オントロジーを活用したJavaFXプログラミング

永田　清

大東文化大学　経営学部

nagata@ic.daito.ac.jp

キーワード　オントロジー，SPARQL，JavaFX プログラミング

1　はじめに

Tim Berners-Lee 等によって提唱された Semantic Web[2] の概念は，Web 上の様々な情報を機械的に処理する枠組みを提供し，それによって大量かつ多様なデータの意味を考慮した処理システムの開発を可能にするものである [3]. 全体のレイヤー構成を表すものは Semantic Web Layer Cake，または Semantic Web Layer Stack などと呼ばれている（図 1).

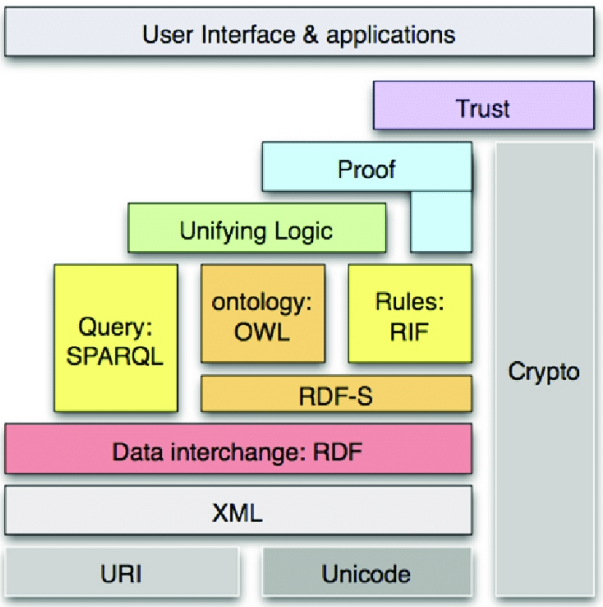

図 1:　Semantic Web ”Layer Cake” (Berners-Lee & Swick, 2006)

Layer Cake において，最下層の URI(Uniform Resource Identifier) や XML(eXtensible Markup Language) などは，統一された情報の表現法であり，アプリケーション作成のために必要な技術は中間層に示され，これらは W3C によって標準化されている．RDF(Resource Description Framework) はいわゆる Triple Graph 形式で情報を表現するための枠組みを提供し，そのための基本言語は RDFS(RDF Schema) によって与えられる．RDFS を拡張し，RDF 構文に対しより深い意味を与えるものとして OWL(Web Ontology Language) があり，論理表現に基づいて Semantic Web の有理性などを提供している．それらは N-Triples 形式として記述されるが，より簡潔な Turtle 形式なども使われる．

Turtle やその他の形式で表されたオントロジーファイルは，Web 上の様々な場所に存在し，巨大なデータベースと考えることが出来るが，それらから必要とする情報を取得するには，RDB(Relational Data Base) に対する SQL(Structured Query Language) に相当する問合せ言語が必要になる．SPARQL(SPARQL Protocol and RDF Query Language) は，その為に開発され W3C の RDF Data Access Working Group (DAWG) によって標準化された言語であり，いくつかのアプリケーション開発環境から利用することが出来る．

本研究では，JavaFX による Window アプリケーションプログラムにおいて，SPARQL を介してオントロジーファイルを操作し，結果を表示する一連の流れを確認し，実際に情報セキュリティ分野のアプリケーションプログラムを作成する為の課題を明らかにする．

全体の流れは次のようになっている．次節では，オントロジーに関する一般的な事柄を示し，作成のためのソフトウェアなどについて述べる．3 節では，SPARQL の簡単な説明と例を示す．4 節では，JavaFX の説明と，SPARQL 対応の Java 言語 API ライブラリである Apache Jena について，その概要と注意点を述べる．5 章では，実際にオントロジーを利用した Window アプリケーションプログラムを作成し，その結果を考察する．最終節は，結論と今後の課題である．

2　オントロジー (Ontology)

この節では，オントロジーの概念と，既存のオントロジーおよび実際に作成するためのソフトウェアなどを紹介する．

2.1 オントロジーの概念

“オントロジー”という言葉は哲学から借用されたもので「存在の体系的な説明」と定義されているが，Thomas R. Gruber [4] によれば，知識ベースシステムにおける「存在する」とはまさに表現できるものであり，その意味で「オントロジーとは概念化の明示的な仕様である」と考えられる．

“An ontology is an explicit specification of a conceptualization.”

また，オントロジーの開発における一般的な目標の 1 つは，人々やソフトウェアエージェントの間で情報の構造に関する共通の理解を共有することである．Natalya Noy と Deborah McGuinness はオントロジー開発のガイド [10] を発表したが，そこではオントロジーは，ドメイン内で情報を共有する必要がある研究者のための共通語彙として定義され，ドメイン内の基本概念とそれらの間の関係の機械解釈可能な定義が含まれる．また彼らは，オントロジーを開発する理由として以下のようなものを挙げている．

- 人々やソフトウェアエージェントの間で情報の構造に関する共通の理解を共有する
- ドメイン知識の再利用を可能にする
- ドメインの仮定を明示的にする
- ドメイン知識を運用知識から分離する
- ドメイン知識を分析する

Hendler [5] は，セマンティックウェブの観点から，オントロジーが一連の知識セットを提供することに注目し，特定のタスクやドメインに関する語彙や簡単な推論規則を含む基本概念の集合であると考えた．

2.2 既存オントロジーと作成

オントロジーは，世の中に存在する知識や語彙を PC などの機械が処理するために，概念を定式化し，インターネット上を介するアクセスと処理を可能にする．実際，さまざまな種類のオントロジーが作成されており，汎用的なものは想定したドメインに特化するように改変し，再度公開することで再利用を促している．

Dublin Core[1] は Web 上のリソースのメタデータを記述するもので，1995 年に米国ダブリンで開発され，電子書籍やジャーナルなどのタイトル，著者，日付，キーワード，言語といった書誌情報やファイル形式を記述するための基本プロパティを含んでいる．

FOAF(Friend Of A Friend)[2] は，その名称からも分かるように人やグループなどに関する情報を記述するもので，人物，グループ，組織といったクラスと，それらの属性を表す名前，勤務先，参加したプロジェクト，ホームページなどをそのプロパティとして含んでいる．

CContology(Customer Complaint Ontology)[3] は，オンラインビジネスなどでカスタマーサポートに対応するために開発されたもので，苦情問題，苦情解決，苦情申立人，苦情受理者,「ベストプラクティス」などの苦情処理ルールの分類などから構成され，ヨーロッパの 11 言語に対応している．

LKGOntology(Legal Knowledge Graph Ontology)[4] は，Lynx ドキュメント と呼ばれる法務知識グラフに含める価値のあるコンプライアンス関連のドキュメントを表現する目的で，ELI (European Legislation Identifier) のメタデータ要素を多用して作成されている．

医学や生理学間関連のオントロジーも多く開発され，有償のものが多いようであるが，公開されたいくつかの論文 [11] からオントロジーを作成することが可能である．また BioPortal サイト[5] には，さまざまな種類のオントロジーが登録されているので，ここから目的のものをダウンロードすることもできる．例えば，CARO(The Common Anatomy Reference Ontology) は，さまざまな種類の既存の解剖学オントロジー間の相互運用性を促進するために開発され，新しい解剖学オントロジーを構築するためのテンプレートを提供している．

組織構造や情報セキュリティ管理の分野でのオントロジーと関連する論文と実際のオントロジーをいくつか紹介する．

Edlira Kalemi と Edlira Martiri は [7]，FOAF オントロジーを基に学術機関の人やコミュニティとそれらの性質を考察し，オントロジーを作成した[6]．

情報リスクに対する IT 資産オントロジーについて，A. Kayode Adesemowo 等は情報資産を “人”，“ネットワーク”，“サービス”，“データ”，“ハードウェア”，“ソフトウェア”，および “情報” に分割して，作成するオントロジーを提案している [1]．

Sweden, Linköpings 大学の Almut Herzog 等は，資産，

[1] https://www.dublincore.org/

[2] http://xmlns.com/foaf/spec/

[3] https://www.jarrar.info/CContology/

[4] https://lynx-project.eu/doc/lkg/

[5] https://bioportal.bioontology.org/ontologies/

[6] https://vocab.org/aiiso/

脅威，脆弱性，対策の他，機密性や整合性といったセキュリティ目標達成のための防御戦略を含む “Cyber Security Ontology” を提案した [6]．これらを参照して作成された 4 種類の OWL 形式オントロジーがネット上に置かれている[7]．

既存のオントロジーを組合せ，修正や追加を行うことで独自のオントロジーを作成するためのアプリケーションソフトウェアとして Protégé があり，Free Open Source としてダウンロード可能である[8]．また，Natalya F. Noy 等は Protégé を使ったオントロジー作成の基本をガイドブック [10] にまとめており，そこではワインに関するオントロジー作成を例として取り上げている．図 2 は，“Cyber Security Ontology” を取り込んで，クラスの第 3 レベルまでを表示し，それらの関係を OntoGraf によって表示したものである．矢印の多くはサブクラス関係を表しているが，別の Property を表すものもあり，マウスを使ってその内容を表示できる．

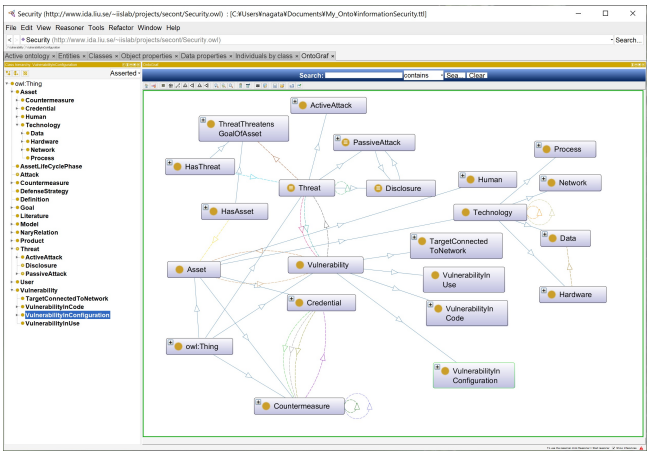

図 2: Protégé に Cyber Security Ontology を取り込んだクラス階層

図 3 は Threat と Vulnerability の関係を表した Triple Graph で，下の黒点線は以下のような Turtle 形式 RDF に対応している．

```
@prifix : <http://www.ida.liu.se/~iislab/
projects/secont/Security.owl#>.
  :Vulnerability :enablesThreat :Threat .
```

このように，実際に作成されるオントロジーには，階層構造を持ったクラスの集合，属性を表すプロパティ(Slot)，クラスのインスタンス (インディビジュアル) などがあり，それらが主語，述語，目的語の組である Triple Graph によって表されている．

[7]https://www.ida.liu.se/divisions/adit/security/projects/secont/
[8]https://protege.stanford.edu/

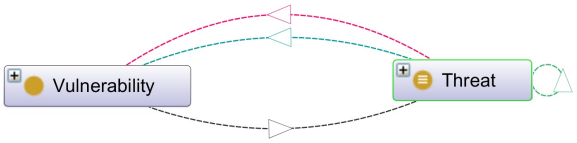

図 3: Threat と Vulnerability の関係

オントロジー作成手法に関しても，汎用的なもの [8] や各ドメイン (領域) ごとのものなどが論文で示されているが，基本的には以下の手順で行うことになる．

1. 対象とするドメインを決定する
2. どのような問いに答えるものを作成するかを決めるために，所謂 Conpetency Question を作成する
3. クラスとそれらの階層構造を決定する
4. 述語に対応するプロパティを，定義と地域を含めて作成する
5. オントロジー全体に論理的矛盾がないかを確認する
6. クラスのインスタンスを作成する
7. 作成したオントロジーを Trutle 形式などのファイルに保存する

上記 3 以降は Protégé において比較的簡単に作成することができ，特に論理的矛盾やサブクラスの定義域設定などは Reasoner と呼ばれる機能を使うことで達成できる．

残念ながら Protégé の一般的な使い方を詳しく解説した本は見当たらないが，GitHub にダウンロード，インストール，およびオントロジー作成手順を含めた解説があり，Youtube へのリンクも張られている[9]．また，Youtube 上にはその他の解説動画[10]もあり，英語ではあるが基本的な操作は分かるようになっている．

3 SPARQL

オントロジーは Triple Graph による RDF モデルという一種のデータベースであるが，そこから必要なものを取り出すためには問合せ言語 (Query Language) による処理が必要である．SPARQL は，それを可能にするもので，W3C で標準化されている[11]．基本的には SELECT で抽出する Triple の要素を，WHERE で Triple に対す

[9]https://protegeproject.github.io/protege/getting-started/
[10]https://www.youtube.com/watch?v=bpjMYBc98bk
[11]https://www.w3.org/TR/rdf-sparql-query/

る条件を記述し，CONSTRUCT を使って新しい Triple を生成する．図 4 のような Triple Graph の集まりに対応する RDF は次のようになる．(_:a のように記されているものは名前付けされない匿名の空白ノードであり，またインスタンスは四角で表されている．)

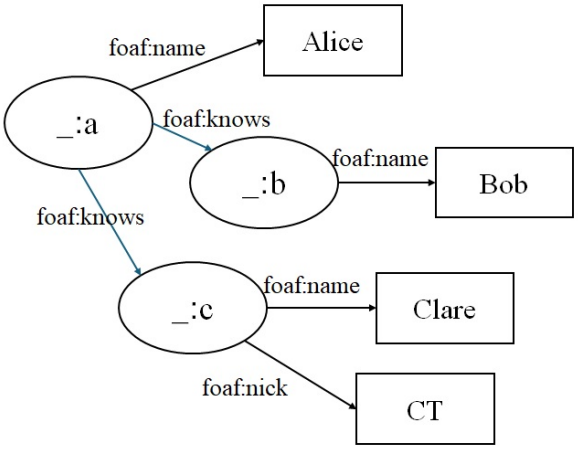

図 4: 空白ノードを含む Triple Graph

```
@prifix foaf:<http://xmlns.com/foaf/0.1/> .
   _:a    foaf:name   "Alice" .
   _:a    foaf:knows  _:b .
   _:a    foaf:knows  _:c .
   _:b    foaf:name   "Bob" .
   _:c    foaf:name   "Clare" .
   _:c    foaf:nick   "CT" .
```

上記 RDF から名前 (name) とニックネーム (nick) を抽出する SPARQL 文は次のようになる．

```
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?nameX ?nameY ?nickY
WHERE
  { ?x foaf:knows ?y ;
       foaf:name ?nameX .
    ?y foaf:name ?nameY .
    OPTIONAL { ?y foaf:nick ?nickY }
  }
```

ここで，変数は “?” で始まり，OPTIONAL は条件が成り立つ場合，つまり?y に対応する空白ノードがニックネームを持つ場合にだけ，その値を変数?nickY に入れる．この他にも FILTER, BIND, VALUES, LIMIT, ORDERED BY, DISTINCT, GROUP BY, SUM, COUNT などいくつかの用語や関数があるが，Java などでプログラムを作成する場合は，取り込んだデータをそちらで処理することも出来る．

4 JavaFX プログラミングと Apache Jena

Turtle 形式のオントロジーがあり，そこから必要なデータを問合せる SPARQL が作成できれば，それを Java の Window アプリケーションに落とし入れることになる．Java における Window アプリケーション作成には，Swing か JavaFX ライブラリを用いることが多いが，私見では JavaFX の方が使い勝手が良いようである．ここでは統合環境として Eclipse を使うが，JavaFX は JDK11 以降 JDK(Java Development Kit) から削除されてしまい Java 自体の Versoin が新しいとそのままでは使えない．一方，Java から SPARQL のコマンドを動かすには Apache Jena を使うことになるが，こちらは古い Version の Java では動かない．

解決手段として，OpenJFX を外部ライブラリとして使うこととし，その際の若干の注意事項を以下の 4.1 に示す．

4.1 OpenJFX の環境構築

まず，JavaFX ホームページ (https://openjfx.io/) の “Download” をクリックすることによって，各プラットホームに対応した SDK(zip ファイル) をダウンロードして適当な場所に解凍する．Eclipse を起動してプロジェクトを作成したら，$[Window] \rightarrow [Preferences] \rightarrow [Java - BuildPath - UserLibraries - New...]$ の手順で UserLibrary(“JavaFX”) を新しく作成する．次に，このライブラリに $[AddExternalJARs...]$ からダウンロードしたフォルダの “lib” フォルダに移動し，全ての “jar” ファイルをライブラリに取り込んで適応させる．

プロジェクトを右クリックして表示されるプルダウンから $[BuildPath] \rightarrow [ConfigureBuildPath...]$ で表示されるウィンドの “Libraries Path” タブから “Classpath” を選び，$[AddLibrary] \rightarrow [UserLibrary]$ から JavaFX を登録する．(“Classpath” でうまくいかない場合は “ModulePath” で試す)

JavaFx 対応のクラスを作成し，$[Run] \rightarrow [Configurations...]$ の “(x)=Arguments” タブにおける “VM arguments:” に以下を追加する．(ただし” ” 内は JavaFx のライブラリフォルダの場所を指定)

```
--module-path "C:\Program Files\
Java\javafx-sdk-23.0.1\lib" --add-modules
 javafx.controls,javafx.fxml
```

4.2 Apach Jena の環境構築とオントロジー利用

Apache Jena の環境構築も JavaFX の場合と同様であり，Apache Jena ホームページ (https://jena.apache.org/) から “Download” をクリックし，各プラットホームに対応した Jena zip ファイル (2025 年 3 月現在 “jena-5.3.0-source-release.zip”) をダウンロードして適当な場所に解凍する．以下ライブラリの作成から “Classpath” への登録まで同様の手順で行うことができる．

実際のプログラム作成に関してはページ https://jena.apache.org/tutorials/rdf_api.html#preface を参照すればよいが，例として以下に第 3 節で示した名前とニックネームを取得するものを示す．

```
public static void main(String[] args) {
  FileManager.get().addLocatorClassLoader(
    ⇒ JenaTest.class.getClassLoader());
  Model model=FileManager.get().loadModel
      ⇒ ("C:\\short.ttl");
  String queryStr=genQueryStr(
     ⇒  "C:\\test.rdf");
  Query query=QueryFactory.create(queryStr);
  QueryExecution qexec=QueryExecutionFactory.
         ⇒  create(query, model);
  try {
   ResultSet results=qexec.execSelect();
   while(results.hasNext()) {
    QuerySolution soln=results.nextSolution();
// Resource org=soln.getResource("x");
    Literal nameX=soln.getLiteral("nameX");
    Literal nameY=soln.getLiteral("nameY");
    Literal nickY=soln.getLiteral("nickY");
    }
   }finally {
      qexec.close();
   }
}
```

基本的には，Model でオントロジーの Turtle ファイル (“C:short.ttl”) から “model” を，SPARQL ファイル (”C:test.rdf”) からの “queryStr” を使って Query クラスのオブジェクトを作成し，それらを QueryExecution のコンストラクタに与えてオブジェクト “qexec” と作る．

検索されたデータは Result クラスのオブジェクト “results” に取り込まれるので，QuerySolution のオブジェクト “soln” で順次取得してメソッド getLiteral(literal 要素の場合) で Literal オブジェクトに入れる．もしも，ここで返ってくるものがリソースならば Resource クラスのオブジェクトに入れることになる．Literal の場合も文字列として表示するときは .toString で文字列に変換することになる．

5 Window アプリケーションの作成

情報セキュリティポリシー作成がためにオントロジーを活用することを提案した [9]. 図 5 は，そのイメージであり，組織の特性に対応したオントロジーを読込み基本ポリシーテンプレートに反映させるように設計されている．反映させるデータは，組織構造，管理体制，責任の所在，および重要情報資産などであるが，今回は特に情報資産をオントロジーから抽出し，チェックボックスとして提示する部分を実装する．

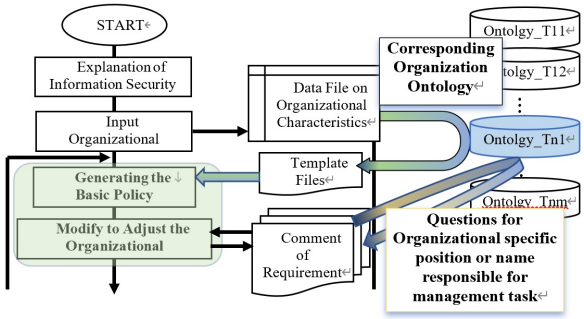

図 5: オントロジーを組込んだ情報セキュリティポリシー作成システム概念

テストケースなので，これまで作成してきたものとは分離して実行を検証することとした．図 6 は，初期画面で処理の流れを解説し，[*StartProgramme*] ボタンを押すと組織プロファイルのファイル指定を行うようになっている (図 7).

図 6: オントロジー処理のための初期画面

選択された組織プロファイルに対応するオントロジー Turtle ファイルが読込まれ，Asset の上位第 3 階層までの情報資産を取り出し，チェックボックスとして配置し

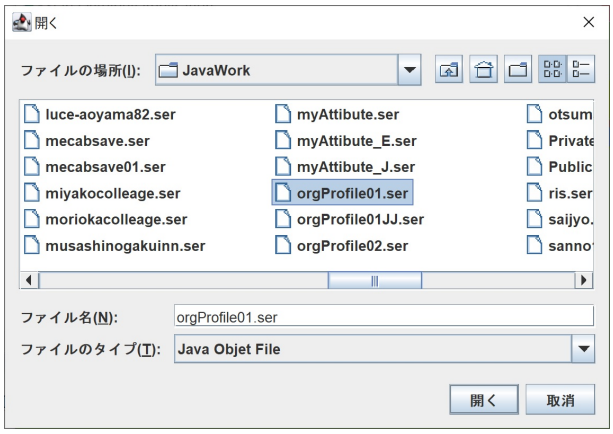

図 7: 組織プロファイルを選択

たものが図 8 であり，その上には組織プロファイルを表示している．

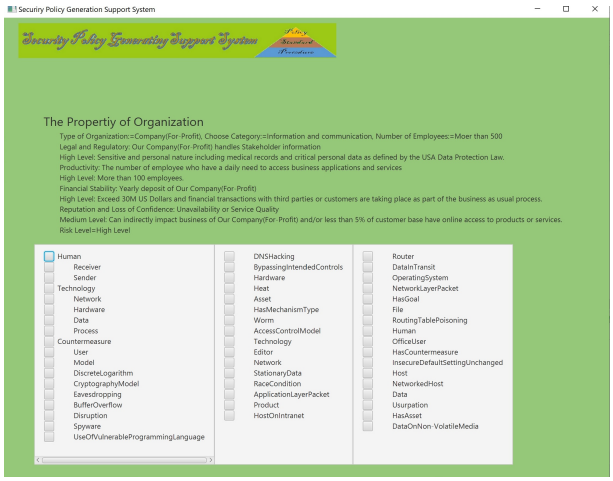

図 8: 情報資産から重要資産候補を選択

6 結論と今後の課題

オントロジーを活用した JavaFX によるウィンドウアプリケーションプログラム作成の流れを示し，実際に簡単なアプリケーションプログラムを作成した．ここで使ったオントロジーは，2.2 で述べた Cyber Security Ontology をそのまま読み込んだので，組織プロファイルにおける組織分類と直接関係がないものである．

また，図 2 のように，情報資産が Asset クラス以外と複雑に結びついていること，hasTreat プロパティによる関係設定が分かりにくいことなどから，思うような結果を表示してくれないことが分かった．これらのことと Competency Question を参考に，組織プロファイルは反映したオントロジーの作成が必要である．

参考文献

[1] Adesemowo, A.K., von Solms, R. and Botha, R.A., ITAOFIR: IT Asset Ontology for Information Risk in Knowledge Economy and Beyond. In: Communications in Computer and Information Science, vol 630, Springer, Cham. pp 173–187 (2016), https://doi.org/10.1007/978-3-319-51064-4_15

[2] Berners-Lee, T., Hendler, James, and Lassila, Ora, The Semantic Web, Scientific American. Vol. 284, no. 5. pp. 34–43 (2001)

[3] Cregan, A. M., Overview of Semantic Technologies, In book: Handbook of Ontologies for Business Interaction, pp. 1-20 (2007)

[4] Gruber, T. R., A Translation Approach to Portable Ontology Spcifications, Knowledge Acquisition, Vol. 5, (2), pp. 199-220 (1993), doi: 10.1006/knac.1993.1008

[5] Hendler, J., Agents and the Semantic Web, IEEE Intelligent System, 16(2), pp. 30-37 (2001)

[6] Herzog, A., Shahmehri, N., and Duma, C., An Ontology of Information Security. International Journal of Information Security and Privacy (IJISP), 1(4), pp. 1-23 (2007) , https://doi.org/10.4018/jisp.2007100101

[7] Kalemi, E., and Martiri, E., FOAF-Academic Ontology: A Vocabulary for the Academic Community, Proceedings of Third International Conference on Intelligent Networking and Collaborative Systems (Fukuoka, Japan, 2011), pp. 440-445 (2011)

[8] Nicola, A. D., Missikoff, M. Navigli, R., A software engineering approach to ontology building, Information Systems, Volume 34, Issue 2, pp. 258-275 (2009), ISSN 0306-4379, https://doi.org/10.1016/j.is.2008.07.002.

[9] Nagata, K., Automatic Generating System of Information Security Policy, Athens Journal of Technology and Engineering - Volume 10, Issue 4, pp. 227-236 (2023)

[10] Noy, N. F., and McGuiness, D. L., Ontology Development 101: A Guide to Creating Your First Ontology, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 (2001)

[11] 古崎 晃司, 国府 裕子, 周 俊, 今井 健, 大江 和彦, 溝口 理一郎, 「臨床医学オントロジーの構築とその基本思想」, 人工知能学会第二種研究会資料, 2008 巻, SWO-019 号, pp. 0901-0907, ISSN 2436-5556, https://doi.org/10.11517/jsaisigtwo.2008.SWO-019_09