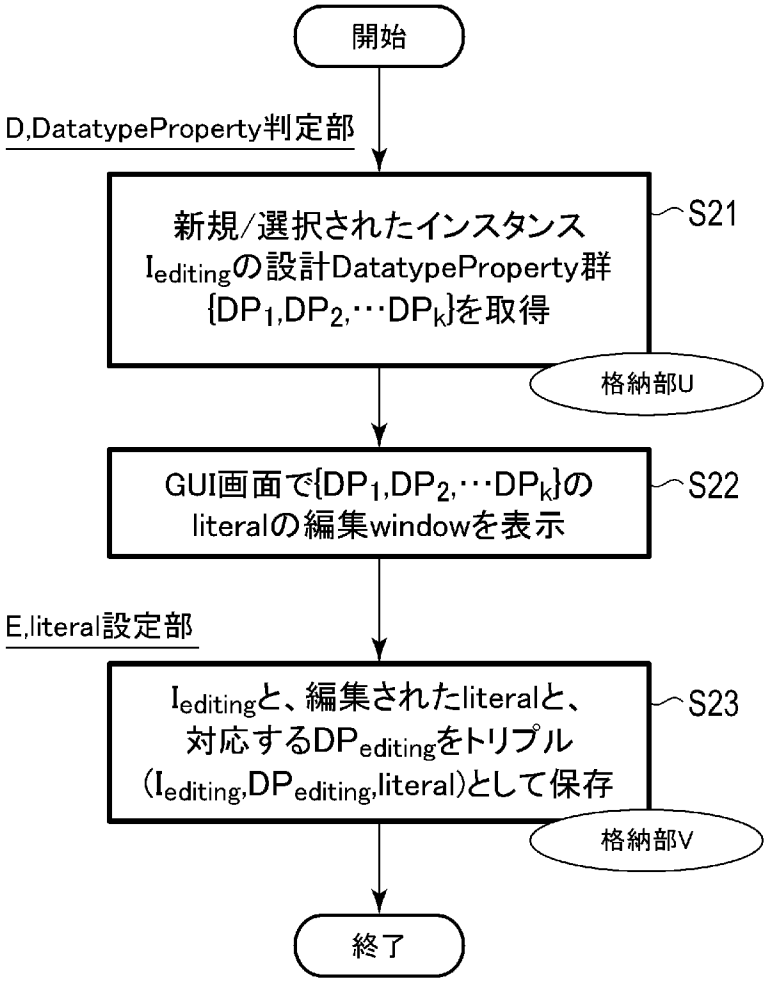
[図3]


人間
10
オントロジー作成GUIシステム
11
A,クラス選択部
X.クラス定義格納部
11a
12
B,ObjectProperty判定部
Y.ObjectProperty
情報格納部
12a
13
C,関係性設定部
Z.関係性情報格納部
13a
14
D,DatatypeProperty判定部
U.DatatypeProperty
情報格納部
14a
15
E,literal設定部
V.literal情報格納部
15a
16
F,owl出力部
W.json/owl
情報格納部
16a
Owlファイル

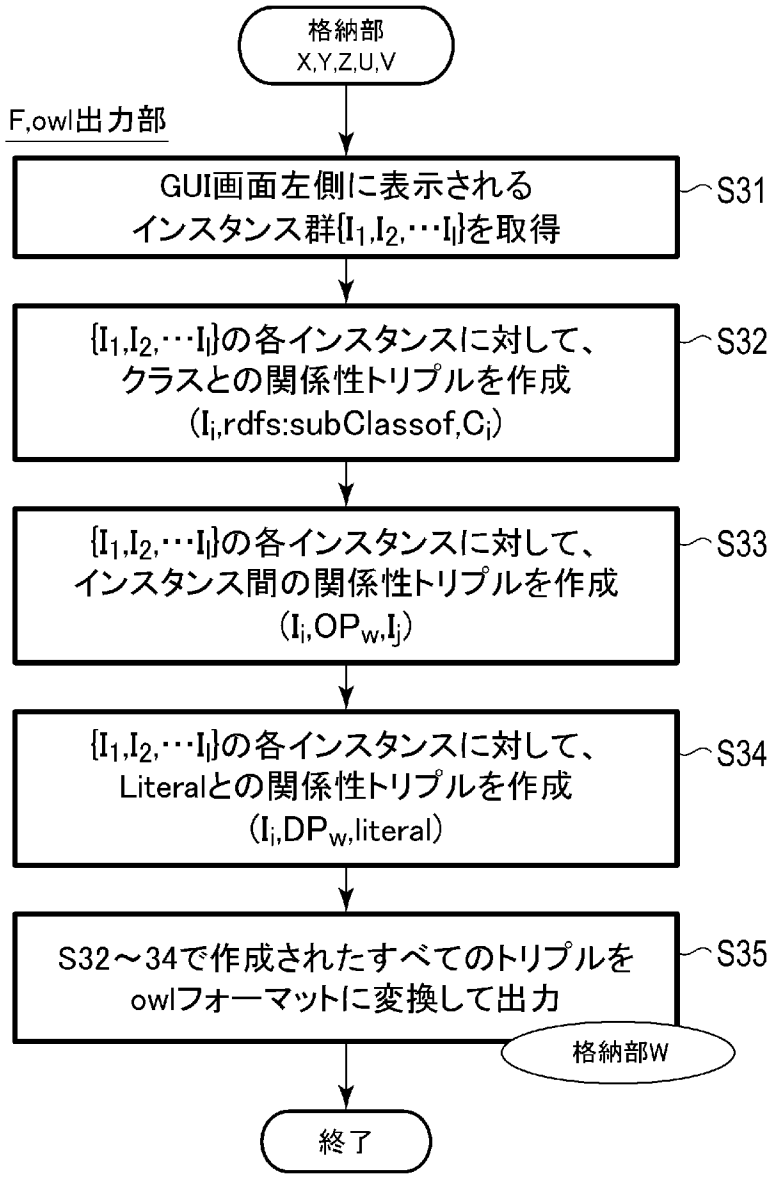
[図4]


開始
A,クラス選択部
S11
新規作成されるインスタンスの所属クラスCselectをGUI画面右側に表示される設計クラス群{C1,C2,…Cm}からユーザにより選択される
格納部X
S12
GUI画面左側にCselectのインスタンスIcreatedが自動で新規作成される
B,ObjectProperty判定部
S13
Cselectが目的語(Object)であるObjectProperty群OPSselectを設計ObjectProperty群{OP1,OP2,…OPn}から取得
格納部Y
S14
OPSselectを構成する各ObjectPropertyに対して主語(Subject)クラス群SSselectを取得
C,関係性設定部
S15
GUI画面左側にSSselectに属するインスタンスが存在？
No
Yes
S16
GUI画面左側のSSselectのインスタンス{Parent1,Parent2,…Parentj}をすべてハイライトにして、連結可能な状態にする
S17
GUI画面でユーザによりParentselectが選択される
S18
Parentselectと Icreatedを連結して、関係性OPselect(OPselect∈OPselect)を付与
格納部Z
S20
DatatypeProperty設定サブフロー(S21～23)
S30
owl生成サブフロー(S31～35)
終了

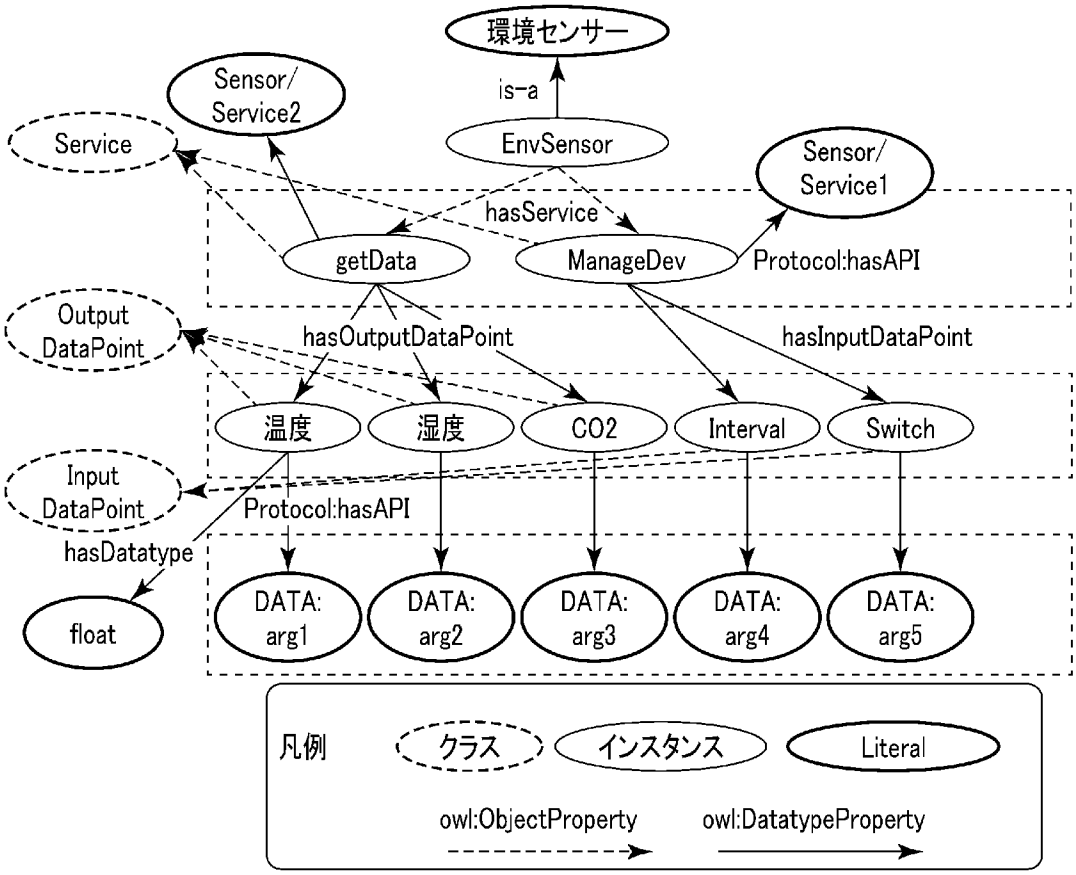
[図5]


開始
D,DatatypeProperty判定部
新規/選択されたインスタンス
$I_{editing}$の設計DatatypeProperty群
{DP_1,DP_2,…DP_k}を取得
S21
格納部U
GUI画面で{DP_1,DP_2,…DP_k}の
literalの編集windowを表示
S22
E,literal設定部
$I_{editing}$と、編集されたliteralと、
対応する$DP_{editing}$をトリプル
($I_{editing}$,$DP_{editing}$,literal)として保存
S23
格納部V
終了

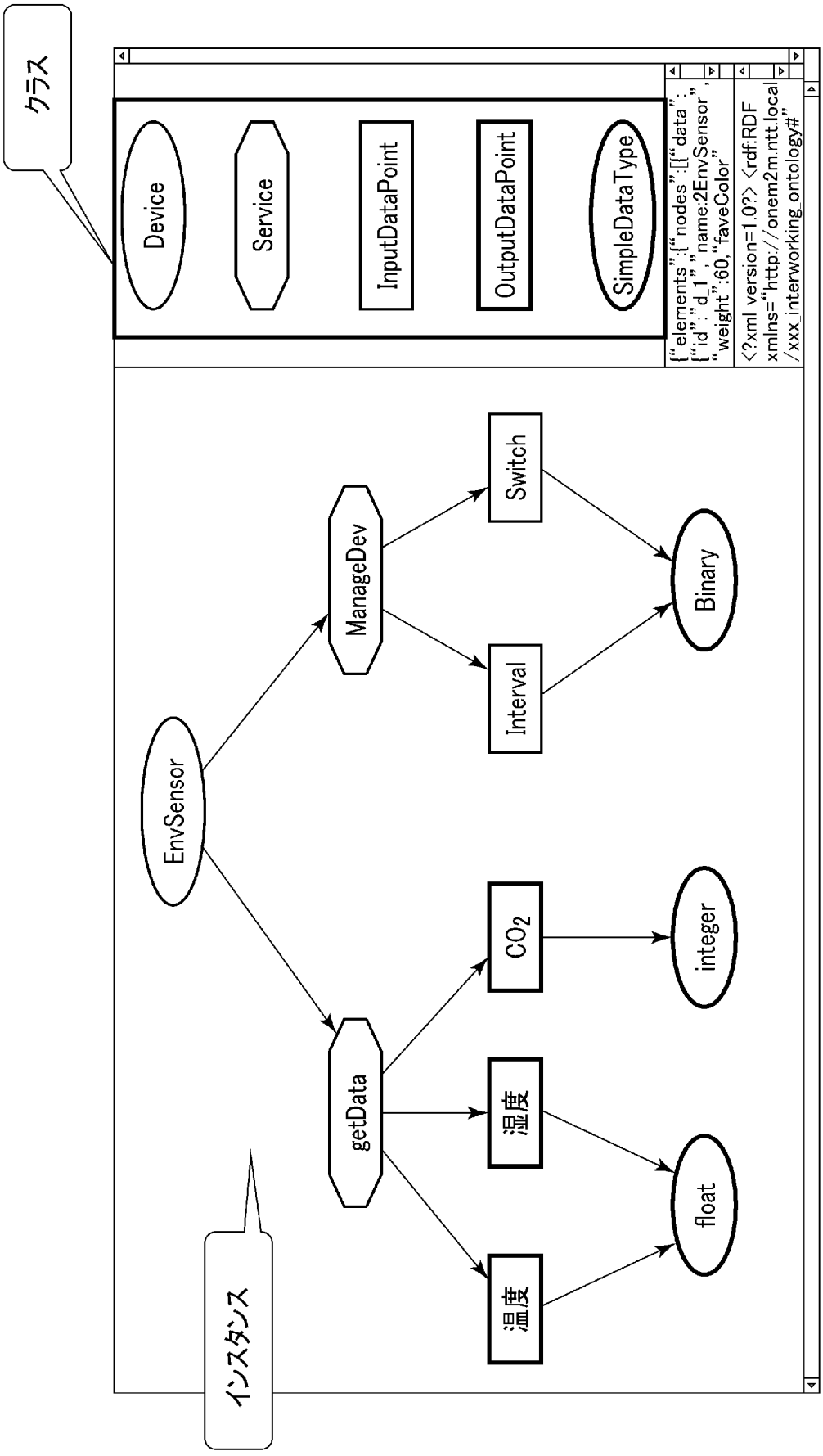
[図6]


格納部
X,Y,Z,U,V
F,owl出力部
GUI画面左側に表示される
インスタンス群{I1,I2,…Il}を取得
S31
{I1,I2,…Il}の各インスタンスに対して、
クラスとの関係性トリプルを作成
(Ii,rdfs:subClassof,Ci)
S32
{I1,I2,…Il}の各インスタンスに対して、
インスタンス間の関係性トリプルを作成
(Ii,OPw,Ij)
S33
{I1,I2,…Il}の各インスタンスに対して、
Literalとの関係性トリプルを作成
(Ii,DPw,literal)
S34
S32～34で作成されたすべてのトリプルを
owlフォーマットに変換して出力
S35
格納部W
終了

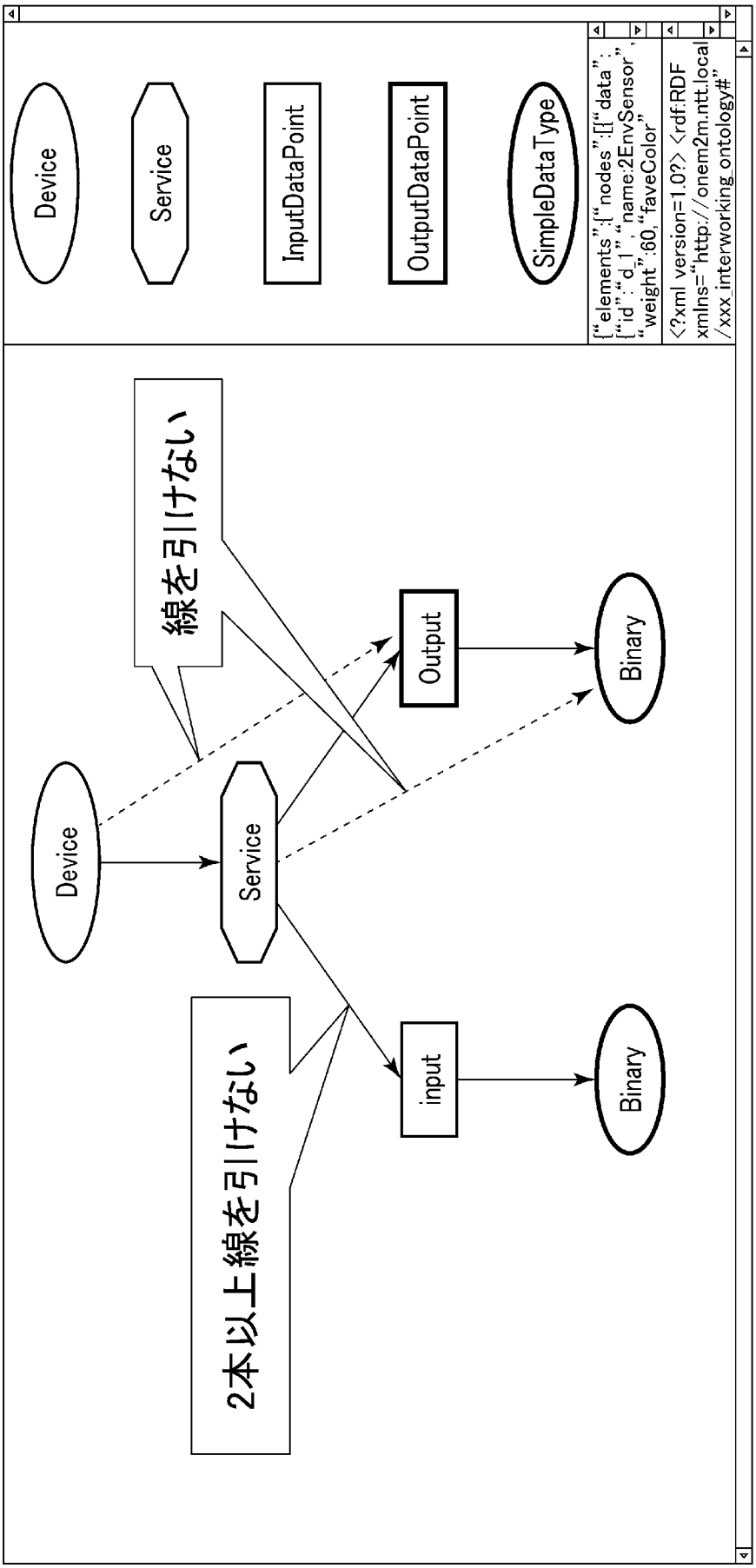
[図7]


環境センサー
is-a
EnvSensor
Sensor/
Service2
Service
Sensor/
Service1
hasService
getData
ManageDev
Protocol:hasAPI
Output
DataPoint
hasOutputDataPoint
hasInputDataPoint
温度
湿度
CO2
Interval
Switch
Input
DataPoint
Protocol:hasAPI
hasDatatype
float
DATA:
arg1
DATA:
arg2
DATA:
arg3
DATA:
arg4
DATA:
arg5
凡例
クラス
インスタンス
Literal
owl:ObjectProperty
owl:DatatypeProperty

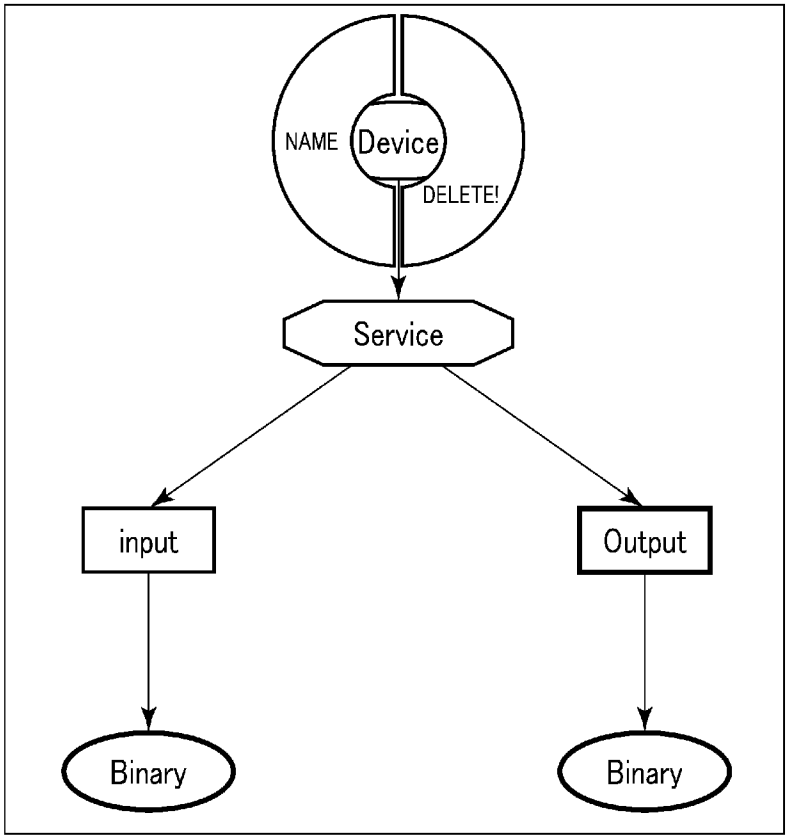
[図8]


クラス
Device
Service
InputDataPoint
OutputDataPoint
SimpleDataType
{"elements":[{"nodes":[{"data":{"id":"d_1","name:2EnvSensor","weight":60,"faveColor"
<?xml version=1.0?> <rdf:RDF xmlns="http://onem2m.ntt.local/xxx_interworking_ontology#"
EnvSensor
ManageDev
Switch
Interval
Binary
getData
CO2
integer
湿度
温度
float
インスタンス

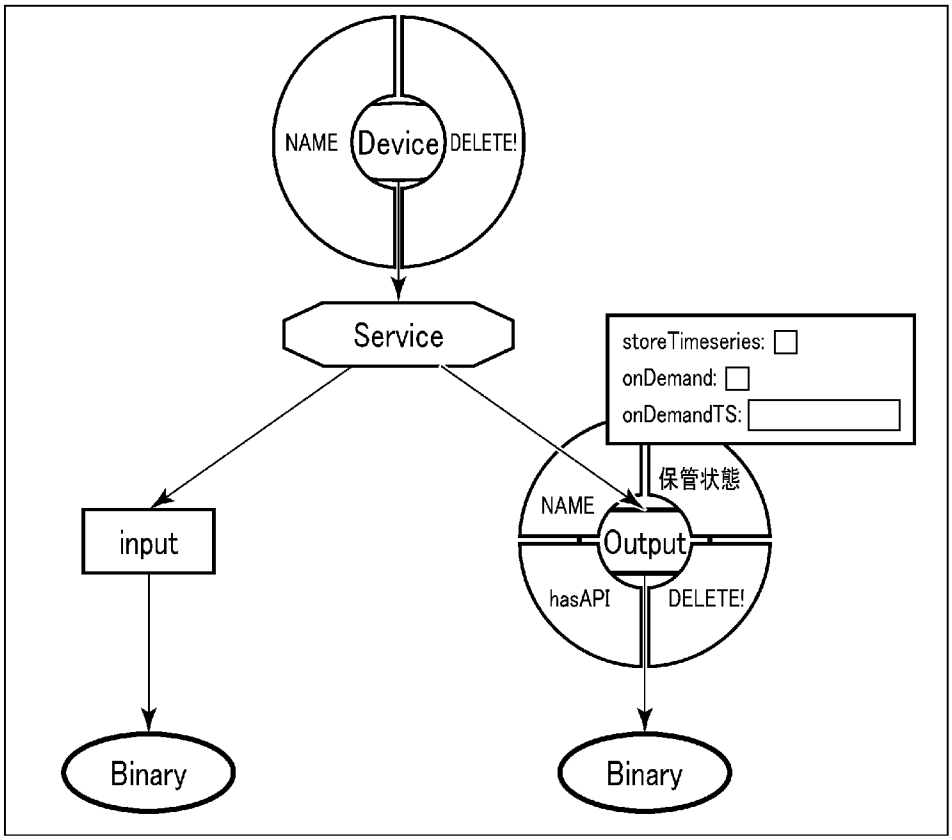
[図9]


Device
Service
InputDataPoint
OutputDataPoint
SimpleDataType
{"elements":[{"nodes":[{"data":{"id":"d_1","name:2EnvSensor","weight":60,"faveColor"
<?xml version=1.0?> <rdf:RDF xmlns="http://onem2m.ntt.local/xxx_interworking_ontology#"
線を引けない
Device
Service
Output
Binary
2本以上線を引けない
input
Binary

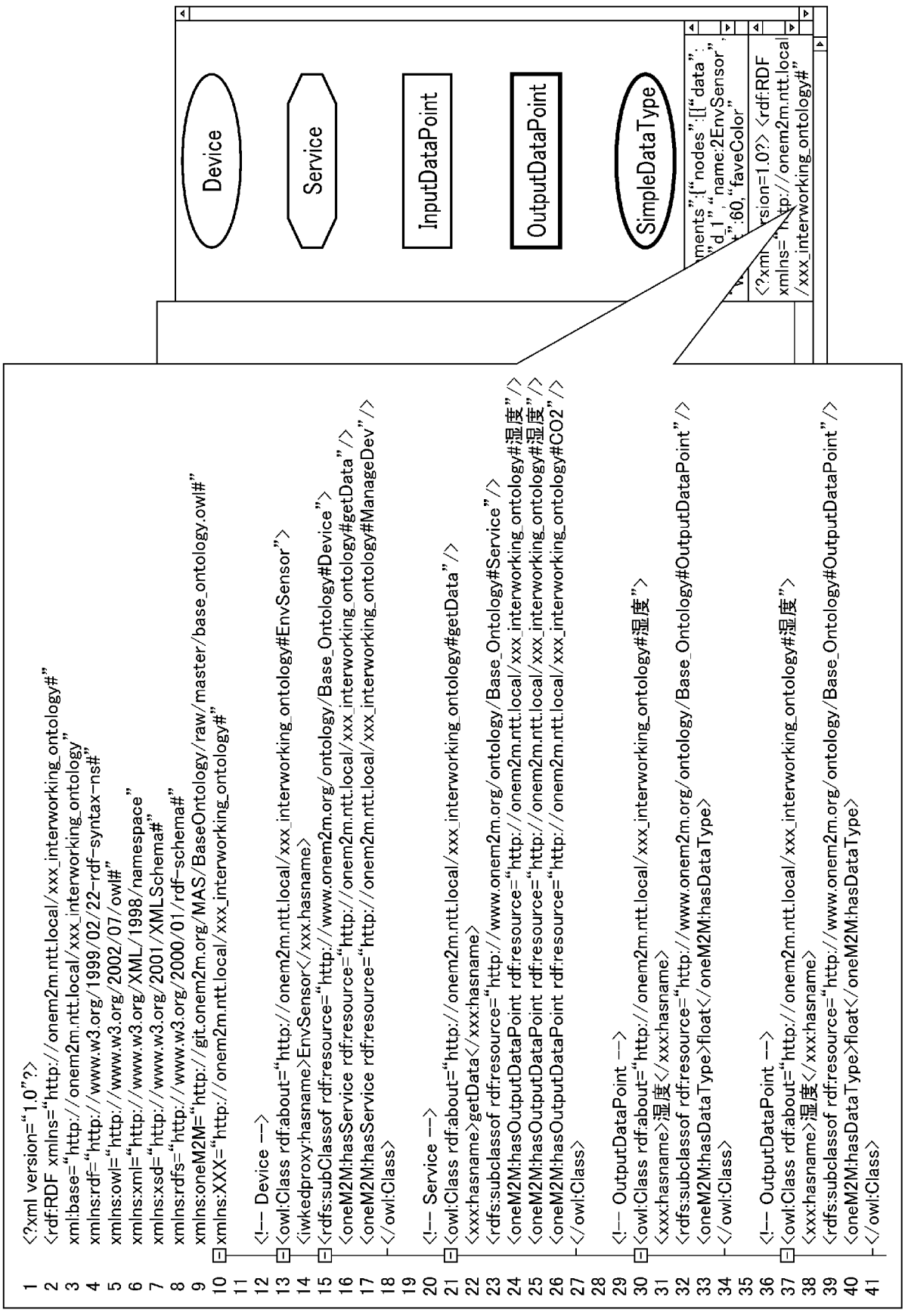
[図10]


NAME
Device
DELETE!
Service
input
Output
Binary
Binary

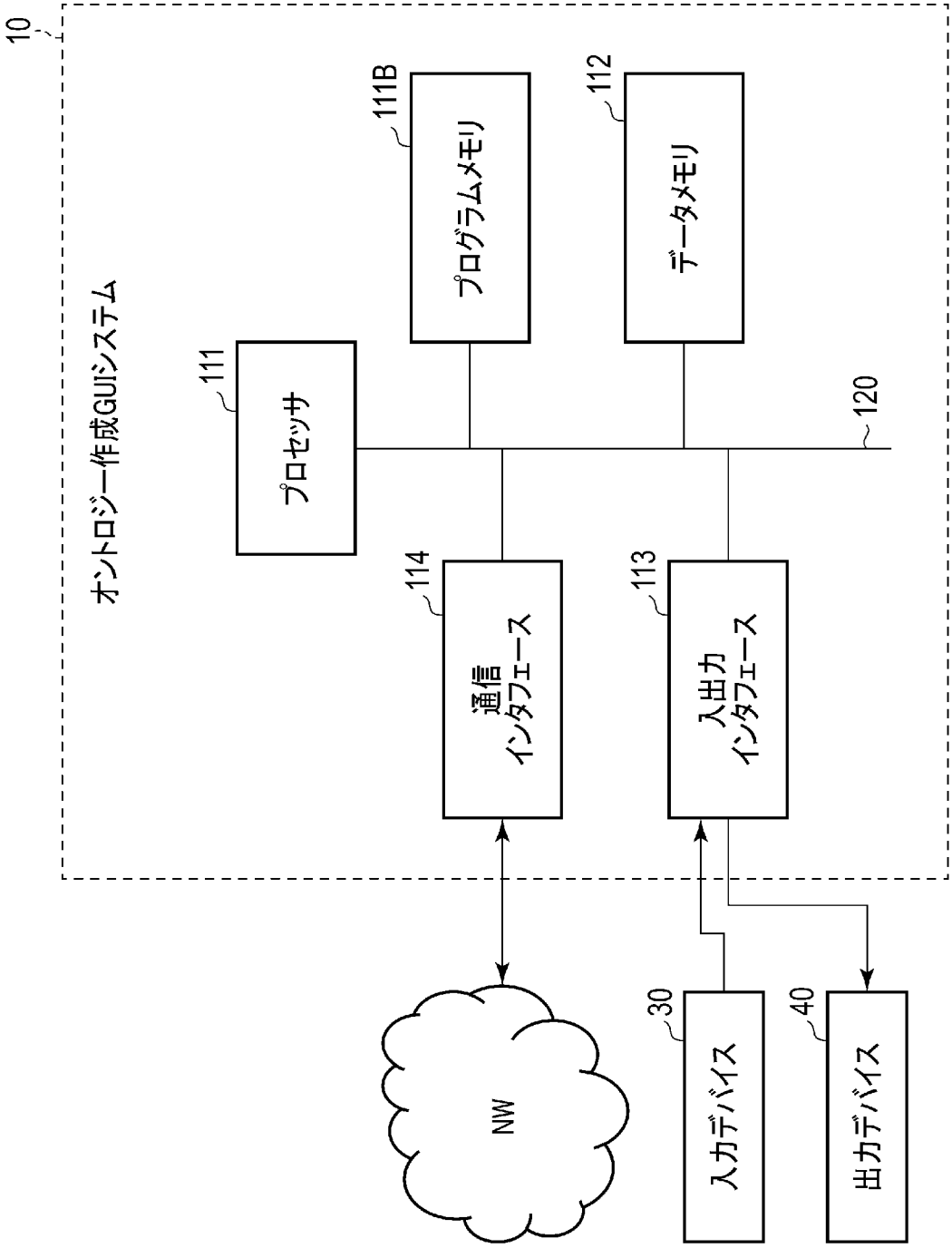
[図11]


NAME
Device
DELETE!
Service
storeTimeseries:
onDemand:
onDemandTS:
保管状態
NAME
Output
hasAPI
DELETE!
input
Binary
Binary

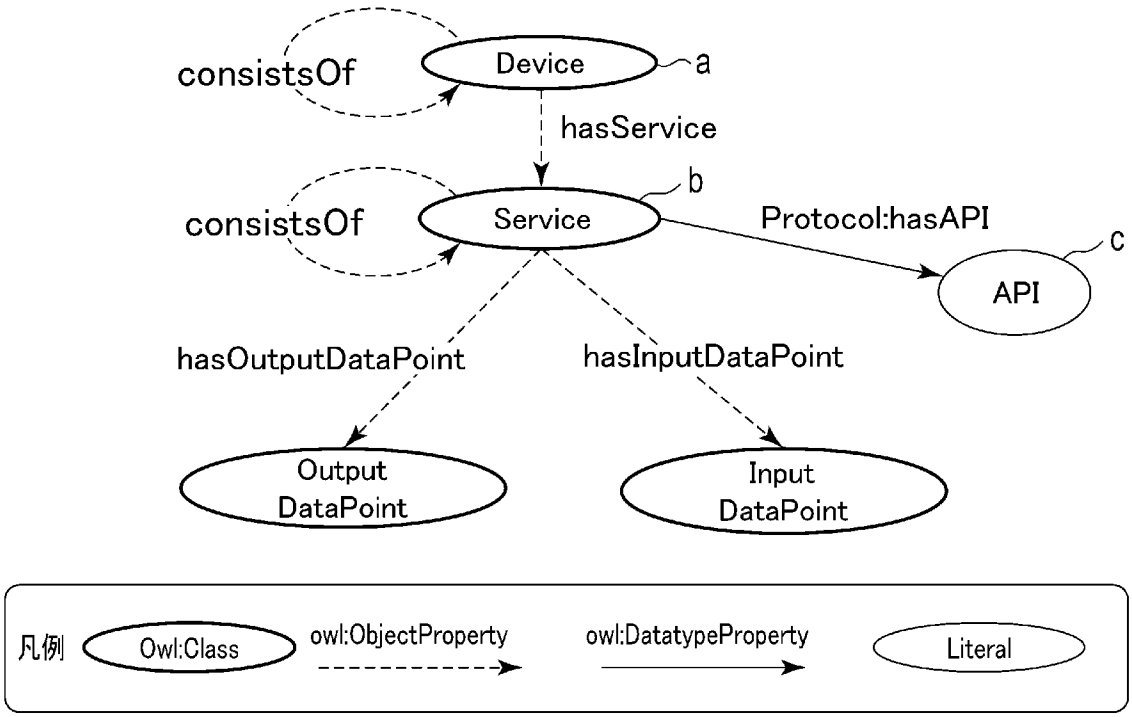
[図12]


Device
Service
InputDataPoint
OutputDataPoint
SimpleDataType
ments":[{"nodes":[{"data":
"d_1","name:2EnvSensor",
":60,"faveColor"
rsion=1.0?> <rdf:RDF
xmlns="http://onem2m.ntt.local
/xxx_interworking_ontology#"
<?xml version="1.0"?>
<rdf:RDF xmlns="http://onem2m.ntt.local/xxx_interworking_ontology#"
xml:base="http://onem2m.ntt.local/xxx_interworking_ontology"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:oneM2M="http://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#"
xmlns:XXX="http://onem2m.ntt.local/xxx_interworking_ontology#"
<!-- Device -->
<owl:Class rdf:about="http://onem2m.ntt.local/xxx_interworking_ontology#EnvSensor">
<iwkedproxy:hasname>EnvSensor</xxx:hasname>
<rdfs:subClassof rdf:resource="http://www.onem2m.org/ontology/Base_Ontology#Device">
<oneM2M:hasService rdf:resource="http://onem2m.ntt.local/xxx_interworking_ontology#getData" />
<oneM2M:hasService rdf:resource="http://onem2m.ntt.local/xxx_interworking_ontology#ManageDev" />
</owl:Class>
<!-- Service -->
<owl:Class rdf:about="http://onem2m.ntt.local/xxx_interworking_ontology#getData" />
<xxx:hasname>getData</xxx:hasname>
<rdfs:subclassof rdf:resource="http://www.onem2m.org/ontology/Base_Ontology#Service" />
<oneM2M:hasOutputDataPoint rdf:resource="http://onem2m.ntt.local/xxx_interworking_ontology#湿度" />
<oneM2M:hasOutputDataPoint rdf:resource="http://onem2m.ntt.local/xxx_interworking_ontology#湿度" />
<oneM2M:hasOutputDataPoint rdf:resource="http://onem2m.ntt.local/xxx_interworking_ontology#CO2" />
</owl:Class>
<!-- OutputDataPoint -->
<owl:Class rdf:about="http://onem2m.ntt.local/xxx_interworking_ontology#湿度">
<xxx:hasname>湿度</xxx:hasname>
<rdfs:subclassof rdf:resource="http://www.onem2m.org/ontology/Base_Ontology#OutputDataPoint" />
<oneM2M:hasDataType>float</oneM2M:hasDataType>
</owl:Class>
<!-- OutputDataPoint -->
<owl:Class rdf:about="http://onem2m.ntt.local/xxx_interworking_ontology#湿度">
<xxx:hasname>湿度</xxx:hasname>
<rdfs:subclassof rdf:resource="http://www.onem2m.org/ontology/Base_Ontology#OutputDataPoint" />
<oneM2M:hasDataType>float</oneM2M:hasDataType>
</owl:Class>

[図13]


10
オントロジー作成GUIシステム
111
プロセッサ
111B
プログラムメモリ
112
データメモリ
120
114
通信
インタフェース
113
入出力
インタフェース
NW
30
入力デバイス
40
出力デバイス

[図14]


consistsOf
Device
a
hasService
consistsOf
Service
b
Protocol:hasAPI
API
c
hasOutputDataPoint
hasInputDataPoint
Output DataPoint
Input DataPoint
凡例
Owl:Class
owl:ObjectProperty
owl:DatatypeProperty
Literal

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2019/048281

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl. G06F8/34(2018.01)i, G06F16/36(2019.01)i
FI: G06F16/36, G06F8/34

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl. G06F8/34, G06F16/36

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan	1922-1996
Published unexamined utility model applications of Japan	1971-2020
Registered utility model specifications of Japan	1996-2020
Published registered utility model applications of Japan	1994-2020

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	沈 偉, "オントロジーWiki サイトの構築", 電子情報通信学会技術研究報告 AI2008-1~15, 23 June 2008, vol. 108, no. 119, pp. 7-14, particularly, p. 8, left column to p. 10, left column, (SHIN, I., "Building of ontology Wiki", IEICE Technical Report AI2008-1~15)	1-6
A	JP 2001-67224 A (FUJITSU LIMITED) 16 March 2001, paragraphs [0048]-[0069], fig. 5-9	1-6
A	近藤 恵一, "オントロジーに基づく業務モデリングとシステム開発", 電子情報通信学会技術研究報告 KBSE2006-68~84, 21 March 2007, vol. 106, no. 618, pp. 25-28, particularly, p. 25, left column to p. 28, left column, (KONDO, Keiichi, "Business modeling and system development based on ontologies", IEICE Technical Report KBSE2006-68~84)	1-6

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search 09.01.2020	Date of mailing of the international search report 21.01.2020
Name and mailing address of the ISA/ Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/JP2019/048281

JP 2001-67224 A 16 March 2001 (Family: none)

国際調査報告

国際出願番号
PCT/JP2019/048281

A. 発明の属する分野の分類（国際特許分類（ＩＰＣ））

G06F 8/34(2018.01)i; G06F 16/36(2019.01)i
FI: G06F16/36; G06F8/34

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

G06F8/34; G06F16/36

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２０２０年
日本国実用新案登録公報	１９９６－２０２０年
日本国登録実用新案公報	１９９４－２０２０年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献のカテゴリー*	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
A	沈 偉，”オントロジーＷｉｋｉサイトの構築”，電子情報通信学会技術研究報告 Ａ Ｉ２００８－１～１５，2008.06.23，第１０８巻，第１１９号，，ｐ．７－１４ 特に第８ページ左欄－第１０ページ左欄	1-6
A	JP 2001-67224 A (富士通株式会社) 16.03.2001 (2001-03-16) 段落[0048]-[0069]，図5-9	1-6
A	近藤 恵一，”オントロジーに基づく業務モデリングとシステム開発”，電子情報通信学会技術研究報告 KBSE2006－68～84，2007.03.21，第１０６巻，第６１８号，ｐ．２５－２８ 特に第25ページ左欄－第28ページ左欄	1-6

☐ Ｃ欄の続きにも文献が列挙されている。

☑ パテントファミリーに関する別紙を参照。

＊ 引用文献のカテゴリー
“A” 特に関連のある文献ではなく、一般的技術水準を示すもの
“E” 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
“L” 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
“O” 口頭による開示、使用、展示等に言及する文献
“P” 国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
“T” 国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの
“X” 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
“Y” 特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
“&” 同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
09.01.2020	21.01.2020

名称及びあて先
日本国特許庁(ISA/JP)
〒100-8915
日本国
東京都千代田区霞が関三丁目４番３号

権限のある職員（特許庁審査官）
後藤　彰 5N 4226

電話番号 03-3581-1101 内線 3586

国際調査報告
パテントファミリーに関する情報

国際出願番号
PCT/JP2019/048281

引用文献	公表日	パテントファミリー文献	公表日
JP 2001-67224 A	16.03.2001	（ファミリーなし）	