

# オントロジーを利用した構造化文書 記述内容の妥当性検証に関する研究

乗松真二

2014 年

## 内容梗概

インターネットの普及にともなって電子申請や EDI が増加している。電子申請や EDI を行うシステムでは交換される XML 等の構造化文書データに対する文書構造のチェックを行う機能を有していることから、送信前に文書形式に関する問題は検出可能となっている。しかしながら文書形式上の問題はなく送信が成功したとしても、その記載内容に整合性が欠けている等の文書内容の妥当性に問題がある場合は、データ送信後に損害や不利益を生じる場合がある。このことから、申請や取引の文書内容が妥当であることを送信前に検証することが重要である。しかしながら、特に文書内容の妥当性の判断に専門的知識を必要とする分野においては、深い知識のルール化が難しいこともあり、文書内容の妥当性検証に関するコンピュータ支援が進んでいるとはいえない。

一方、次世代 Web 技術としてセマンティック Web が注目を集めている。セマンティック Web は、Web 上のリソースについてのメタデータを機械可読可能な情報として記述し、さらにメタデータの意味や相互関係を定義するオントロジーやルールを与えて高度な検索や推論等を行えるようにする技術である。セマンティック Web 関連技術として発展し、標準化が進むオントロジー技術は Web 以外にも応用が可能であると考えられる。

そこで本論文では専門的知識を必要とする分野における構造化文書内容の妥当性検証を行うことを目的として、オントロジーを利用した推論を行うことにより構造化文書内容の妥当性検証を行うシステムを提案する。さらに提案手法を応用して不動産登記申請の妥当性検証へ適用する。

以下、本論文の構成について述べる。

はじめに、第 1 章において、本研究の背景、問題、目的について述べる。

第 2 章では、本論文に関連する技術および関連研究として、セマンティック Web の

概要を述べ、その中から本論文に関係が深い技術について述べる．続いてオントロジーの定義を述べ、オントロジーの構築と法律分野の既存研究について述べる．最後に、代表的な構造化文書である XML とその妥当性検証に関する既存研究について述べる．

第3章では、本論文で提案するオントロジーを用いた構造化文書の妥当性検証システムの仕組み、オントロジーの構築方法およびシステムの実装、そしてそれらの評価について述べる．提案システムの特徴は、文書ドメインオントロジーと検証オントロジーの2つのオントロジーを利用した推論により構造化文書内容の妥当性を判断することである．また妥当かどうかの判断だけでなく、推論途中に導出される妥当と判断した根拠情報を含む結果オントロジーを検証結果と併せて出力する．これにより利用者に有用な情報を提供できる．このような推論を行うためのオントロジーを専門家と情報処理技術者が共同して構築するためのオントロジー構築方法と、そのシステムの実装方法を示す．そして、これらの評価を行うことで妥当性検証システムの有用性を示す．

第4章では、オントロジーを用いた構造化文書の妥当性検証システムを不動産登記申請内容の妥当性検証へ適用したシステムの構築と評価について述べる．妥当な不動産登記申請情報を作成するためには多くの法律知識を必要とするため、その多くは法律専門家によって行われている．近年、法律専門家が一度に扱う登記案件の量と複雑さが増大傾向にあることから、正確性や迅速性をより高めるために、法律専門家が行う不動産登記申請情報の妥当性検証の支援が求められている．そこで不動産登記申請情報の妥当性検証を目的とした申請オントロジー（文書ドメインオントロジー）と検証オントロジーの構築を行う．そして、具体的な不動産登記申請に対する妥当性検証システムの動作を評価することでその有用性を示す．

最後に、第5章で本論文の成果をまとめ、今後の課題と展望について述べる．

# 目次

|       |                            |    |
|-------|----------------------------|----|
| 第1章   | 序論 .....                   | 1  |
| 1.1   | 背景と目的 .....                | 1  |
| 1.2   | オントロジーを利用した妥当性検証システム ..... | 3  |
| 1.3   | 不動産登記申請の妥当性検証 .....        | 4  |
| 1.4   | 論文の構成 .....                | 5  |
| 第2章   | 関連研究 .....                 | 7  |
| 2.1   | 概要 .....                   | 7  |
| 2.2   | セマンティック Web .....          | 7  |
| 2.2.1 | 概要 .....                   | 7  |
| 2.2.2 | RDF .....                  | 10 |
| 2.2.3 | RDFS .....                 | 11 |
| 2.2.4 | OWL .....                  | 11 |
| 2.2.5 | SWRL .....                 | 12 |
| 2.3   | オントロジー .....               | 13 |
| 2.3.1 | オントロジーの定義と種類 .....         | 13 |
| 2.3.2 | 構築方法 .....                 | 14 |
| 2.3.3 | オントロジー開発ツール .....          | 15 |
| 2.3.4 | 法律分野のオントロジー .....          | 15 |
| 2.4   | XML 文書の妥当性検証 .....         | 16 |
| 2.4.1 | XML .....                  | 16 |
| 2.4.2 | スキーマ言語による妥当性検証 .....       | 16 |
| 2.4.3 | その他の手法による妥当性検証 .....       | 17 |
| 2.5   | まとめ .....                  | 18 |
| 第3章   | オントロジーを利用した妥当性検証システム ..... | 19 |
| 3.1   | 概要 .....                   | 19 |
| 3.2   | 妥当性検証システム .....            | 22 |
| 3.2.1 | 概要 .....                   | 22 |
| 3.2.2 | 文書ドメインオントロジー .....         | 24 |
| 3.2.3 | 検証データ .....                | 24 |
| 3.2.4 | オントロジーコンバーター .....         | 25 |
| 3.2.5 | 検証オントロジー .....             | 26 |
| 3.2.6 | 妥当性検証システムの動作 .....         | 30 |
| 3.3   | オントロジーの構築 .....            | 30 |

|        |                           |    |
|--------|---------------------------|----|
| 3.4    | システムの実装 .....             | 36 |
| 3.4.1  | システムのプログラム構造 .....        | 36 |
| 3.4.2  | オントロジーコンバーターの実装 .....     | 39 |
| 3.5    | 評価 .....                  | 42 |
| 3.6    | まとめ .....                 | 44 |
| 第4章    | 不動産登記申請の妥当性検証 .....       | 46 |
| 4.1    | 概要 .....                  | 46 |
| 4.2    | 不動産登記概要 .....             | 48 |
| 4.3    | 不動産登記申請妥当性検証システムの概要 ..... | 50 |
| 4.4    | 申請オントロジーと申請データ .....      | 52 |
| 4.4.1  | 申請オントロジーの構築 .....         | 52 |
| 4.4.2  | オントロジーコンバーター .....        | 55 |
| 4.4.3  | 申請データ .....               | 55 |
| 4.5    | 検証オントロジー .....            | 57 |
| 4.5.1  | OK 申請情報 .....             | 59 |
| 4.5.2  | 検証パターンの決定 .....           | 61 |
| 4.5.3  | 記載内容等に関する検証ルールの記事 .....   | 63 |
| 4.6    | 妥当性検証システムの動作 .....        | 68 |
| 4.7    | 評価と考察 .....               | 71 |
| 4.7.1  | 動作評価 .....                | 71 |
| 4.7.2  | オントロジーの詳細化と拡張 .....       | 75 |
| 4.8    | まとめ .....                 | 76 |
| 第5章    | 結論 .....                  | 78 |
| 参考文献   | .....                     | 81 |
| 謝辞     | .....                     | 84 |
| 研究業績一覧 | .....                     | 85 |

## 図目次

|                                    |    |
|------------------------------------|----|
| 図 2-1 Web とセマンティック Web の関係 .....   | 9  |
| 図 2-2 セマンティック Web のレイヤーケーキ .....   | 10 |
| 図 2-3 TOVE 方法論によるオントロジー開発手順 .....  | 14 |
| 図 3-1 妥当性検証システムの考え方 .....          | 21 |
| 図 3-2 妥当性検証システムの構成 .....           | 22 |
| 図 3-3 妥当性検証システムのコンポーネント概要 .....    | 36 |
| 図 3-4 JXML2OWL の概要 .....           | 40 |
| 図 3-5 JXML2OWL を利用した変換定義作成 .....   | 41 |
| 図 4-1 システム構築と運用 .....              | 47 |
| 図 4-2 不動産登記申請の妥当性検証システム概要 .....    | 50 |
| 図 4-3 申請オントロジーの例 .....             | 55 |
| 図 4-4 申請データ例 .....                 | 56 |
| 図 4-5 申請データのクラス推論例 .....           | 57 |
| 図 4-6 検証オントロジーの作成例 .....           | 58 |
| 図 4-7 OK 申請情報の定義例 .....            | 61 |
| 図 4-8 検証パターンの決定 .....              | 62 |
| 図 4-9 不動産の表示と義務者の検証例 .....         | 66 |
| 図 4-10 拡張ルール推論の例 .....             | 68 |
| 図 4-11 検証結果の例 .....                | 69 |
| 図 4-12 結果オントロジーのイメージ .....         | 70 |
| 図 4-13 妥当な場合の結果オントロジー取込例 .....     | 72 |
| 図 4-14 妥当とならない場合の結果オントロジー取込例 ..... | 73 |
| 図 4-15 問題が無い義務者情報の例 .....          | 74 |
| 図 4-16 問題がある義務者情報の例 .....          | 74 |

# 第 1 章 序論

## 1.1 背景と目的

本論文は専門的知識を必要とする分野における構造化文書内容の妥当性検証に関するものである。

インターネットの普及にともなって電子申請（インターネットを利用して申請・届出等の行政手続を行うこと）や EDI（Electronic Data Interchange, 電子データ交換：異なる組織間での取引のためのメッセージを標準的な規約を用いてコンピュータ間で交換すること）が増加している。ここでやり取りされるデータは XML（Extensible Markup Language）[1]に代表される構造化文書（タグ等を用いて文書の構造を表現する文書）で仕様定義されることが多い。電子申請や EDI を行うシステムでは交換されるデータに対する文書構造のチェックを行う機能を有していることから、送信前に文書形式に関する問題は検出可能となっている。しかしながら、文書形式上の問題はなく送信が成功したとしても、その記載内容に整合性が欠けている等の文書内容の妥当性に問題がある場合は、データ送信後に損害や不利益を生じる場合がある。例えば高額取引において意図と異なる契約が成立してしまった場合、取引が不成立となってしまった場合、あるいは権利に関する申請が却下となった場合等では大きな損害や不利益を生じる可能性がある。このことから、申請や取引の文書内容が妥当であることを送信前に検証することが重要である。

このような重要な申請や取引分野における文書内容の妥当性検証の重要性は電子申請や EDI に固有の問題ではなく、書面等を用いた従来からの手続きにおいても同様に重要な問題であった。従来から行われている書面による方法は、専門的知識をもつ人が

書面の作成から検証まで責任をもって行うことが多く、取引等の安心、安全性を確保していたといえる。しかし近年 IT インフラが様々な分野に広がっていくにつれ取引案件の増大、迅速化、複雑化が進んでいる。データ作成等の手続的処理においては IT インフラの普及や進歩によって効率化が進み電子申請や EDI システムを利用することで近年の社会的要請に答えているといえる。また文書内容の妥当性判断が深い知識を必要とせず機械的に検証可能なものについては、従来から研究されている文書ルール記述言語等[2][3][4]を導入して検証することで取引等の安心、安全も確保できる。しかしながら、特に文書内容の妥当性の判断に専門的知識を必要とする分野においては、深い知識のルール化が難しいこともあり、文書内容の妥当性検証に関するコンピュータ支援が進んでいるとはいえない。このためデータ作成処理が高速化されたとしても専門家等による人手による内容の妥当性判断には量的、質的、迅速性等の限界があり、取引等の安心、安全を確保した上で社会的要請に答えていくのは難しい。このような時代的背景からもコンピュータによる文書内容の妥当性検証支援に対する必要性が高まっている。

ところでインターネットの普及により Web を利用して誰でも簡単に情報を公開したり入手したりすることができるようになった。しかしながら、情報の氾濫ともいえる状況を生むことになり有益な情報を抽出することが難しいという新たな問題が生じている。このような問題を解決する次世代 Web 技術としてセマンティック Web[5][6]が注目を集めている。セマンティック Web は、Web 上のリソースについてのメタデータを機械可読可能な情報として記述し、さらにメタデータの意味や相互関係を定義するオントロジーやルールを与えて高度な検索や推論等を行えるようにする技術である。つまり Web に機械可読なメタデータを与えてその意味解釈を行うことで、より高度な Web 上の情報検索、統合、共有、再利用等の様々な応用を可能にしようというものである。ここでオントロジーとは、もともとは哲学を起源とする用語であるが、人工知能研究の立場からは「概念化（対象世界において興味をもつ概念とそれらの間の関係）の明示的な



規約」[7]と定義され、セマンティック Web が提唱される以前から新しい知識ベースあるいは知識処理技術として研究されてきたものである。これがセマンティック Web 関連技術として W3C (World Wide Web Consortium) によりオントロジー記述言語やルール記述言語等が仕様化されたこともあり、現在様々なオントロジーが活発に研究、開発され、利用されている。

セマンティック Web 関連技術として発展し標準化が進むオントロジー技術は Web 以外にも応用が可能であると考えられる。本論文が対象とする文書の妥当性検証の分野において、これまで書面による取引では紙面から文書内容の妥当性判断に必要となる情報を機械的に抽出することが困難であったが、電子申請や EDI で利用される構造化文書では、必要とされる情報の多くがタグ化され機械可読となっている。

このような状況から、セマンティック Web が Web に対して行うこと、つまりオントロジーを用いて Web 上の情報の意味を考慮した応用を行うことと同様に、電子申請や EDI が対象とする分野のオントロジーを用いることで構造化文書の意味を考慮した妥当性検証への応用が可能であると考えられる。そこで、本論文では専門的知識を必要とする分野における構造化文書内容の妥当性検証を行うことを目的として、オントロジーを利用した推論を行うことにより構造化文書内容の妥当性検証を行うシステムを提案する。さらに提案手法を応用して不動産登記申請の妥当性検証へ適用する。

## 1.2 オントロジーを利用した妥当性検証システム

専門的知識を必要とする分野における文書内容の妥当性検証を行うシステムを構築するためには以下の 2 つの課題を解決する必要がある。

一つ目の課題は、専門的知識を必要とする複雑な妥当性検証ルールを記述することである。一般に専門家が判断を行う分野に必要とされる知識は広く深い。専門的知識の概念整理を行うことなく検証ルールを記述しようとしてもルール数と複雑さが増大し、一

貫性のあるルールセットを作成することは難しい。仮に作成できたとしてもメンテナンスが困難である。そこで専門的知識とルールを体系的かつ柔軟に記述できる手法が必要とされる。

二つ目の課題は、特定分野の専門家と IT 技術者の共同作業を可能とすることである。IT 技術者は専門的知識の概念整理を行うことは難しいため、この作業は対象分野の専門家が行うこととなる。しかし専門家は情報処理知識をもっていることは少ないため専門家自身がルールを記述することは難しい。そこで専門家と IT 技術者がお互いの意図を正しく伝達でき、相互理解可能な共同作業のための基盤が必要となる。

そこで本論文では、文書ドメインオントロジーと検証オントロジーの2つのオントロジーを利用した推論により構造化文書内容の妥当性を判断するシステムを提案する[8]。さらに、専門家と情報処理技術者が共同してオントロジーを構築するための構築方法とシステムの実装方法を示す。オントロジーを主体とすることで、複雑な検証ルールを記述でき、専門家と情報処理技術者の継続的な共同作業を可能とした。

### 1.3 不動産登記申請の妥当性検証

不動産登記は不動産の現況と権利関係を登記簿に記録して公示する制度である。登記手続は、不動産取引の安全と権利の保護並びに登記の真正確保の観点から、申請情報に登記事項を記載するだけでなく、本人性や登記原因を証明する添付情報を提供する必要がある。このように妥当な登記申請情報を作成するためには、事実を証明する調査や多くの法律知識を必要とするため、その多くは法律専門家によって行われている。

近年、大型マンション建築による一度に大量の登記の発生、ゴルフ場のような多数の関与者が複雑に絡み合う案件等、扱う案件が大規模化、複雑化する傾向がある。このような登記案件の量、複雑さの増大により、登記に必要とされる正確性や取引を円滑に進めるための迅速性を確保するため、法律専門家が行う知的作業部分に対し、より高度な

支援が求められている。

一方、電子署名法の成立、不動産登記法の改正等により登記の電子申請が可能となり、法務省や民間ベンダからオンラインで登記申請を行うソフトウェアも提供され、コンピュータによる登記申請の支援の環境も整いつつある。しかしながら、これらの申請用ソフトウェアは、XML ファイル仕様等に基づくメッセージ形式やプロトコル仕様のチェックを行う機能は装備されるものの、申請情報の意味に踏み込んだ妥当性を検証できるものはない。結果、申請書作成支援ソフトウェアで申請情報をエラーなく送信しても、登記官の審査によって、補正や却下となる可能性がある。

そこで不動産登記申請情報の意味に踏み込んだ妥当性検証を行えるように、オントロジーを利用した構造化文書の妥当性検証システムを不動産登記申請の妥当性検証へ適用する[9]。このため不動産登記申請情報の妥当性検証を目的とした申請オントロジー（文書ドメインオントロジー）と検証オントロジーの構築を行う。そして具体的な不動産登記申請に対する妥当性検証システムの動作を評価する。

#### 1.4 論文の構成

以降、本論文の構成は次の通りである。

第2章では、本論文に関連する技術および関連研究として、セマンティック Web の概要を述べ、その中から本論文に関係が深い技術とツールについて述べる。続いてオントロジーの定義を述べ、オントロジーの構築と法律分野の既存研究について述べる。最後に、代表的な構造化文書である XML とその妥当性検証に関する既存研究について述べる。

第3章では、本論文で提案するオントロジーを用いた構造化文書の妥当性検証システムの仕組み、オントロジーの構築方法およびシステムの実装、そしてそれらの評価について述べる。

第4章では, オントロジーを用いた構造化文書の妥当性検証システムを不動産登記申請内容の妥当性検証へ適用したシステムの構築と評価について述べる.

最後に, 第5章で本論文の成果をまとめ, 今後の課題と展望について述べる.

## 第2章 関連研究

### 2.1 概要

本章では本論文に関連する技術および関連研究について述べる。はじめに，2.2節でセマンティック Web の概要を述べ，その中から本論文に関係が深い技術について述べる。2.3節でオントロジーの定義を述べ，オントロジーの構築と法律分野の既存研究について述べる。最後に，2.4節では代表的な構造化文書である XML とその妥当性検証に関する既存研究について述べる。

### 2.2 セマンティック Web

#### 2.2.1 概要

Web とは，正式には World Wide Web (WWW) と呼ばれ，インターネット上のハイパーテキストシステムのことである。

Web 上の文書(Web ページという)等のリソースは URL (Uniform Resource Locator) で特定され，Web ブラウザ等で閲覧できる。また，Web ページに別の Web ページの URL を埋め込むことをハイパーリンクといい，このリンクをたどることで，インターネット上に分散して存在する Web ページを相互に参照できる。

現在，Web ページの多くは HTML (Hyper Text Markup Language) とその表示方法を指示する CSS (Cascading Style Sheets) で作成され，他の HTML や画像，動画等とのハイパーリンクをもつ。HTML で記述する Web ページは，マークアップを用いて自然言語で記述する文章に対して段落や章題等の文書構造やフォント指定等の表示方法を与えることができる。しかしこれらのマークアップは人間が文書の理解を助ける

ための表示に関するものが中心であり、本質的には自然言語によるプレーンテキストといえる。このため人間が読んで意味を理解することはできても、人間が理解するような意味を機械が抽出することは難しい。またハイパーリンクにも意味づけが与えられていないため、ハイパーリンクで結ばれている Web ページがどのような関係にあるかは人間が見て解釈することはできても、機械的に処理することは困難である。さらに、画像や動画から意味のある情報を機械的に抽出することも困難である。このような現在の Web は人間が見て解釈することを前提としており、「文書の Web」ともいわれる。そのためインターネット上にある膨大な情報の高度な活用は限定的となっている。

これに対しセマンティック Web は、Web の創始者でもある Tim Berners-Lee によって提唱され、W3C (World Wide Web Consortium) が推進する Web 上の情報を機械的に処理するための枠組みである[5][6][10][11]。セマンティック Web は「文書の Web」に対して「データの Web」ともいわれる。

その基本的な考え方は、HTML 文書等の Web 上のリソースから機械的に意味を抽出するのではなく、Web 上のリソースについてのメタデータ (データについてのデータ) を機械可読可能な情報として記述し、さらにメタデータの意味や相互関係を定義するオントロジーやルールを与えて高度な検索や推論等を行えるようにする。つまり現在の Web にメタデータを与えてその意味解釈を行うことで、インターネット上にある膨大な情報の統合、共有、再利用等を可能にしようというものである。

## セマンティックWeb (データのWeb)

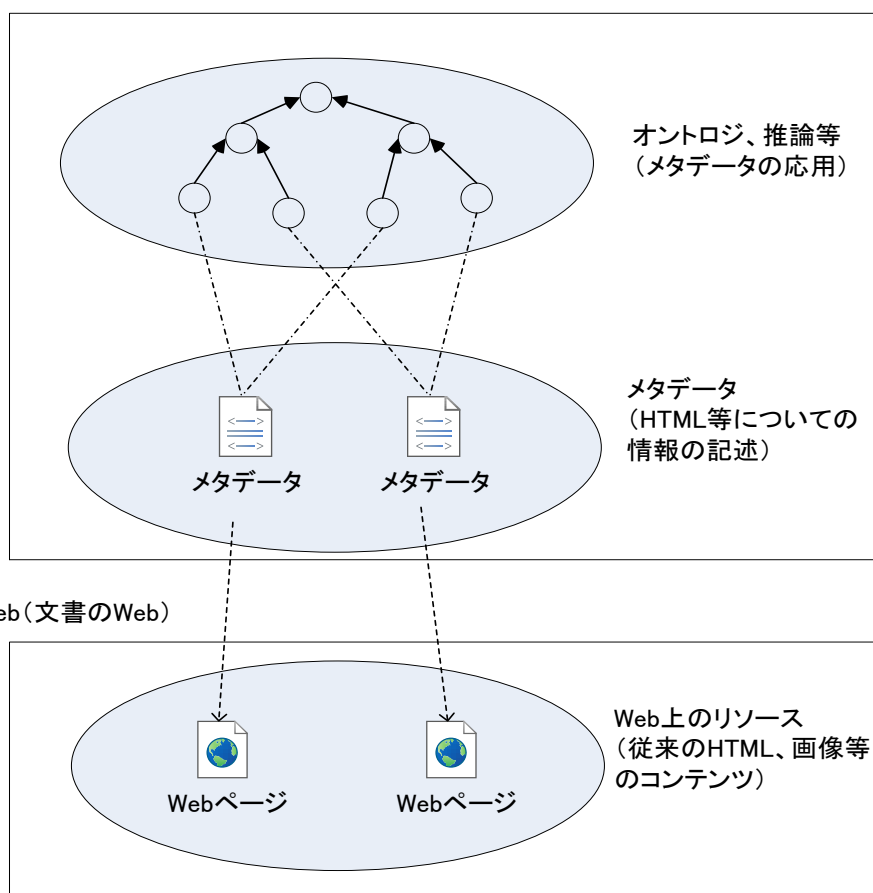
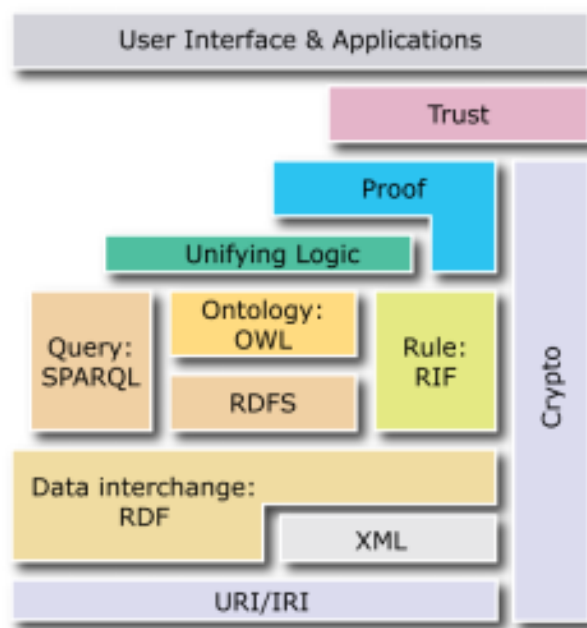


図 2-1 Web とセマンティック Web の関係

セマンティック Web の構想に関連する技術の標準化が W3C により行われている。この技術体系は現在の Web 技術に変更を加えるものではなく、意味を機械処理する技術をレイヤーとして加える形となっており、「レイヤーケーキ」と呼ばれる図で説明されることが多い。この階層において、上位の技術は下位の技術の変更を行うことなく追加される。



<http://www.w3.org/2007/03/layerCake.png>

図 2-2 セマンティック Web のレイヤーケーキ

以下では、本論文に関係が深い RDF, RDFS, OWL と SWRL について述べる。ここで SWRL はルールを記述する言語であり、レイヤーケーキの Rule に位置するものである。

### 2.2.2 RDF

RDF (Resource Description Framework) [12][13]は、リソースに関する情報を表わすための枠組みである。ここでリソースとは、ドキュメント、人間、物体、抽象的な概念を含む様々なものであり、IRI (International Resource Identifier) で同定する。

RDF はリソースを<主語> <述語> <目的語>の三つ組の構造体で記述する。これはトリプル (Triple) と呼ばれる。トリプルのうち、主語には IRI が記述され、リソースを同定する。目的語には値には文字列や数字などのリテラル値またはその他のリソースへの関係を表す IRI が記述される。目的語に IRI を指定した場合は他のトリプルの主語に



関係づけられることになる。述語には主語と目的語の関係の性質を表す IRI が記述される。この関係は、主語から目的語への方向性を持ち、RDF ではプロパティと呼ばれる。

### 2.2.3 RDFS

RDF が Web 内の情報を記述するために開発された汎用目的言語であるのに対し、RDFS (RDF Schema) [14]は RDF の語彙を記述することを目的とした RDF 語彙記述言語である。RDFS を用いることにより、クラス定義とその継承関係、プロパティの継承関係を記述することができる。クラスとは型やカテゴリーに相当するもので、クラスに属するリソースをインスタンスという。

RDFS が提供する特徴的な機能として、プロパティに対する主語の定義域 (`rdfs:domain`) と目的語の値域 (`rdfs:range`) の定義がある。これにより、プロパティの主語や目的語のリソースが属するクラスを推論により求めることが可能となる。

RDFS が提供するクラスや継承の概念は Java 等のオブジェクト指向言語が提供する型システムと考え方が似ているが、大きく異なる点が多いので注意が必要である [14][15][16]。

### 2.2.4 OWL

RDFS では扱いきれない複雑な語彙を定義するためには本格的なオントロジー言語が必要となることから、OWL (Web Ontology Language) [17]が策定された。オントロジーという用語は様々な意味をもつが、OWL の仕様においては、OWL を、人間の知識のある部分に関する強力な汎用モデル化言語と位置づけて、このモデル化作業の結果をオントロジーと呼んでいる [18]。

OWL は、RDFS を拡張し、異なる RDFS クラス間の複雑な関係の表現と、特定のク

ラスとプロパティにより多くの正確な制約の表現を可能にしている。

また、OWL は記述論理 (DL : Description Logic) との対応がとられていることから[19]、論理計算の完全性 (すべての含意は計算可能) と決定可能性 (問題を解くためのアルゴリズムが存在してそれが必ず停止すること) が確保されている (但し、OWL Full を除く)。さらに、知識の一貫性を検証したり、暗黙知を明確にしたりするために、OWL で表現した知識を推論可能となっている[15][16]。

ここで、OWL の推論における意味論は開世界仮説 (OWA : Open World Assumption) に基づいていることに注意が必要である。データベースシステムの閉じた世界は閉世界仮説 (CWA : Closed World Assumption) に基づくため、真 (True) か偽 (False) が明示されていない事実は偽とみなされる。しかし Web の世界では情報が絶えず追加修正されることから、明示されていない事実が必ずしも偽とはいえないため、真か偽かが明示されていない事実は、現時点では不明 (Unknown) とみなすべきである。このため開世界仮説の意味論では不明 (Unknown) と偽 (False) が区別される。

また、OWL は単一名仮説 (UNA : Unique Name Assumption) を採用していない。単一名仮説とは、異なる個体名は別々の実体を表すことを仮定する。OWL ではこの単一名仮説を採用していないことから、2つの名前があるとき、それが同じ (ows:sameAs) か、異なる (owl:differentFrom) のかを明示する必要がある。

開世界仮説の採用と単一名仮説を採用していないことは、セマンティック Web の基本要件に適合している一方で、数え上げや否定などの扱いを難しくしている。

### 2.2.5 SWRL

ルールの記述方法を標準化したマークアップ言語として RuleML (Rule Markup Language) [20]がある。セマンティック Web での利用を目的として、RuleML のルール表現に OWL Lite/DL のクラスやプロパティを導入したのが SWRL (Semantic Web

Rule Language) [21]である。SWRL によるルール記述を用いることで OWL では表現できない知識を推論により求めることが可能である。

なお、OWL と対応関係にある記述論理は、その必要条件として推論が決定可能（問題を解くためのアルゴリズムが存在してそれが必ず停止すること）であることとされるが、SWRL は決定不能である[19]。

## 2.3 オントロジー

### 2.3.1 オントロジーの定義と種類

オントロジー (Ontology) とは本来哲学用語であり「存在に関する体系的な理論（存在論）」を意味する。人工知能研究の立場からは「概念化（対象世界において興味をもつ概念とそれらの間の関係）の明示的な規約」と定義される。オントロジーの基本的目標は、計算機に知識を記述・格納・利用する際に共有される基盤を提供することで、計算機による知識に基づいた知的処理を実現することである[7][22]。

オントロジーはさまざまな観点から分類できる。一つは上位オントロジー (Upper Ontology) とドメインオントロジー (Domain Ontology) による分類である。上位オントロジーとは世界を記述する際に一般に必要とされるような、物体、時間、空間、プロセスといった基盤的概念を定義したものである。代表例として、Sowa によるもの、DOLCE, BFO, YAMATO 等[23]がある。一方ドメインオントロジーは、ビジネスや法律等の、ある対象世界を表現するオントロジーであり、多くのドメインオントロジーが研究、開発されている[22]。

別の観点からはライトウェイト (light-weight) オントロジーとヘビーウェイト (heavy-weight) オントロジーに分類できる。ライトウェイトオントロジーは実用性を重視し、オントロジーを「厳密に定義された用語集合」と位置付けて、具体的な目標に対する必要な情報の効率的な記述とその有効利用を優先したものである。一方、へ

ビーウェイトオントロジーとは厳密な意味での一般－特殊関係や全体－部分関係などの多くの関係や公理によって概念の意味が記述され、哲学的考察に基づくものである。

### 2.3.2 構築方法

オントロジーの構築方法論あるいは構築ガイドラインとして、M.Uschold らの方法[24], TOVE 方法論[25], AFM (Activity-First Method) [26]などがある。また近年はソフトウェア開発におけるデザインパターンと同様にオントロジー構築におけるパターンやベストプラクティスについての研究も進んでいる[27][28][29]。

ここでは TOVE 方法論について概略を述べる[7][25]。TOVE 方法論は M.Gruninger と M.Fox によって開発されたオントロジー開発方法論で、TOVE (TOronto Virtual Enterprise) プロジェクトのオントロジー開発に利用された実績をもつ。TOVE 方法論のオントロジー開発手順を図 2-3 に示す。

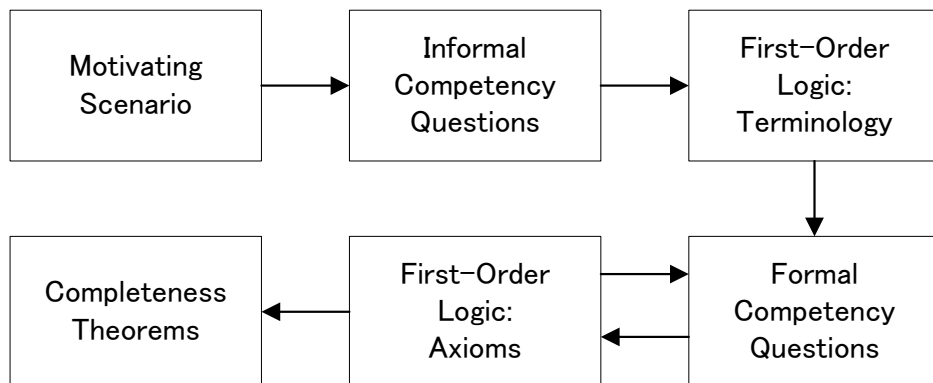


図 2-3 TOVE 方法論によるオントロジー開発手順

TOVE 法の特徴は、オントロジーの範囲を定義する基礎として能力質問 (Competency question) を導入したことである。基本的な流れは、オントロジーの要求仕様を定式化するために動機づけとなるシナリオを作成し、オントロジーに基づいて作られたモデルが回答するべきと思われる能力質問を作成する。ここまでは形式的な記

述ではないが、以降は能力質問から語彙を抽出し、一階述語論理等を用いて能力質問の公理記述を行うことで定式化する。最終的なオントロジーの完全性は能力質問によって明確になる。

### 2.3.3 オントロジー開発ツール

オントロジー開発を支援するツールがある[22]。ここでは広く利用されている Protégé と法造について述べる。

Protégé はスタンフォード大学が開発したオープンソースのオントロジーエディターである。Protégé は Java 言語で開発され、多くのプラグインによる拡張も提供されている。OWL2 によるオントロジー開発と SWRL によるルール記述が可能であり、プラグインで提供される記述論理の推論器 (Reasoner) を利用することにより、推論を行うことができる。

法造はオントロジーエディター、概念工房 (オントロジー構築ガイドシステム)、オントロジーサーバーから構成されるオントロジー構築・利用環境である[30]。法造のオントロジーエディターは「ロール」と「関係」に関する基礎的な考察をシステムに反映させたことを特徴としている。

### 2.3.4 法律分野のオントロジー

人工知能技術の法律分野への応用はこれまで多くの研究が存在し、法的知識の構造の解明や法律エキスパートシステムの構築も行われている[31]。ここでは法律概念の階層構造構築を支援する DODDLE を用いて、国連売買条約の法律知識を記述した法律オントロジーを構築している[32]。海外では法律知識を OWL で記述する試みとして ESTRELLA プロジェクトの LKIF (Legal Knowledge Interchange Format) がある

[33][29]. LKIF-Core は法律分野の知識モデリング支援とアプリケーション間の知識交換を目的とし、法律の基本概念を定義するための概念語彙定義と設計方法を示している。

これまでの研究開発成果の多くは、法律知識の構造解明が中心課題とされ、オントロジーの記述方法やその支援環境に関する成果が中心となっており、具体的な法律のオントロジー開発事例の報告は少ない[34].

## 2.4 XML 文書の妥当性検証

以下では代表的な構造化文書である XML とその妥当性検証について述べる。

### 2.4.1 XML

XML (Extensible Markup Language) [1]は, SGML (Standard Generalized Markup Language) を Web 上で送受信と処理できることを目的として, W3C により策定されたマークアップ言語作成のためのメタ言語である. マークアップ言語とは, 文書の一部を特殊な文字列 (タグ) で囲むことにより, 文書の修飾情報や論理構造などを記述するための言語である.

XML 仕様の概要を述べる. XML 文書は, 要素 (element) と属性 (attribute) により構成され, 要素は内部に子要素を含むことができる. 属性は要素に付随し, 属性の内部に子要素を含むことはできない. 要素は開始タグ「<要素名>」と終了タグ「</要素名>」で内容を挟んで記述する.

XML 仕様で定める文法に従った文書を整形形式 (well-formed) XML 文書という.

### 2.4.2 スキーマ言語による妥当性検証

XML の文書構造定義を行うためにスキーマ言語が定義されている. 整形形式 XML 文書で, かつスキーマ言語で定義される文書構造に従って記述された XML 文書を, 妥当

(valid) な XML 文書という。スキーマ言語を用いた妥当性検証とは、スキーマが定める文書構造に適合しているかどうかを検査することである。

以下に広く利用されているスキーマ言語の標準規格について述べる。まずスキーマ言語として XML の初期仕様から定義されている DTD (Document Type Definition) [1] がある。しかし DTD では要素構造等の記述はできるものの、名前空間やデータ型の表現ができない等の問題があった。そこで W3C は、それらの問題を克服した XML Schema[2][35][36]を 2001 年に策定した。XML Schema はオブジェクト指向設計の影響を多く受けており、オブジェクト指向言語のプログラム開発環境による支援も充実している。

XML Schema と同時期に、簡潔で高い表現力をもつ RELAX NG[37]が策定され、OASIS や ISO/IEC の標準規格となっている。

#### 2.4.3 その他の手法による妥当性検証

スキーマ言語では記述できない制約条件がいくつかある。アプリケーションがスキーマ言語では記述できない制約を検証するためには、DOM などの XML 文書構造木を直接操作するプログラミングが必要になり、「アプリケーションプログラム中への検証ロジック埋め込みによる開発効率低下」、および「文書規約変更時のコードの修正・コンパイル作業による保守性低下」が生じるという問題がある。

これに対し、自然言語処理分野で用いられる素性論理に基づいた新たな文書ルール記述言語 (DRDL; Document Rules Description Language) と DRDL に基づく XML アプリケーション開発フレームワーク (DRDL フレームワーク) の提案がある[3]。

一方、SWRL を用いて XML 文書の検証を行う研究がある[4]。この手法は、XML 文書そのものを検証するのではなく、XML 文書を OWL へ変換して SWRL で記述したエラー検出ルールの推論を実行することで、記述内容のエラーを検出するものである。し

かし, XML 文書全体の妥当性の判定方法については言及していない.

## 2.5 まとめ

本章では本論文に関連する技術および関連研究について述べた. まず, セマンティック Web の概要を述べ, その中から本論文に関係が深い技術について述べた. 次にオントロジーの定義を述べ, オントロジーの構築方法と法律分野の既存研究について述べた. 最後に, 代表的な構造化文書である XML とその妥当性検証に関する既存研究について述べた.



## 第3章      オントロジーを利用した妥当性検証システム

本章ではこれまでに述べた背景知識や関連研究を踏まえ、オントロジーを利用した構造化文書の妥当性検証システムを提案する[8].

### 3.1    概要

インターネットの普及にともなって電子申請 EDI が増加している. ここでやり取りされるデータは XML に代表される構造化文書で仕様定義されることが多い. 電子申請や EDI を行うシステムでは交換されるデータに対する文書構造のチェックを行う機能を有していることから, 送信前に文書形式に関する問題は検出可能となっている. しかしながら, 文書形式上の問題はなく送信が成功したとしても, その記載内容に整合性が欠けている等の文書内容の妥当性に問題がある場合は, データ送信後に損害や不利益を生じる場合がある. 例えば高額取引において意図と異なる契約が成立してしまった場合, 取引が不成立となってしまった場合, あるいは権利に関する申請が却下となった場合等では大きな損害や不利益を生じる可能性がある. このことから, 申請や取引の文書内容が妥当であることを送信前に検証することが重要である. 文書内容の妥当性判断が深い知識を必要とせず機械的に検証可能なものについては, 従来から研究されている文書ルール記述言語等を導入して検証することで取引等の安心, 安全も確保できる. しかしながら, 特に文書内容の妥当性の判断に専門的知識を必要とする分野においては, 深い知識のルール化が難しいこともあり, 文書内容の妥当性検証に関するコンピュータ支援が進んでいるとはいえない.

専門的知識を必要とする分野における文書内容の妥当性検証を行うシステムを構築

するためには以下の2つの課題を解決する必要がある。

一つ目の課題は、専門的知識を必要とする複雑な妥当性検証ルールを記述することである。一般に専門家が判断を行う分野に必要とされる知識は広く深い。専門的知識の概念整理を行うことなく検証ルールを記述しようとしてもルール数と複雑さが増大し、一貫性のあるルールセットを作成することは難しい。仮に作成できたとしてもメンテナンスが困難である。そこで専門的知識とルールを体系的かつ柔軟に記述できる手法が必要とされる。

二つ目の課題は、特定分野の専門家とIT技術者の共同作業を可能とすることである。IT技術者は専門的知識の概念整理を行うことは難しいため、この作業は対象分野の専門家が行うこととなる。しかし専門家は情報処理知識をもっていることは少ないため専門家自身がルールを記述することは難しい。そこで専門家とIT技術者がお互いの意図を正しく伝達でき、相互理解可能な共同作業のための基盤が必要となる。

ところで近年、次世代Web技術としてセマンティックWebが注目を集めている。セマンティックWebは、Web上のリソースについてのメタデータを機械可読可能な情報として記述し、さらにメタデータの意味や相互関係を定義するオントロジーやルールを与えて高度な検索や推論等を行えるようにする技術である。W3C (World Wide Web Consortium) によりオントロジー記述言語やルール記述言語等をはじめとしたセマンティックWeb関連技術が仕様化され、活発に研究、開発され、利用されている。

そこで本論文ではセマンティックWebがWebに対して行うこと、つまりオントロジーを用いてWeb上の情報の意味を考慮した検索や推論を行うことと同様に、セマンティックWeb技術を利用して電子申請やEDIが対象とする分野のオントロジーを構築して推論を行うことにより、専門的知識を必要とする分野における構造化文書の妥当性検証を行うシステムを提案する。

以降、本章の構成は以下の通りである。3.2節で構造化文書の妥当性検証システム

について概要について述べた後、その構成要素と動作について詳細に説明する。3.3節で妥当性検証システムの核となるオントロジーの構築方法について述べる。3.4節でJava言語を用いた妥当性検証システムの実装構成と既存ツールを利用したオントロジーコンバーターの実装について述べる。3.5節で妥当性検証システムの評価について述べる。最後に3.6節で本章のまとめと今後の課題について述べる。

#### オントロジーを利用した妥当性検証システム

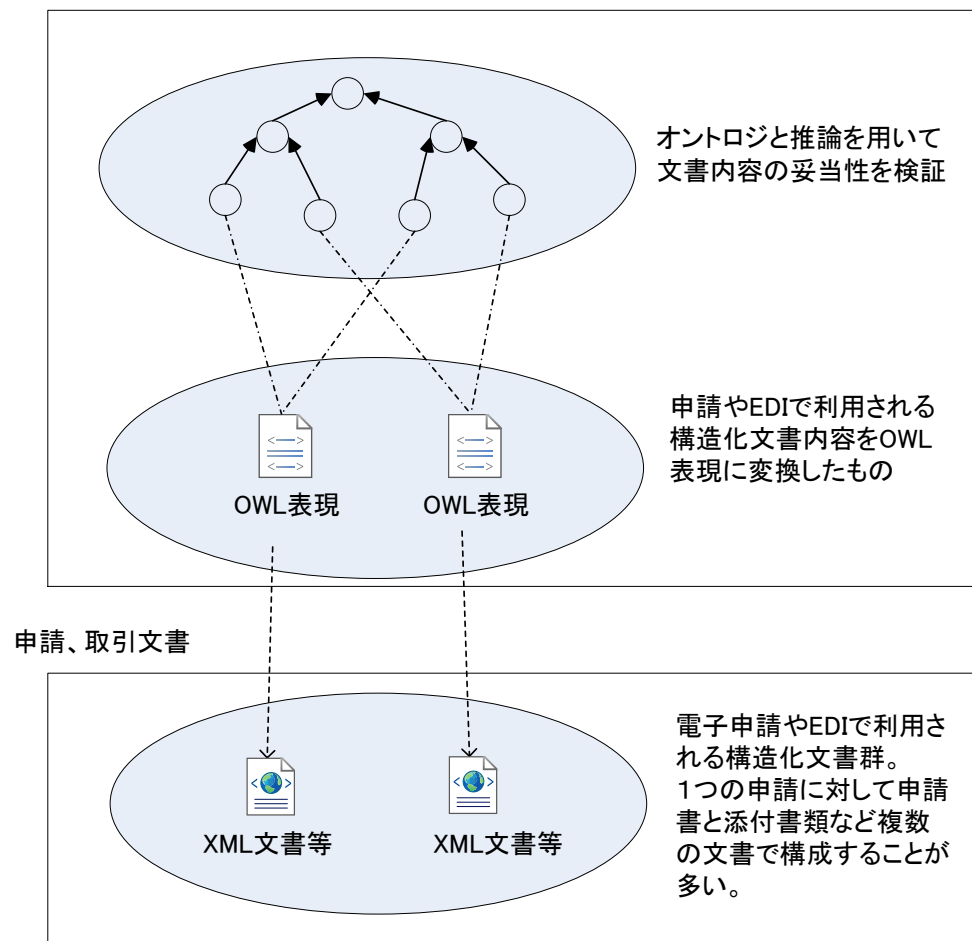


図 3-1 妥当性検証システムの考え方

## 3.2 妥当性検証システム

### 3.2.1 概要

提案する妥当性検証システムは、検証対象の構造化文書を読み込んでオントロジーを用いた妥当性検証処理を行った後、文書が妥当かどうかの判断結果を表示するとともに判断の根拠となった推論過程の情報を含んだ結果オントロジーの出力を行うものである[8].

本システムの構成概要を図 3-2 に示す.

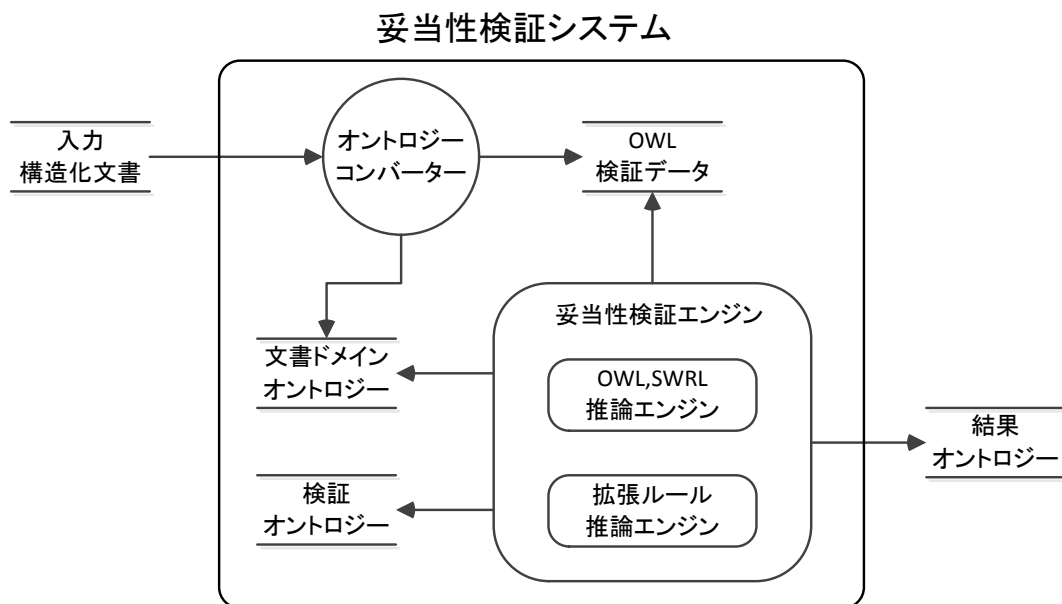


図 3-2 妥当性検証システムの構成

OWL を用いてオントロジーを作成する利点の 1 つとして推論が定義されていることがある。これは、クラスやプロパティを形式化し、個体と個体間のプロパティを与えることで、クラスや個体に関する推論が可能ということである[15].

本システムの特徴は、この性質を利用し、検証対象の構造化文書内容を個体と個体間のプロパティで記述する OWL 表現に変換して推論を行い、検証対象の個体が妥当であることを表すクラスのインスタンスとなるかどうかを判定することにより妥当性検証

を行うことである。このような推論による検証を行うためには、与えられた検証データ（個体と個体間のプロパティ）に対してクラス等を推論により導出できるオントロジーを構築する必要がある。

本システムではこのような推論による検証を行うためのオントロジーを以下の2つに分割して構築する。1つは入力となる構造化文書の内容を OWL で表現するための語彙を定義する文書ドメインオントロジーであり、OWL で記述する。もう1つは文書内容に対する検証項目が妥当であることを表すクラスとそれを導出する推論規則にあたる検証ルールを定義する検証オントロジーであり、OWL, SWRL, 拡張ルールを用いて記述する。ここで拡張ルールとは OWL, SWRL では記述できないルールを Java 言語等のプログラミング言語を用いて実装するものである。

本システムの妥当性検証処理はこれらのオントロジーを用いて以下のように行われる。構造化文書が与えられたとき、まずオントロジーコンバーターが構造化文書の記述内容を文書ドメインオントロジーで定義された語彙を用いて OWL 表現へ変換し、検証データとして出力する。続いて妥当性検証エンジンが検証データに対して文書ドメインオントロジーと検証オントロジー、拡張ルールに基づく推論を実行する。推論が停止すると、文書内容の妥当性検証結果を表示するとともに、推論途中に導出される妥当と判断した根拠情報を含む結果オントロジーを出力して終了する。

ここで構築した文書ドメインオントロジーと検証オントロジーの範囲において、文書内容の妥当性検証処理は、文書全体を表す個体が、推論停止時点において OKDocument（検証オントロジーで定義された文書全体が妥当であることを表すクラス）のインスタンスとなる場合にのみ妥当な構造化文書であると判定して“妥当である”と表示する。OKDocument のインスタンスとならない場合は“妥当とは言えない”と表示する。妥当とは言えない理由は、必要とされる記載項目の欠如や記載内容に齟齬があること等が原因と考えられる。このとき、出力された結果オントロジーを参照することにより問題

---

を特定するためのヒント情報を得ることができる。

以下では、提案システムの詳細を述べる。

### 3.2.2 文書ドメインオントロジー

文書ドメインオントロジーは、検証対象とする構造化文書の記述内容（概念と要素間の関係）とその検証に必要な関連知識を表す語彙を OWL のクラス、プロパティ、その他のコンストラクターを用いて記述するドメインオントロジーである。

本システムでは妥当性検証のためのオントロジーを、文書ドメインオントロジーと検証オントロジーに分けて構築することから、検証項目と検証ルールは文書ドメインオントロジーには記述せず、検証ルールの記述に必要な語彙を記述することを基本とする。但しクラスやプロパティの制約と検証ルールの区別は難しい場合がある。例えば業務要件（制約）をクラスやプロパティの制約として記述すると、その制約が推論により検証ルールとして機能する場合がある。このような制約を文書ドメインオントロジーの制約として記述すべきか検証オントロジーに検証ルールとして記述すべきか判断に迷う場合もありうるが、指針として制約記述が検証ルールとは独立して汎用的に利用でき、OWL で十分な記述ができる内容については文書オントロジーに含めることとする。

### 3.2.3 検証データ

検証データは、構造化文書の内容を文書ドメインオントロジーで定義された OWL のクラス、プロパティその他の OWL のコンストラクターを用いて表現したものである。このため検証データの内容は、クラスやプロパティ等の新たな語彙定義は含まず、構造化文書の要素や文書内容の抽象的な要素を表す個体とそれらの個体間の関係を定義するオブジェクトプロパティ、個体に関するデータプロパティで構成される。

構造化文書から検証データへの変換はオントロジーコンバーターにより行う。

#### 3.2.4 オントロジーコンバーター

オントロジーコンバーターは、構造化文書の内容を文書ドメインオントロジーで定義された語彙を用いて OWL 表現へ変換し、検証データとして出力するコンポーネントである。

本システムの妥当性検証処理は、OWL の個体と個体間のプロパティから推論により関連するクラスを導出するアプローチをとる。従ってオントロジーコンバーターによる変換結果は、構造化文書の文書要素とその関係を OWL の個体と個体間のプロパティの宣言へ変換したものとなる。

この変換処理は2つの役割をもつ。

1つはファイル形式等の文法あるいは構造の変換処理である。OWL は抽象構造であり、アプリケーション間でオントロジーの交換を行うには OWL 仕様が定めるシリアル化形式 (RDF/XML, OWL/XML, Turtle 等) を利用する必要がある。よって入力となる構造化文書の形式から OWL のシリアル化形式へ形式変換を行う必要がある。例えば入力となる構造化文書が政府の定める電子申請用 XML ファイル形式である場合、同じ XML 形式でも OWL 仕様の RDF/XML や OWL/XML とは構造が異なるため、OWL 仕様が定めるシリアル化形式へ変換が必要となる。

もう一つはオントロジーマッピングに関する処理であり、入力文書の記載事項が意味する内容を文書ドメインオントロジーの語彙表現へ変換することである。もし入力元文書が OWL 仕様のシリアル化形式で与えられた場合、ファイル形式の変換は不要である。しかしながら、その入力オントロジーが文書ドメインオントロジーとは異なる場合には、入力オントロジーから文書ドメインオントロジーへの語彙変換が必要となる。このようなオントロジーマッピング処理には、入力オントロジーのプロパティを文書ドメ

インオントロジーのプロパティの語彙へ単純に置き換えればよいレベルのものから、複数の要素の統合や分割などの構造変更が必要なレベルまで様々である。

### 3.2.5 検証オントロジー

検証オントロジーは、検証項目が妥当であることを示すクラスと、それを導出する検証ルールを文書ドメインオントロジーの語彙を用いて OWL, SWRL と拡張ルールで記述したものであり、検証システムの推論に利用される。

まず、与えられた個体が、ある検証項目に対して妥当かどうかの判定を以下のように定義する。

“検証項目 Y が妥当であることを表すクラスを OKY とすると、与えられた個体 X が検証項目 Y に対して妥当と判定するのは、X が OKY のインスタンスであるときのみである。”

このとき、個体 X は検証項目 Y に対して妥当であるという。

また、個体 X が検証項目 Y に対して妥当であることを検証するルールを以下のように定義する。

IF X が検証項目(Y)の必要条件を満たす

THEN X は OKY のインスタンスである (X は Y に対して妥当である)

ここで直接“X は OKY のインスタンスである”を帰結してもよいし、直接帰結せず、関連する個体間のオブジェクトプロパティを帰結し、その推論により間接的に“X は OKY のインスタンスである”を導出してもよい。

なお、本システムでは、検証項目が妥当であることを表すクラス名を”OK 検証項目”のように、検証項目名の前に OK を付けて命名する。

以上の考え方に従い、与えられた構造化文書の妥当性の検証は以下のように行う。まず、構造化文書全体の内容が妥当であることを表す“OKDocument”クラスを定義す



る。そして、推論を行った結果、構造化文書全体の内容を表す個体（一般的は XML 文書のルート要素がこれに該当する）が **OKDocument** クラスのインスタンスとなれば妥当と判定し、インスタンスとならない場合は妥当とはいえないと判定する。

推論により **OKDocument** クラスを帰結するルールは、与えられた構造化文書の各記載項目に対する検証項目が妥当であることを表すクラスの演算により条件を表現し、OWL の等価公理を用いて定義できる。

例えば、与えられた文書が満たすべき検証項目が、**OKItem1** かつ、(**OKItem2** または **OKItem3**)、である場合、**OKDocument** は以下のように定義できる。（以下は Manchester Syntax[38]により表記する）

Class: **OKDocument** EquivalentTo:

**OKItem1** and ( **OKItem2** or **OKItem3** )

一方、上記 **Item1**～**3** のような **OKDocument** の前提条件となる各検証項目についても、その検証項目が妥当であることを導出するルールを記載する必要がある。各種検証項目の妥当性検証ルールは、文書全体を表す個体ではなく、文書中の各記載項目を表す個体に対する検証条件を記述することが中心となる。この各記載項目に対する検証項目が妥当であることを示すクラスを導出する検証ルールは、以下の順に適用可能な状況を検討して記述する。

1. プロパティを帰結し、間接的にクラスを帰結するルール
2. クラスを直接帰結するルール
3. 拡張ルールによる検証処理（OWL, SWRL では記述できない検証ルール）

なお、OWL、SWRLの双方で記述可能なルールは、OWLで記述する。OWLで記述することにより、記述論理による決定可能性を確保でき、さらに記述がクラスやプロパティに対して行われるため可読性や保守性がよいという利点が得られる。

以下にこれらについて述べる。

#### (1) プロパティを帰結し、間接的にクラスを帰結するルール

詳細な検証項目の多くは、文書の記載項目間に齟齬がないかを確認するものである。特に電子申請のように申請書と事実情報を表す添付情報で構成されるような文書においては、申請内容の記載事項と添付情報の記載事項に齟齬が無いかを確認することが基本となる。このような検証ルールは、記載内容の個体と関連情報の個体、関連するクラスやプロパティ等のパターンマッチングとして考えることができるので、その多くをSWRLで記述可能である。

SWRLの帰結部には、検証項目が妥当であることを表すクラスを直接帰結するのではなく、記載情報の個体と検証結果が妥当である根拠となる個体を関連付けるオブジェクトプロパティの宣言を帰結する。帰結するオブジェクトプロパティの定義域もしくは値域に検証項目が妥当であることを表すクラスを定義しておくことで、推論により妥当であるクラスを導出する。これは本システムの個体と個体間のプロパティから推論によりクラス等を導出するアプローチに沿ったものである。これによりさらなる推論に利用できるだけでなく、記載事項が妥当である根拠を示す情報へのリンクを明示できる利点がある。

#### (2) クラスを直接帰結するルール

検証項目の中には、記載内容自体の要件に対する検証項目がある。これはOWLの等価公理等を用いて記述できるものとSWRLで記述できる場合があるが、OWLで記述

できる場合は OWL で記述する.

### (3) 拡張ルールによる検証処理

OWL と SWRL は, 開世界仮説 (OWA : Open World Assumption) の採用と, 単一名仮説 (UNA : Unique Name Assumption) を採用していないことにより, 全ての個体が満たす条件の推論や否定に関する推論, 数え上げ等が困難か記述不可能である. 一般に, このような推論を行えるようにするためには, 世界を閉じるための特別な構造表現や個体に関する追加の宣言等が必要となるので, モデルが複雑化して現実的ではない.

また, OWL と SWRL は汎用プログラミング言語ではないため, 複雑な計算処理等, 記述できない処理も多い.

そこで, OWL と SWRL の推論モデルの制限等により記述できない検証ルールを, Java 言語と OWLAPI を用いて拡張ルールとして実装する. 拡張ルールにおいても推論 (Java 言語によるルールの適用) を行い, 最終的に検証オントロジーで定義する検証項目が妥当であることを表すクラスやプロパティを導出する. 例えば, オブジェクトプロパティを帰結する場合, プロパティからクラスを導出する規則は, 検証オントロジーの OWL や SWRL を用いて定義する. 但し, 拡張ルールを適用する範囲は最小限にとどめる.

拡張ルールによる検証処理には主に以下の特徴がある.

- 拡張ルールの適用 (推論) は, OWL, SWRL の推論後に行う. これにより, OWL, SWRL の推論により導出された内容を元に拡張ルールを実行できる.
- 拡張ルールの実装は, 開世界仮説を前提とせず, 与えられた個体のみで閉じられた世界として処理を行うことで, 全ての個体が満たす条件や, 個体の個数等の判定を行えるものとする.
- Java 言語による実装により, 他のデータベース等との情報連携が可能である.

### 3.2.6 妥当性検証システムの動作

検証システムは以下のように動作する.

- ① 前処理として, 入力となる構造化文書をオントロジーコンバーターにより検証データ (OWL 形式) へ変換する.
- ② 文書ドメインオントロジー, 検証オントロジー, 検証データを読み込み, この状態のオントロジーを  $O$  とする.
- ③  $O$  に対して OWL と SWRL の推論を実行し, その結果のオントロジーを  $OR$  とする.
- ④  $OR$  に対して全ての拡張ルールを適用し, その結果のオントロジーを  $OE$  とする.
- ⑤  $OR \neq OE$  のとき, 新たな知識の追加があったので,  $O=OE$  として, ③に戻る.
- ⑥  $OR == OE$  のとき, 新たな知識追加がないので,  $OE$  を結果オントロジーとして保存して終了. このとき, 文書全体を表す個体が OKDocument のインスタンスであれば, “妥当な文書” と判定して表示する. そうでない場合は “妥当とはいえない” と表示する.

なお, 処理後出力される結果オントロジーは, 推論の過程で導出されたクラスやプロパティを含んでいる. これを Protégé 等のオントロジーエディターで読み込むことで, 妥当である根拠を示すリンクや, 妥当でない場合に妥当となっていない項目が何かを確認することが可能である.

### 3.3 オントロジーの構築

妥当性検証システムで利用する文書ドメインオントロジーと検証オントロジーの二

つのオントロジーの構築方法について述べる。

オントロジーの構築方法論は様々な提案がなされているが、一般に広く深い業務知識を完全に表現するオントロジーを新しく開発することは容易ではない[7]。しかし本システムで利用するオントロジーの目的は妥当性検証のための推論であることから、妥当性検証に必要な知識という観点で作成するオントロジーの対象知識を絞ることが可能である。また具体的な対象文書の妥当性検証を行うことで、オントロジーの記述内容に対するテストも容易に行うことができる。このことから、本システムで利用するオントロジーの構築は、まず妥当性検証ルールを検討し、そのルール記述に必要となるクラスやプロパティを抽出してオントロジーを記述してテストとレビューを行う。この作業を繰り返すことで、順次オントロジーの完成度を高めながら構築していくことが可能となる。

以下に妥当性検証システムのためのオントロジー構築手順を述べる。なお、オントロジーの構築は情報処理技術者と対象分野の専門家が共同して作業することを想定している。

- ① 情報処理技術者が構造化文書の仕様書等からたたき台となる初版オントロジーを作成する。

電子申請や EDI では一般に構造化文書に関する仕様書が存在する。ここに記載されたタグ定義やタグの構造からクラス候補とプロパティ候補を抽出して初版オントロジーを作成する。電子申請や EDI システムのマニュアルがある場合、ここには業務説明や運用上の注意点等、システムの仕様書には記載されていない有用な情報が記載されている可能性がある。これを利用することでクラスやプロパティの意味理解の助けにすることや検証項目の抽出ができる。これらの情報も初版オントロジーに反映する。

この段階は情報処理技術者が作業することにより、対象分野の専門家との知識共有のたたき台を作成するだけでなく、情報処理技術者の基礎知識習得、疑問点の抽出を行うことができる。但し一般に情報処理技術者は検証に必要な深い知識までは持ち合わせていないので、この段階では文書ドメインオントロジーの作成が中心となる。

また、この段階でのオントロジー記述は形式的なものである必要はないが、オントロジーエディターを用いて記述できるものは記述する。早い段階で実装を考慮することにより、早期に不明点や記述が困難な点を予見できる可能性がある。オントロジーエディターを用いる場合、アノテーションを積極的に利用し、定義や疑問点等のコメントを記述する。以降の作業においてもアノテーションの内容を更新していくことが重要である。

② 専門家と初版オントロジーのレビューを行う。

初版オントロジーで定義したクラスやプロパティを情報処理技術者と専門家が共同でレビューする。一般に専門家はオントロジー技術についての知識がないため、この段階でオントロジー技術の考え方とあわせて初版オントロジーの内容を説明することで、今後の共同作業の土台を作ることが主な目的となる。専門家は用語の意味や関係については詳しいことから、抽出した概念とその関係について議論を行い、その結果を初版オントロジーに反映させる。

③ 具体的な検証項目とそのルールを検討してテストケースを作成する。

検証オントロジーに組み込まれていない具体的な構造化文書に対する検証項目とその検証手順を洗い出して検討する。検証手順を考えることで、検証ルールを記述するために必要な概念とその関係が文書ドメインオントロジーに記述されているかを検証する。この段階で、仕様書やマニュアルには記

載されていない深い知識が抽出される可能性がある。このような知識には、システムの仕様書には記載が無い業界知識によるもの、法令等の要件、実務上の慣習等が考えられる。例えば仕様書に金額の記載項目があるとする。一般に電子申請や EDI における仕様書には数値型の指定等の要件しか記載されていないことが多いが、実務上は金額の正当性が重要であることがある。例えば、他の記載項目により税率が変化して金額が決定する場合は、金額の妥当性を検証できるルールを記述するために、文書ドメインオントロジーに法令等を参照して知識を追加する必要がある。

この段階は専門家と情報処理技術者によるブレインストーミングのように開始し、最終的には今回対象とする具体的な検証項目を少数に絞る。そして検証項目に対する検証ルール（手続）と追加あるいは更新が必要となる語彙を抽出する。この結果、文書ドメインオントロジーと検証オントロジーに反映する内容を決定する。さらに、そのテストケースを作成する。ここでは、検討がプログラム実装に偏りすぎて専門家との議論がかみ合わなくなることを避けるため、ルールや語彙を形式的記述ではなく自然言語で記述する。

④ ③の内容をコーディングする。

情報処理技術者は、③で検討した検証項目に対するルールと必要な語彙を、文書ドメインオントロジーと検証オントロジーに追加する。文書ドメインオントロジーは OWL で記述し、検証オントロジーは OWL, SWRL, 拡張ルール（Java 言語）を用いて記述する。この段階で実装が難しい点や以前のオントロジーと不整合があるなどの問題が発見された場合は、次回の専門家とのレビュー時に検討する。

⑤ テストケースを用いて妥当性検証システムのテストを行う。

情報処理技術者は、③で検討したテストケースに対する具体的なデータを用

意し、④でコーディングしたオントロジーを妥当性検証システムに組み込んでテストを行う。また、テストツール等を利用して過去に作成したテストケースと今回追加したテストと併せて回帰テストを行うことにより、知識やルールの追加、修正による問題を検出する。

⑥ ③に対するレビューを共同して行う。

③で検討した検証項目に対する妥当性検証システムの動作を専門家と情報処理技術者が共同でレビューする。コーディング時やテストで問題があった場合は、検証ルールの定義に問題が無いか専門家と検討して対処方法を考える。

文書ドメインオントロジーと検証オントロジーの記述内容が増えて複雑化した場合は、クラスやプロパティの継承関係やルール記述内容の見直しを行い、可能であればオントロジーを洗練する。但しクラスやプロパティの構成を変更する場合は、変更した結果が業務的な意味と異なることが無いか専門家と共に検討する必要がある。

⑦ ③から⑥の手順を繰り返す。

妥当性検証に必要な検証項目を網羅したオントロジーが作成できるまで③から⑥の手順を繰り返す。

上記手順のように妥当性検証を切り口とすることで、無用にオントロジーの作成対象範囲を広げることなく文書ドメインオントロジーと検証オントロジーを作成することが可能である。また、順次検証可能範囲を広げていく手順であることから、適用可能な範囲から妥当性検証システムを運用することが可能となる利点もある。

ところで上記手順はオントロジーを新規に作成する手順について述べたものであるが、近年は様々な分野のオントロジーが開発されていることから、対象分野のオントロ



ジーが既に作成されている可能性もある。既に対象分野のオントロジーが OWL で構築済みであれば、そのオントロジーを文書ドメインオントロジーとして利用できる可能性がある。OWL 以外で構築されたオントロジーがあれば、OWL で再構成することで文書ドメインオントロジーとして利用できる可能性がある。もし既存のオントロジーが存在するが対象分野の一部しか表現していない場合でも、そのオントロジーに不足する知識を追加することで文書ドメインオントロジーとして利用できる可能性もある。OWL には他のオントロジーをインポートするメカニズムが用意されており、オントロジーの再利用を比較的容易に行うことが可能である。

このように文書ドメインオントロジーは対象分野の知識の記述であることから、既存のオントロジーが再利用できる可能性が高く、新規にオントロジーを作成する前に既存のオントロジーの存在を調査することは有用である。但し一般にオントロジーの再利用には注意が必要とされている[24]。特に既存オントロジーの中に、妥当性検証目的として再利用が難しい類似概念を含んでいる場合には、検証目的に別の概念を定義するか、もしくは再利用をやめて新規に作成するべきかを状況によって判断する必要がある。この判断にも検証項目とルールを検討することで一定の判断材料とすることが可能である。既存オントロジーに、検証ルールを記述するために必要となる知識が記述されているか、意図と異なる類似概念が含まれているかを検討する。そして、修正作業や利用における制限事項が多いようであれば、妥当性検証用に特化したライトウェイトなドメインオントロジーとして新規に作成したほうがよいと思われる。但し、何が問題で既存オントロジーを再利用できないかをよく検討し、その理由を明確にしておくことが重要である。

### 3.4 システムの実装

#### 3.4.1 システムのプログラム構造

本システムは Java 言語を用いて実装した。OWL と SWRL の解釈には OWLAPI を用いた。妥当性検証エンジンは OWL と SWRL の推論エンジンと Java 言語で実装された拡張ルール推論エンジンで構成される。OWL と SWRL の推論エンジンには OWLAPI 経由で Pellet を用いた。またオントロジーエディターである Protégé を用いて各オントロジーファイルの表示や編集を行った。

図 3-3 に妥当性検証システムの実装に関する主なコンポーネント構成を UML 図で示す。なお図には明示していないが、オントロジーの操作等は OWLAPI の API モデルを利用している。また詳細なメソッドやシグニチャは省略している。

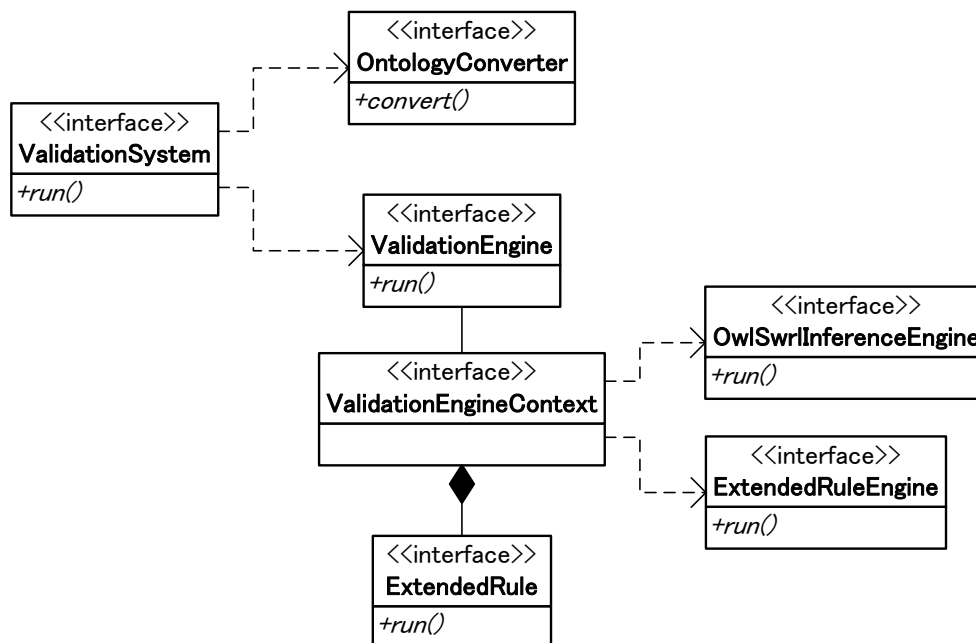


図 3-3 妥当性検証システムのコンポーネント概要

これらのコンポーネントは Spring Framework 等の DI (Dependency Injection : 依存性の注入) コンテナを用いることで実装を取り換えることを可能とする。例えば対象

分野毎にオントロジーコンバーター，拡張ルール，推論エンジン等を置き換えることが可能であり，適用分野毎にカスタマイズした検証システムを構成できる．

以下にそれぞれのコンポーネントの役割について述べる．

### ***ValidationSystem***

妥当性検証システム全体の動作制御を行うオブジェクトを定義するインターフェイス．具象オブジェクト生成時に `OntologyConverter` と `ValidationEngine` のインスタンスを設定すると同時に文書ドメインオントロジーと検証オントロジーを準備する．`run` メソッドが呼ばれると，まず指定された構造化文書ファイルを引数として `OntologyConverter` の `convert` メソッドを実行し，検証データを作成する．続いて検証データ，文書ドメインオントロジー，検証オントロジーを引数として `ValidationEngine` の `run` メソッドを実行し，判定結果と結果オントロジーを得る．そして判定結果を表示するとともに結果オントロジーを保存して終了する．

### ***OntologyConverter***

与えられた構造化文書を検証データに変換するオブジェクトを定義するインターフェイス．本インターフェイスを実装するオブジェクトであれば変換処理の実装方法に制限はない．よって，本インターフェイスを直接実装する Java ライブラリで構成してもよいし，外部プログラムを呼び出すラッパーオブジェクトで構成しても良い．

### ***ValidationEngine***

文書ドメインオントロジーと検証オントロジーをもとに検証データに対する推論を実行し，検証データの妥当性の判断を行い，検証結果と結果オントロジーを出力する検証エンジンオブジェクトを定義するインターフェイス．具象オブジェクトの生成時に `OwlSwrlInferenceEngine` と `ExtendedRuleEngine` の二つの推論エンジンと拡張ルー

ルオブジェクトリストのインスタンスを設定する。具象オブジェクトは渡されたオントロジーと検証データ、さらに推論に利用するオントロジーモデルと検証状態の管理も行う。具象オブジェクトが `ValidationEngineContext` インターフェイスを実装することで、推論エンジンから `ValidationEngine` がもつオントロジー等のリソースや状態へアクセスを可能とする。

`ValidationEngine` は妥当性検証システムの推論動作を実装し、オントロジーとその状態を管理するが、推論処理は推論エンジン呼び出すことにより行う。

### ***ValidationEngineContext***

推論エンジンが推論に必要となるオントロジーやその状態等のリソースへアクセスするためのインターフェイスを定義する。一般に `ValidationEngine` インターフェイスを実装する具象クラスが本インターフェイスを実装する。妥当性検証エンジンの外部仕様として `ValidationEngine` を定義し、推論エンジンとの関係を内部仕様として `ValidationEngineContext` を定義して、外部向けと内部向けのインターフェイスに分離することにより、各コンポーネントの依存度を下げ、容易にカスタマイズできるようにすることを目的としている。

### ***Ow/Swr/InferenceEngine***

OWL と SWRL の推論を実行するオブジェクトを定義するインターフェイス。本システムはオントロジー等のモデル操作に OWLAPI を利用していることから、OWLAPI に対応している推論エンジンから選択することになる。OWLAPI に対応した推論エンジンとして FaCT++, HermiT, Pellet 等がある。論文執筆時点において、SWRL に関する推論のサポートが最も充実していることから、本システムは主に Pellet を利用している。

### ***ExtendedRuleEngine***

拡張ルールの推論を実行するオブジェクトを定義するインターフェイス。run メソッドが呼び出されると、引数として渡される ValidationEngineContext からオントロジーのモデルと拡張ルールを実装する ExtendedRule のリストを取得する。オントロジーを引数として、全ての ExtendedRule の run メソッドをリストの順に呼び出すことで個々の拡張ルール推論を実行する。

### ***ExtendedRule***

1 つの拡張ルールを実装するオブジェクトを定義するインターフェイス。ValidationEngineContext から得られるオントロジーに対して、拡張ルールが定義する推論の結果として新しい知識が導出された場合は、その知識を ValidationEngineContext のオントロジーへ追加する。

## 3.4.2 オントロジーコンバーターの実装

電子申請や EDI で利用される構造化文書のほとんどは XML 形式で定義されていることから XML 文書に対するオントロジーコンバーターの実装方法について述べる。

XML は広く普及した技術であり、多くのツールやライブラリが開発されている。対象分野の XML 形式の特性を考慮した上で既存のツールやライブラリを利用することにより開発工数の削減や保守性の向上が期待できる。そこで既存ツールを利用したオントロジーコンバーターの実装について述べる。

XML から OWL への変換には多くの研究が有り、その多くがツールとして公開されている。変換手法も多様であり、目的に応じて選択することができる。

以下では JXML2OWL Project[39]により公開されている JXML2OWL[40]を使用したオントロジーコンバーターの作成例を示す。まず JXML2OWL のプログラム構成と

処理の流れを図 3-4 に示す。

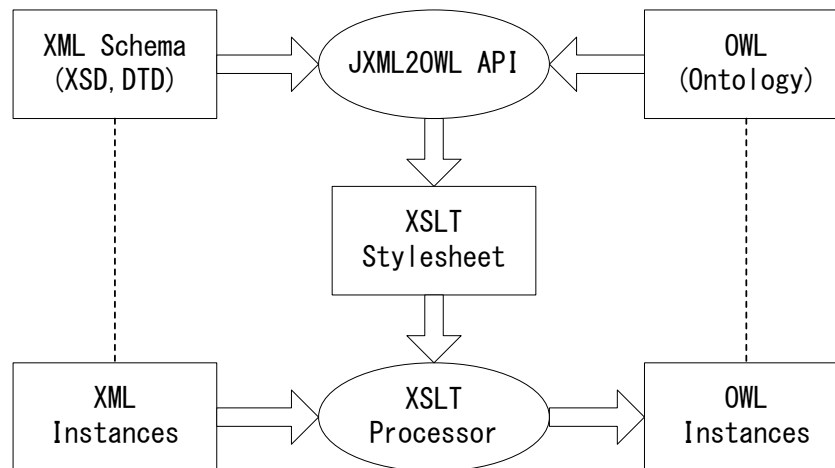


図 3-4 JXML2OWL の概要

JXML2OWL は XML スキーマからオントロジーを自動作成するツールではなく、既存の XML Schema と OWL で記述したオントロジーを用意し、利用者がそれらの XML Schema と OWL との変換規則を JXML2OWL に与えることによって、XML Schema に基づく XML ファイルから OWL のインスタンスファイルを出力する XSLT スタイルシートを生成するツールである。一度 XSLT スタイルシートを作成すれば同じ XML スキーマに基づく XML ファイルは、変換規則に従って OWL のインスタンスファイルへ変換可能となる。なお JXML2OWL は、XML Schema ではなく XML インスタンスファイルを与えた場合でも内部的スキーマ要素を作成して変換規則を反映した XSLT スタイルシートを作成できるので、XML Schema は必須ではない。

JXML2OWL が採用する変換規則の特徴として、XPath を用いて XML 要素と OWL クラスの対応関係を柔軟に指定できること、1 対 1、1 対多、多対 1 の変換ができること、重複しない OWL のインスタンス ID を出力できること等がある。

JXML2OWL Project では、変換規則の作成からインスタンス文書の作成までの全工

程をグラフィカルユーザーインターフェイスから操作できるツール JXML2OWL Mapper も提供している。図 3-5 に変換定義作成時の画面を示す。左側に XML Schema の要素が階層的に表示され、右側に OWL のクラスが表示される。利用者は XML 要素を選択して対応する OWL クラスへドラッグすることで XPath による変換規則が自動作成されて画面下の表に表示される。さらに OWL クラスの変換定義にオブジェクトプロパティとデータプロパティの変換規則も追加作成可能である。(但し OWL のクラス定義と共に、オブジェクトプロパティのドメインとレンジに変換対象のクラスを定義しておく必要がある。またデータプロパティのドメインに変換対象のクラスとレンジにデータタイプを定義しておく必要もある。)

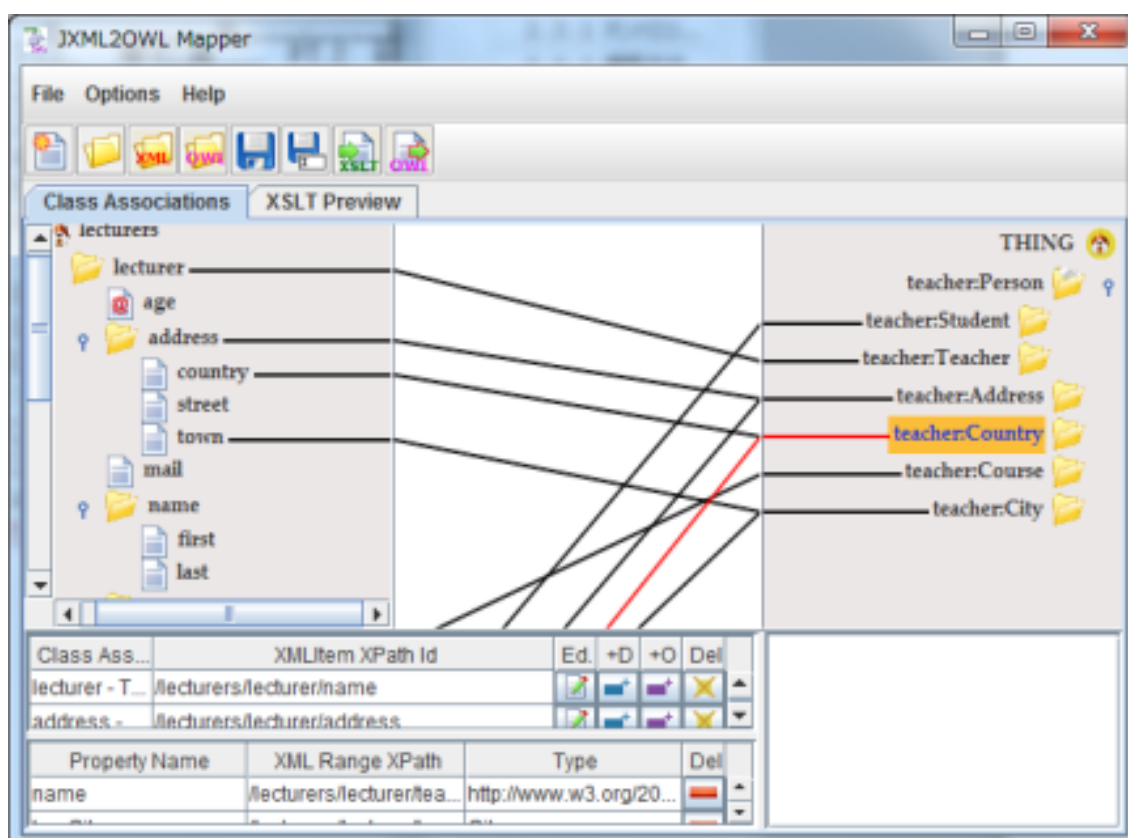


図 3-5 JXML2OWL を利用した変換定義作成

JXML2OWL Mapper へ検証対象の XML Schema（または XML 文書）と文書ドメインオントロジーを与えて変換定義を作成することで、妥当性検証システムのオントロジーコンバーターとして利用可能な XSLT スタイルシートを作成することができる。もし JXML2OWL で完全な変換定義が行えない場合でも、利用可能な部分のみ変換定義を行って XSLT スタイルシートを出力し、これを雛形として必要な追加、修正を行って再利用することで、比較的容易にオントロジーコンバーターを実装することが可能である。

### 3.5 評価

提案手法により、構築した文書ドメインオントロジーと検証オントロジーの範囲において、検証対象の構造化文書が妥当であるか、妥当であるとは言えないかを判断することが可能である。

妥当とは言えない場合は、その原因として主に、文書内容の記載に不備がある場合（必要な記載項目が抜けている等）と記載した値に問題がある場合（記載内容に齟齬ある場合や不正な値が設定されている場合等）が考えられる。本システムでは妥当と判断されない場合に、妥当でない個所やその原因を特定して明示することはできない。なぜなら、妥当であるかどうかの判断を、個体が検証項目に対して妥当であることを表すクラスのインスタンスであるかどうかで判定するのみで行っているため、妥当とならない問題点を検出する仕組みをもっていないことによる。

しかしながら、直接問題点を明示する仕組みはないものの、本システムの検証後出力される結果オントロジーには、推論結果だけでなく推論過程で導出されたクラスやプロパティを含んでいるため、Protégé 等のオントロジーエディターを用いることで記載項目の何が OK で何が OK となっていないかを把握することが可能である。これにより、利用者は問題となる個所を追跡するための情報を得ることができ、問題点を容易



に把握して修正することが可能である。

なお、本システムは文書内容の妥当性検証を目的とするシステムであり、入力された文書構造をオントロロジーへ変換する。そして入力元文書構造とは独立した意味レベルで検証処理を行う。従って検証対象の構造化文書そのものが電子申請や EDI で要求されるファイル形式に適合しているかどうかについては検証を行うことができない。オントロロジーコンバーターは要求されるファイル形式を元に変換処理を行うため、要求されるファイル形式を満たしていない場合は変換失敗または変換結果に欠落等の問題を生じることから、妥当であると判断する可能性は低い。が、有用な結果オントロロジーを出力することができない。従って本システムへ構造化文書を入力する前に、文書構造のチェックツール等を利用して入力ファイルが仕様に沿った形式かどうかを検証しておく必要がある。つまり本システムは既存の文書構造チェックツール等を置き換えるものではなく、文書構造チェックツールと併用することが重要である。実装例として、オントロロジーコンバーターの実行前に文書構造チェックツールを実行することにより構文エラーを検出する、あるいはオントロロジーコンバーターの一部として文書構造チェック機能を持たせることにより、あらかじめ構文エラーを検出する。構文エラーが無い場合にのみ妥当性検証システムを実行することで文書構造レベルの問題と意味レベルの妥当性検証結果を統合して利用者へ提示することが可能である。

一方、提案システムが妥当であるべき文書を全て正しく妥当であると判断できるようにするためには、漏れ無く検証項目が列挙され、かつ正しくその検証ルールが記述されている完全な検証オントロロジーを構築する必要がある。検証オントロロジーに間違いや不備がある場合、本来妥当であるべき文書を妥当でないと誤った判断をする可能性がある。また妥当でない文書を妥当であると誤った判断してしまう可能性もある。本手法ではオントロロジーを基盤として利用することで、専門家と情報処理技術者の知識共有が進み、共同作業がスムーズに進むことが期待できる。これにより、漏れや間違いが少ない検証

オントロジーを作成することが可能である。しかし、一般に複雑な専門分野の完全な知識を獲得して完全な検証オントロジーを構築することは困難である。一方で、本システムで利用するオントロジーの目的は構造化文書の妥当性検証の推論であることから、妥当性検証に必要となる知識という観点で作成するオントロジーの対象知識を絞ることが可能である。また具体的な対象文書の妥当性検証を行うことで、オントロジーの記述内容のテストも容易に行うことができる。このことから、妥当性検証システムに特化したオントロジー構築方法を提案した。この方法は TOVE 方法論[25]に用いられる能力質問 (Competency Question) を検証ルールと対応付けて考えることもできる。また文書の妥当性検証というプログラム開発要素が強い分野のオントロジー構築ということから、近年ソフトウェア開発に多く用いられる反復型開発あるいはアジャイルソフトウェア開発のように、短いサイクルでオントロジーの対象（文書の検証対象）を絞って作成、テスト、改良を繰り返して順次オントロジーの完成度を高める手法を適用しやすい。本構築方法を用いることで、最初から全ての業務分野を網羅するのではなく、重要度の高いものから適用範囲を明確にしてオントロジーを構築し、段階的に利用と拡張を繰り返すことで、実用性の高い妥当性検証が行えるものとする。

### 3.6 まとめ

本章では、専門的知識を必要とする分野における構造化文書内容の妥当性検証を行うことを目的として、オントロジーを利用した推論を行うことにより構造化文書内容の妥当性検証を行うシステムを提案した。

妥当性検証システムは、構造化文書の内容を OWL で表現するための語彙を定義する文書ドメインオントロジーと、文書内容に対する検証項目が妥当であることを表すクラスとそれを導出する推論規則である検証ルールを定義する検証オントロジーの二つのオントロジーを利用した推論を行うことにより構造化文書内容の妥当性を判断する。こ

のような推論を行うためのオントロジーを、専門家と情報処理技術者が共同して構築するための構築方法と、妥当性検証システムの実装方法を示した。

妥当性検証システムは、構築した文書ドメインオントロジーと検証オントロジーの範囲において、検証対象の構造化文書が妥当であるかどうかを判断することができる。また妥当かどうかの判断だけでなく、推論途中に導出される妥当と判断した根拠情報を含む結果オントロジーを検証結果と併せて出力する。妥当でない場合には、結果オントロジーを利用することで何が OK で、何が OK でないかを把握できる。さらにエラー原因の調査にも利用できるなどの有用性を確認できた。オントロジーの構築という観点からは、専門家と情報技術者が知識共有を行い、共同作業を行いながら継続的にオントロジーを構築する具体的な手順を示した。この手順を用いて重要度の高いものから適用範囲を明確にしてオントロジーを構築し、段階的に利用と拡張を繰り返すことで、実用性の高い妥当性検証が行えることを示した。またオントロジーを OWL と SWRL の記述に加えて、OWL と SWRL では記述できない拡張ルールの仕組みを導入することで、複雑な妥当性検証ルールを記述することができることを示した。最後に提案システムのコンポーネントの構成は、カスタマイズが容易で、かつ既存のツールの再利用により構築できることを示した。これにより対象分野に特化したオントロジーの作成に注力だけでなく、妥当性検証システムの開発コストとメンテナンスコストを小さく抑えられるため有用である。

妥当性検証システムの今後の課題を述べる。提案手法では、構造化文書内容が妥当か妥当でないかの判断はできるが、妥当と判断されない場合に、妥当でない個所やその原因を特定して明示することはできない。そこで、文書内容が妥当かどうかの結果だけでなく、妥当でない場合にエラー原因の提示や妥当なデータへ修正するためには何が必要か等の情報を併せて提示できる手法の開発が課題となる。

## 第4章 不動産登記申請の妥当性検証

本章ではオントロジーを利用した構造化文書の妥当性検証システムを不動産登記申請内容の妥当性検証へ適用したシステムと評価について述べる。

### 4.1 概要

不動産登記は不動産の現況と権利関係を登記簿に記録して公示する制度である。登記手続は、不動産取引の安全と権利の保護並びに登記の真正確保の観点から、申請情報に登記事項を記載するだけでなく、本人性や登記原因を証明する添付情報を提供する必要がある。妥当な登記申請情報を作成するためには、事実を証明する調査や多くの法律知識を必要とするため、その多くは法律専門家によって行われている。

近年、大型マンション建築による一度に大量の登記の発生、ゴルフ場のような多数の関与者が複雑に絡み合う案件等、扱う案件が大規模化、複雑化する傾向がある。このような登記案件の量、複雑さの増大により、登記に必要とされる正確性や取引を円滑に進めるための迅速性を確保するため、法律専門家が行う知的作業部分に対し、より高度な支援が求められている。

一方、電子署名法の成立、不動産登記法の改正等により登記の電子申請が可能となり、法務省や民間ベンダからオンラインで登記申請を行うソフトウェアも提供され、コンピュータによる登記申請の支援の環境も整いつつあるが、これらの申請用ソフトウェアは、XML ファイル仕様等に基づくメッセージ形式やプロトコル仕様のチェックを行う機能は装備されるものの、申請情報の意味に踏み込んだ妥当性を検証できるものはない。結果、申請書作成支援ソフトウェアで申請情報をエラーなく送信しても、登記官の審査に

よって、補正や却下となる可能性がある。

そこで申請情報の意味に踏み込んだ妥当性検証を行えるように、オントロジーを利用した構造化文書の妥当性検証システムを不動産登記申請の妥当性検証へ適用したシステムを提案し、そのプロトタイプシステムを構築する。

本システムの構築と運用の概要を図 4-1 に示す。

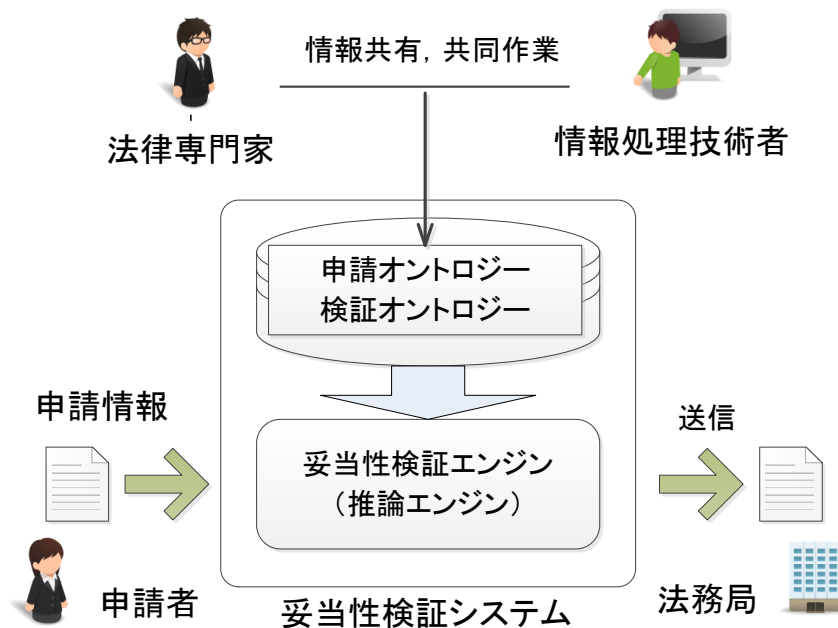


図 4-1 システム構築と運用

不動産登記申請に必要となる知識は広く深いため、妥当性検証を行うための申請オントロジーと検証オントロジーは法律専門家と情報処理技術者が共同して構築し、妥当性検証システムへ組み込む。ここで申請オントロジーとは妥当性検証システムの文書ドメインオントロジーのことである。

申請者は申請情報を法務局へ送信する前に妥当性検証システムを利用して妥当性検証を行う。妥当性検証システムは法律専門家と情報処理技術者が構築したオントロジーを用いた推論を妥当性検証エンジンが実行することにより妥当性を判断する。妥当と判断されれば申請情報を法務局へ送信する。

法改正等によりオントロジーに変更が必要となった場合も法律専門家と情報処理技術者の共同作業によりオントロジーのメンテナンスを行う。

以降、本章の構成は以下の通りである。まず4.2節で不動産登記の概要について述べる。4.3節で不動産登記申請に対する妥当性検証システムの概要を述べる。4.4節で申請オントロジーの構築と申請データの例について述べる。4.5節で検証オントロジーの構築について述べる。4.6節で妥当性検証システムの動作定義と推論例について述べる。4.7節で評価と考察について述べる。最後に4.8節で本章のまとめと今後の課題について述べる。

## 4.2 不動産登記概要

土地の売買、マンション購入など、物権の変動が発生する場合には不動産登記が行われる。不動産登記は、土地や建物の物理的状況（例えば所在、面積等）と権利関係（例えば所有者の住所氏名等）を、法務局という国家機関が管理する登記簿に正確に公示することにより、所有権等の権利に対抗力を付与し、保護を図るとともに、これを一般に公開することにより、権利関係などの状況が誰にでもわかるようにし、不動産取引の安全と円滑を図る制度である。

登記手続は、不動産取引の安全と権利の保護並びに登記の真正確保の観点から、申請情報に登記事項を記載するだけでなく、本人性や登記原因を証明する添付情報を提供する必要がある。登記申請の内容を確定するには、法的判断により要件事実を認定して申請構造を決定していく等、事実を証明する調査や多くの法律知識を必要とすることから、法律専門家が行うことが多い。そして、登記申請が行われると、登記官が申請情報と事実を証する情報である添付情報、登記簿等との記載内容に齟齬がないことを確認する形式審査を行い、妥当であれば登記を受理して登記簿へ記載する。

以下に、単純化した不動産登記申請の例を示す。

『AさんがBさんに土地Tを売った.』という法律行為により、『Aさんの所有権がBさんに移転する』という法律効果が導かれ、それを第三者に対抗するため、所有権移転登記が必要となる。この場合の登記申請書への主な記載事項は以下となる。

- a) 登記の目的：所有権移転
- b) 原因：売買
- c) 権利者：B
- d) 義務者：A
- e) 不動産の表示：土地T

さらにこの登記事項の真実性を証明するため主に以下の添付情報が必要となる。

①登記原因証明情報

売買という法律行為及び原因を証明するもので、売買契約書もしくは司法書士等の専門家が作成する情報。

②登記識別情報（登記済証）

売主が権利者となったときに交付された情報。この情報は現在の所有者しか持ちえないものであることから、売主の本人確認に加え、売買の意思の確認になる。

③住所証明書

権利者（買主）の住民票。権利者の実在証明になる。

④印鑑証明書

義務者本人が申請している事実を証明するもの。

上記内容の登記申請情報を法務局へ申請すると、登記官は、申請情報の登記の目的、原因等から必要な申請情報の記載事項、添付情報を決定し、その申請情報の記載事項と事実を証する情報（登記簿、添付情報等）の記載内容に齟齬がないことを確認する形式審査を行うことで登記の妥当性を判断し、妥当であれば登記を受理し、登記簿へ記載す

る。

### 4.3 不動産登記申請妥当性検証システムの概要

本システムは、与えられた登記申請情報が必要とされる記載事項と添付情報を持ち、その記載事項が事実を表す情報（登記簿、添付情報等）と齟齬がないことを確認することにより、登記官が行う形式審査と同様の登記申請情報に対する妥当性を検証するものである[9]。

本システムの概要を図 4-2 に示す。

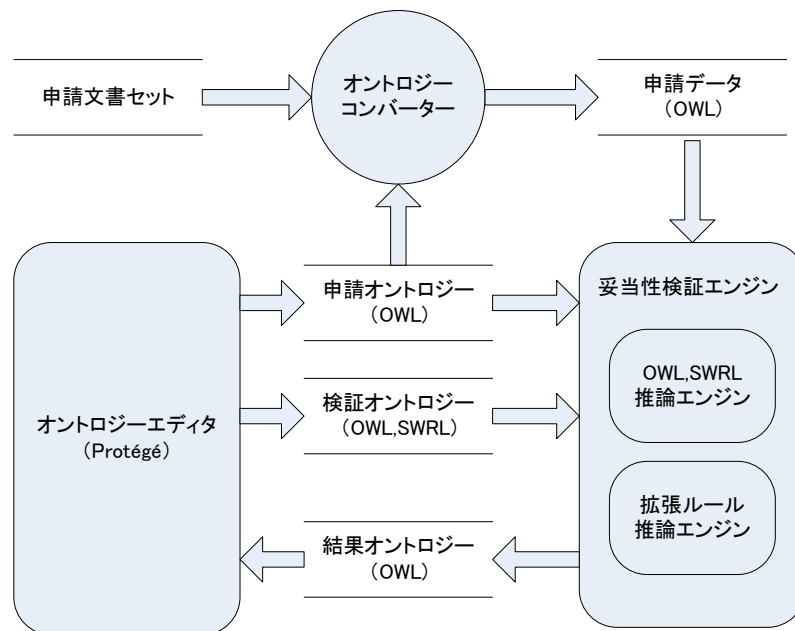


図 4-2 不動産登記申請の妥当性検証システム概要

本システムは、検証対象である申請情報（申請文書セット）を与えると、オントロジーを用いた申請情報の妥当性検証を行う推論を行うことで申請情報の妥当性を判定し、判定結果を表示するとともにその判断の根拠を含む結果オントロジーを出力するシステムである。各オントロジーはオントロジーエディターを利用して作成や表示を行う。

本システムは、妥当性検証の推論に法律専門家と情報技術者が作成した申請オントロ



ジーと検証オントロジーの2つのオントロジーを利用する。申請オントロジーとは登記申請に関する知識（クラス、プロパティ等）を OWL により記述したものである。検証オントロジーとは、申請情報に対する検証項目が妥当であることを表すクラスとそれを導出する推論規則にあたる検証ルールを申請オントロジーの語彙を用いて OWL, SWRL, 拡張ルールで記述したものである。ここで拡張ルールとは OWL, SWRL では記述できないルールを Java 言語等のプログラミング言語を用いて実装するものである。

申請文書セットとは、登記申請の際に登記所へ提出する申請情報一式を意味し、1つの登記申請書とその事実情報等を記載した複数の添付情報で構成されるもので、XML 形式等の構造化文書で記述する。申請データとは申請文書セットの記述内容を申請オントロジーの語彙を用いて OWL で記述したものである。

本システムの妥当性検証処理は以下のように行われる。申請文書セットが与えられたとき、まずオントロジーコンバーターが申請文書セットの内容を申請オントロジーで定義された語彙を用いて OWL 表現へ変換し、申請データとして出力する。続いて妥当性検証エンジンが検証データに対して申請オントロジーと検証オントロジー、拡張ルールに基づく推論を実行する。推論が停止すると、文書内容の妥当性検証結果を表示するとともに、その根拠情報となる推論途中に導出された情報を含む結果オントロジーを出力して終了する。

ここで申請文書セットが妥当であるかどうかは、申請情報全体を表す個体が、推論停止時点において OK 申請情報（検証オントロジーで定義された申請情報全体が妥当であることを表すクラス）のインスタンスとなる場合にのみ妥当な申請情報であると判定して“妥当である”と表示する。OK 申請情報のインスタンスとならない場合は“妥当とは言えない”と表示する。妥当とは言えない理由は必要な記載項目の欠如や記載内容に齟齬がある等が原因と考えられ、出力された結果オントロジーを参照することで問題を特定するためのヒント情報を得ることができる。

本システムの構成は第3章で示した妥当性検証システムの構成と同じであるが、対象領域が不動産登記申請であることにあわせて、文書ドメインオントロジーを申請オントロジー、検証データを申請データ、OKDocument を OK 申請情報と用語を変更している。

以下にこれらのオントロジーの構成方法を示すとともに、検証システムの動作について述べる。

#### 4.4 申請オントロジーと申請データ

##### 4.4.1 申請オントロジーの構築

本システムの土台として申請オントロジーを構築する。これは申請情報、添付情報、登記簿等の申請書内容を記述するために必要となるクラスやプロパティ等を OWL を用いて記述したものである。検証に固有のクラスやプロパティ等は後述の検証オントロジーで定義し、申請オントロジーには含めない。

本システムの目的は実際の登記申請情報の検証である。検証は申請オントロジー上で表現された申請データに対して行うことから、実際の申請情報を機械的に申請オントロジー上の表現へ等価に変換できる必要がある。そこで申請オントロジーの構築は、法律を出発点に実務上の書式を用いて以下の手順で構築した。

- ① 不動産登記法で定義される内容からクラスやプロパティ候補を抽出する。
- ② 申請情報の記載項目や添付情報等の必要な情報は、省令や不動産登記に関する先例、実務で用いられる登記申請書の書式、専門家へのインタビューにより得て、クラスやプロパティ候補を抽出する。
- ③ 調査した書式、添付情報等を中心に、クラス、オブジェクトプロパティ、データプロパティの切り分けを行う。書式には記載項目と、その項目に対する具体的な

値を示すものが多く、記載項目にあたるものはクラス候補、値はデータプロパティ候補とする。このとき値の例も抽出してクラスとの関係を検討する。また書式では記載項目の概念レベルで字下げや表組が行われることが多く、これを参考に申請情報との全体一部分関係を考え、クラス間のオブジェクトプロパティ候補を検討する。様々な書式に対して以上の検討を行い、不動産登記法の定義等から抽出したクラス、プロパティ等との対応をとり、クラスやプロパティを確定する。

- ④ 個体とプロパティからクラスを推論できるように、プロパティの定義域 (Domain)、値域 (Range) を適切に設定する。またクラスやプロパティの制約等も法律要件等から適切に設定する。

以下にオントロジー構築の具体例を示す。例えば、売買による所有権移転の申請情報には以下のような登記の目的と原因の記載が必要である。

登記の目的    所有権移転

原            因    平成○年△月□日売買

申請情報に記載項目として登記の目的、原因があり、これらは不動産登記法で示される概念であるので登記の目的クラス、登記原因クラスとする。これらは申請情報の一部を構成するのでオブジェクトプロパティ (has 登記の目的, has 登記原因) を定義し、申請情報の個体と関連付け、これらプロパティから登記の目的と登記原因クラスを推論できるようにオブジェクトプロパティの値域 (Range) を設定する。また登記の目的や原因は、その申請要件から申請情報に対して1つのみであることから関数型プロパティ制約を設定する。登記の目的は所有権移転等の文字列値を持ち、原因は原因日付と売買等の原因を表す文字列値をもつ。これらはデータプロパティで定義する。これらを以下

のように OWL で記述する．（以下は Manchester Syntax[38]により表記する）

Class: 申請情報

Class: 登記の目的

Class: 登記原因

ObjectProperty: has 登記の目的

Characteristics: Functional

Range: 登記の目的

ObjectProperty: has 登記原因

Characteristics: Functional

Range: 登記原因

DataProperty: has 目的表記

Characteristics: Functional

DataProperty: has 原因表記

Characteristics: Functional

DataProperty: has 原因日付

Characteristics: Functional

その他，本システムの申請オントロジーの例を図 4-3 に示す．（以下は所有権に関するものを中心とし，登録免許税や代理申請等の内容を省略した単純な構成である）

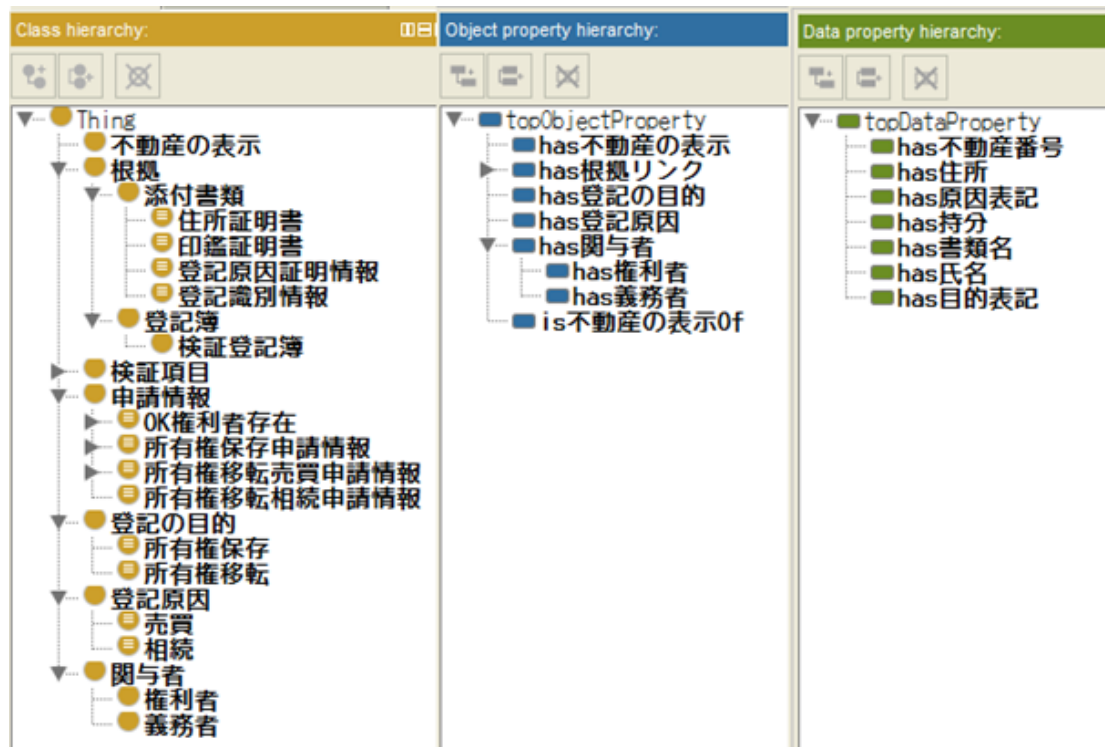


図 4-3 申請オントロジーの例

#### 4.4.2 オントロジーコンバーター

オントロジーコンバーターは申請文書セットの内容を申請オントロジーで定義された語彙を用いて OWL 表現へ変換し、申請データとして出力するコンポーネントである。

オントロジーコンバーターの実装は、既存の XML から OWL への変換ツール等を利用して比較的容易に実装することが可能である。

#### 4.4.3 申請データ

申請データとは、具体的な申請情報と事実を表す情報（添付情報、登記簿等）であり、個体と個体間の関係を申請オントロジーのプロパティを用いて OWL で記述したものである。本システムでは、申請データには基本的にクラスに関する宣言を含まず、推論により申請オントロジーで定義されるクラス等が導出されることになる。但し、検証対

象の申請文書セット全体を表す個体については、その個体が申請文書セット全体を表すクラスである申請情報クラスのインスタンスであること妥当性検証システムが認識できるように、個体とクラスの間を宣言する。

図 4-4 に申請データの例を示す。ここに示すクラスやプロパティは申請オントロジで定義されたものである。この図において、楕円はクラス、丸印は個体、長方形はリテラルを表す。個体からリテラルへ向かう矢印はデータプロパティ、個体から個体への矢印はオブジェクトプロパティ、個体からクラスへ向かう矢印は個体がクラスのインスタンスであることを表す。

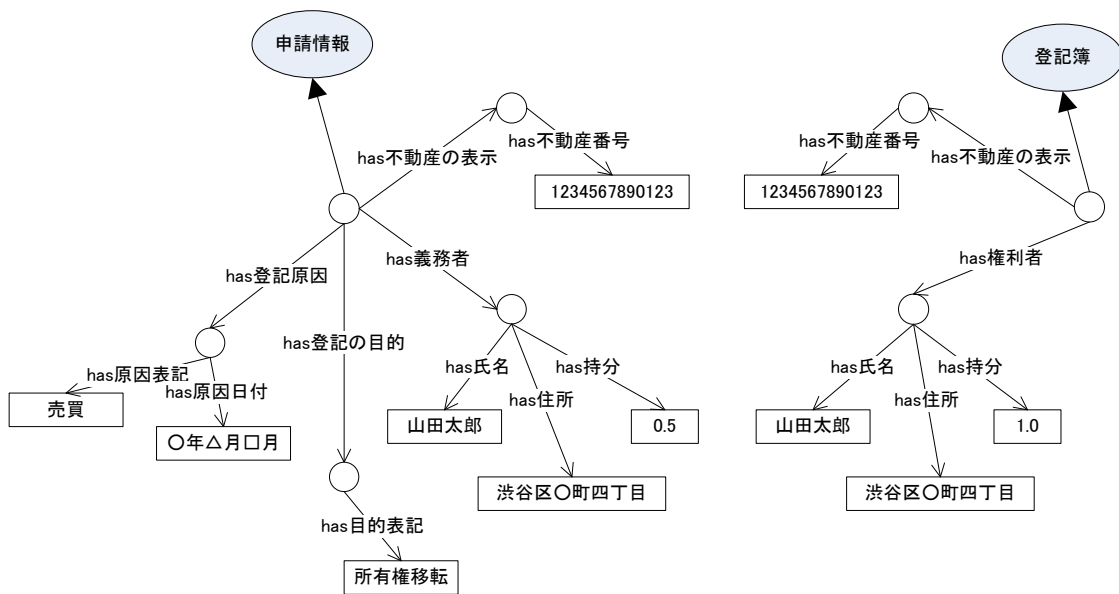


図 4-4 申請データ例

この申請データは申請オントロジーの定義に基づいて推論により個体とクラスの関係が導出される。このイメージを図 4-5 に示す。なお、この図において、破線のクラスやプロパティは推論により導かれた情報を表す。

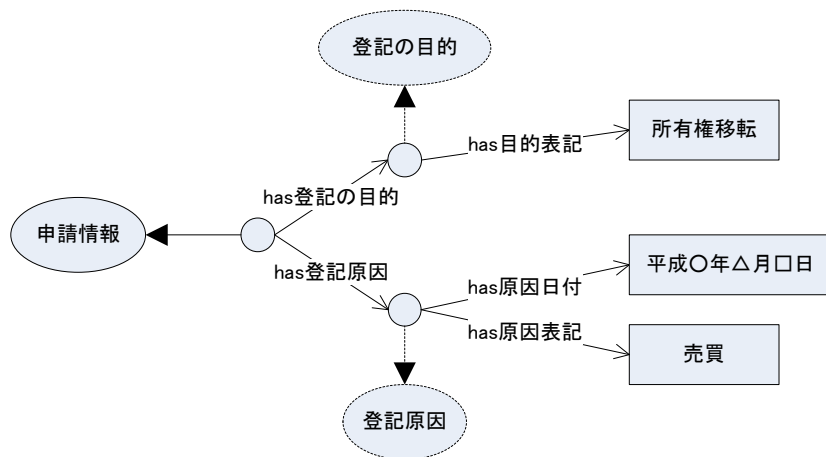


図 4-5 申請データのクラス推論例

まず，has 登記の目的の値となる個体が，値域（Range）の定義により登記の目的クラスのインスタンスであることが推論される．同様に has 登記原因の値の個体が登記原因クラスのインスタンスであることが推論される．

#### 4.5 検証オントロジー

検証オントロジーは，検証項目が妥当であることを示すクラスと，その検証ルールを申請オントロジーの語彙を用いて OWL，SWRL と拡張ルールで記述したものであり，検証システムの推論で利用される．

妥当性検証システムは，与えられた個体が，ある検証項目に対して妥当かどうかの判定を以下のように定義する．

“検証項目 Y が妥当であることを表すクラスを OKY とすると，与えられた個体 X が検証項目 Y に対して妥当と判定するのは，X が OKY のインスタンスであるときのみである．”

このとき，個体 X は検証項目 Y に対して妥当であるという．

また，個体 X が検証項目 Y に対して妥当であることを検証するルールを以下のように

に定義する.

IF X が検証項目(Y)の必要条件を満たす

THEN X は OKY のインスタンスである (X は Y に対して妥当である)

ここで直接“X は OKY のインスタンスである”を帰結してもよいし、直接帰結せず、関連する個体間のオブジェクトプロパティを帰結し、その推論により間接的に“X は OKY のインスタンスである”を導出してもよい.

以上の考え方にに基づき、与えられた申請データが妥当かどうかは、申請情報が妥当であることを表す“OK 申請情報”クラスとし、申請情報の個体が OK 申請情報のインスタンスかどうかで判定する。(本システムでは、検証項目に対応するクラス名を“OK 検証項目”のように、検証項目名の前に OK を付けて命名する。なお、妥当性検証システムの OKDocument を不動産登記申請に合わせて OK 申請情報と名前を変更しているが意図は同じである.)

本システムでは拡張ルールを除いてオントロジーエディターである Protégé を利用して作成した。図 4-6 に検証オントロジーの作成例を示す。

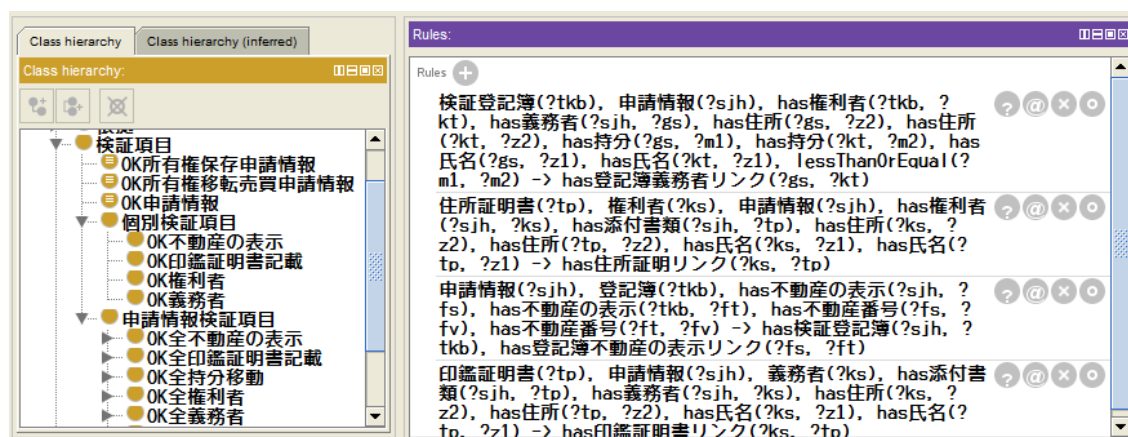


図 4-6 検証オントロジーの作成例

ところで、登記申請は、記載項目を定型フォーム化できるものではなく、申請内容に



より必要とされる記載事項が異なる性質をもつ。申請に対して必要な検証項目を決定するには、まず申請情報に必要な記載事項を決定する必要がある。この必要な記載事項は、登記の目的と登記原因からその多くを決定できることから、本システムでは、その現実的な組合せを検証パターンとして定義する。そして申請全体としての妥当性（OK 申請情報）を、検証パターンと、その検証パターンで必要とされる記載内容に対する妥当性検証項目を組み合わせて定義する。

以下では OK 申請情報の定義方法、検証パターンの決定方法、記載内容に対する検証ルールの記述方法について詳細に述べる。

#### 4.5.1 OK 申請情報

申請情報が妥当であることを表すクラス“OK 申請情報”は、検証パターンと、その検証パターンで必要とされる記載内容に対する妥当性検証項目を組み合わせて定義する。検証内容が妥当であることを表すクラスを命題だと考えると、その定義は命題論理を用いて以下のように表現できる。

$$\text{OK 申請情報} \Leftrightarrow \text{OK 所有権保存申請情報} \vee \text{OK 所有権移転売買申請情報} \dots$$

$$\text{OK 所有権保存申請情報} \Leftrightarrow$$

$$\text{所有権保存申請情報} \wedge \text{OK 全不動産の表示} \wedge \text{OK 全権利者} \wedge \dots$$

$$\text{OK 所有権移転売買申請情報} \Leftrightarrow$$

$$\text{所有権移転売買申請情報} \wedge \text{OK 全不動産の表示} \wedge \text{OK 全権利者}$$

$$\wedge \text{OK 全義務者} \wedge \text{OK 全持分移動} \wedge \text{OK 全印鑑証明書記載} \dots$$

ここで、所有権保存申請情報、所有権移転売買申請情報は、検証パターンを表し、OK 全不動産の表示等は記載項目が妥当である必要がある個々の検証項目を表している。

上記命題論理表現を OWL のクラスで考えなおすと、クラスの論理和(or)、論理積(and)、等価公理(EquivalentTo)を用いて、それぞれ以下のように表現できる。

Class: OK 申請情報 EquivalentTo:

OK 所有権保存申請情報 or OK 所有権移転売買申請情報 or …

Class: OK 所有権保存申請情報 EquivalentTo:

所有権保存申請情報 and OK 全不動産の表示  
and OK 全権利者 and …

Class: OK 所有権移転売買申請 EquivalentTo:

所有権移転売買申請情報 and OK 全不動産の表示  
and OK 全権利者 and OK 全義務者 and OK 全持分移動  
and OK 全印鑑証明書記載 and …

これにより、申請情報のインスタンスが OK 申請情報のインスタンスであるかどうかの判定の解釈は、OK 所有権保存申請情報あるいは OK 所有権移転売買申請情報等の何れかのインスタンスである必要があり、さらに、例えば OK 所有権保存申請情報のインスタンスであるためには、OK 所有権保存申請情報の前提となる所有権保存申請情報、OK 全不動産の表示、OK 全権利者等の全てのクラスのインスタンスである必要がある、となる。

上記クラス関係を表すイメージを図 4-7 に示す。

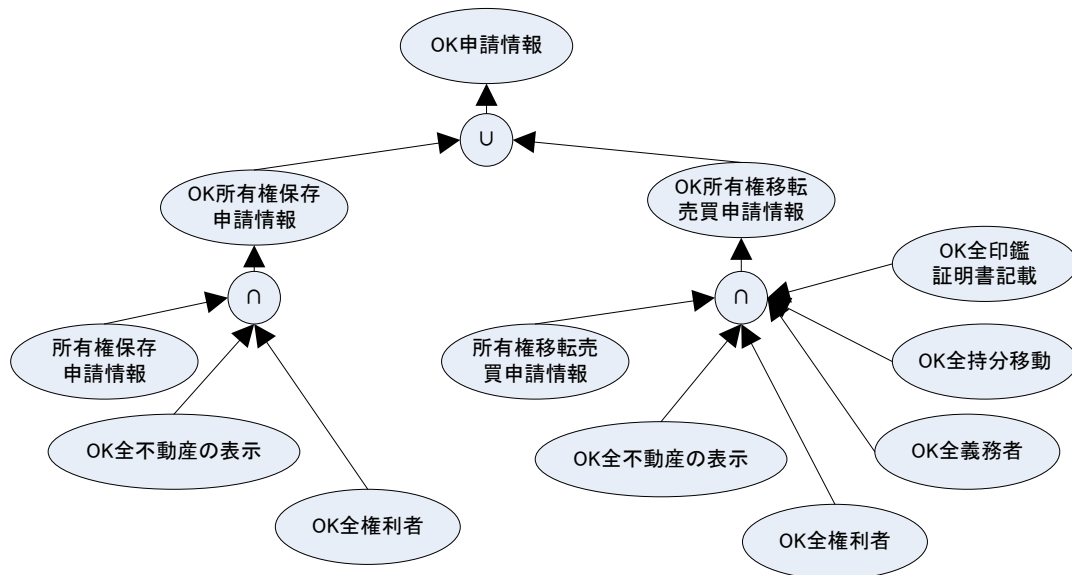


図 4-7 OK 申請情報の定義例

#### 4.5.2 検証パターンの決定

検証パターンは OK 申請情報の定義で利用されるため、申請情報のインスタンスを対象とした階層で定義する必要がある。一方、登記の目的と登記原因の内容は、登記の目的クラスと登記原因クラスのデータプロパティで与えられることから、検証パターンを表すクラスは、登記の目的や登記原因に与えられたデータプロパティの値から等価公理を用いて推論により導出できるように定義する。

検証パターンを表すクラス定義の具体例として、所有権移転売買申請情報を示す。まず、登記の目的クラスのデータプロパティである **has** 目的表記の値に応じて、等価公理を用いて登記の目的の派生クラスである所有権移転を定義する。登記原因に対しても同様に、登記原因の派生クラスである売買を定義する。

Class: 所有権移転 EquivalentTo:

登記原因 and (has 原因表記 value "売買")

and (has 登記原因 some 売買)

```
graph LR; A([申請情報]) --> B([所有権移転]); B --> C([登記の目的]); C -.-> D(( )); D -.-> E([所有権移転]); D -- "has登記の目的" --> F([所有権移転]); D -- "has登記原因" --> G(( )); G -- "has原因日付" --> H[平成〇年△月□日]; G -- "has原因表記" --> I[売買]; G -.-> J([売買]); J --> K([登記原因]); K -.-> L([所有権移転売買申請情報]); L --> A;
```

この例では、登記の目的のインスタンスは、データプロパティである **has** 目的表記の値が所有権移転であることから所有権移転のインスタンスであると推論され、同様に登記原因のインスタンスが売買クラスのインスタンスであると推論される。申請情報のインスタンスの **has** 登記の目的の値が所有権移転のインスタンスであり、**has** 登記原因の値が売買のインスタンスであることから、結果、申請情報のインスタンスは、所有権移転売買申請情報のインスタンスであることが推論され、検証パターンが導出されることになる。

#### 4.5.3 記載内容等に関する検証ルールの記述

個々の記載内容を表す個体に対しても検証項目が妥当であることを示すクラスを導出する検証ルールを記述する。このクラスの導出方法は、以下の順に適用可能な状況を検討して記述する。

- (1) プロパティを帰結し、間接的にクラスを帰結する検証ルール (OWL, SWRL)
- (2) クラスを直接帰結する検証ルール (OWL, SWRL)
- (3) 拡張ルールによる検証処理 (OWL, SWRL では記述できない検証ルール)

ここで、OWL, SWRL の双方で記述可能なルールは、OWL で記述する。以下にこれらの適用範囲と具体例を示す。

- (1) プロパティを帰結し、間接的にクラスを帰結する検証ルール

詳細な検証項目の多くは、申請情報の記載内容と事実を表す情報（添付情報や登記簿等）の記載内容に齟齬がないかを確認するものである。このような記載情報の真実性を確認する検証ルールは、記載内容の個体と事実を表す情報の個体、関連するクラスやプ

ロパティ等のパターンマッチングとして考えることができ、その多くを SWRL で記述可能である。

SWRL の帰結部分には、検証項目が妥当であることを表すクラスを直接記述せず、記載情報の個体と事実を表す情報の個体に対し、検証結果が妥当である根拠を関連付けるオブジェクトプロパティへの宣言を帰結する。帰結するオブジェクトプロパティの定義域もしくは値域に検証項目が妥当であることを表すクラスを定義しておくことで、推論により妥当であるクラスを導出する形をとる。

これは本システムの個体と個体間のプロパティから推論によりクラス等を導出するアプローチに沿ったものであるが、これによりさらなる推論に利用できるだけでなく、記載事項と事実情報とのリンク関係（妥当性の根拠）を明示できる利点がある。

以下に例として、申請情報の不動産の表示が、登記簿の不動産の表示と一致していることを検証する SWRL ルールと、その帰結となるオブジェクトプロパティの定義を示す。

```
申請情報(?sjh),   登記簿(?tkb),  
  
has 不動産の表示(?sjh, ?fs),   has 不動産の表示(?tkb, ?ft),  
  
has 不動産番号(?fs, ?fv),   has 不動産番号(?ft, ?fv)  
  
-> has 検証登記簿(?sjh, ?tkb),   has 登記簿不動産の表示リンク(?fs, ?ft)
```

ObjectProperty: has 検証登記簿

Range: 検証登記簿

ObjectProperty: has 登記簿不動産の表示リンク

Domain: OK 不動産の表示

---

これは、申請情報と登記簿における不動産の表示のインスタンスにおいて不動産番号の値が一致すれば、申請情報と登記簿のインスタンス間に **has 検証登記簿**を設定し、さらに不動産の表示のインスタンス間に **has 登記簿不動産の表示リンク**を設定する。そして、**has 検証登記簿**の値域(Range)と **has 登記簿不動産の表示リンク**の定義域(Domain)から、登記簿が検証対象の登記簿であることを表すクラス(検証登記簿)と不動産の表示が妥当であるクラス(OK 不動産の表示)を導出する。

この推論結果を用いた例として、検証登記簿が決まると、申請情報の義務者が登記簿の権利者に記載された内容と一致する条件は、以下のように記述できる。

```

検証登記簿(?tkb),   申請情報(?sjh),

has 権利者(?tkb, ?kt), has 義務者(?sjh, ?gs),

has 住所(?gs, ?z2), has 住所(?kt, ?z2),

has 持分(?gs, ?m1), has 持分(?kt, ?m2),

has 氏名(?gs, ?z1), has 氏名(?kt, ?z1),

lessThanOrEqual(?m1, ?m2)

-> has 登記簿義務者リンク(?gs, ?kt)

```

ObjectProperty: has 登記簿義務者リンク

Domain: OK 義務者

これは、申請情報の義務者と検証登記簿の権利者の住所、氏名が一致し、その持分が登記簿に記載された持分以下であれば、インスタンス間に **has 登記簿義務者リンク**を設定する。そして **has 登記簿義務者リンク**の定義域(Domain)の定義により、申請情報の義務者の記載が妥当であることを表すクラス(OK 義務者)を導出する。

上記の不動産の表示と義務者の検証例を図 4-9 に示す。

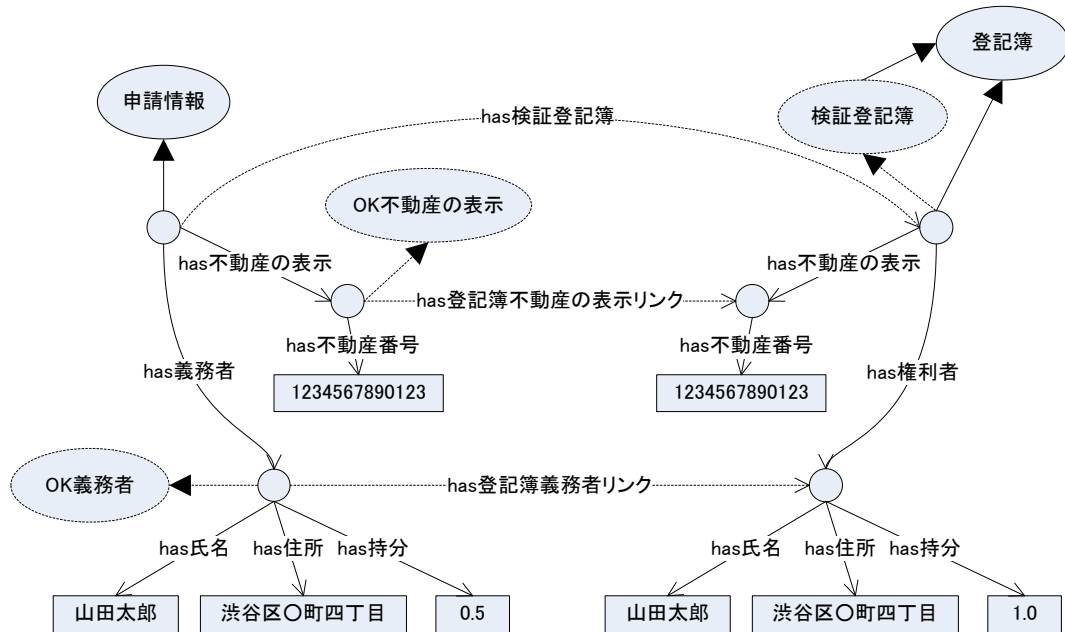


図 4-9 不動産の表示と義務者の検証例

## (2) クラスを直接帰結する検証ルール

検証項目の中には、事実情報とは無関係に記載内容自体の要件に対する検証項目がある。これは OWL の等価公理等を用いて記述できるものと SWRL で記述できるものがある。例えば、申請情報に 1 人以上の権利者が存在していることを表す OK 権利者存在というクラスは、等価公理を用いて以下のように記述できる。

Class: OK 権利者存在 EquivalentTo:

申請情報 and (has 権利者 some 権利者)

## (3) 拡張ルールによる検証処理

OWL, SWRL は、開世界仮説 (Open World Assumption) の採用と、単一名仮説



(Unique Name Assumption) を採用していないことにより、全ての個体が満たす条件の推論や否定に関する推論、数え上げ等が困難か記述不可能である。また、OWL, SWRL はプログラミング言語ではないため、複雑な計算処理等、記述できない処理も多い。

例えば、OK 全不動産の表示の検証規則の前提条件は “申請情報の全ての不動産の表示のインスタンスが OK 不動産の表示クラスのインスタンスであること” である。しかし、この前提条件を OWL, SWRL を用いて推論することは困難である。また、OK 全持分移動の検証規則の前提条件は “申請情報の全ての権利者の持分合計と全ての義務者の持分合計が等しいこと” であるが、同様に OWL, SWRL で推論することは困難である。一般に、このような推論を行えるようにするためには、世界を閉じるための特別な構造表現や個体に関する追加の宣言等が必要となり、モデルが複雑化し現実的でなくなる。

そこで、OWL, SWRL の推論モデルの制限等により記述できない検証規則を、Java 言語と OWLAPI を用いて拡張規則として実装する。拡張規則においても、先の妥当性検証規則と同じ考え方であり、拡張規則に関する推論（Java 言語による規則の適用）を行い、最終的に妥当であることを表すクラスを導出する。但し、拡張規則を適用する範囲は最小限にとどめる。例えば、プロパティを帰結する場合、プロパティからクラスを導出する規則は、検証オントロジーの OWL, SWRL を用いて定義する。

先の OK 全不動産の表示を拡張規則により推論された例を図 4-10 に示す。

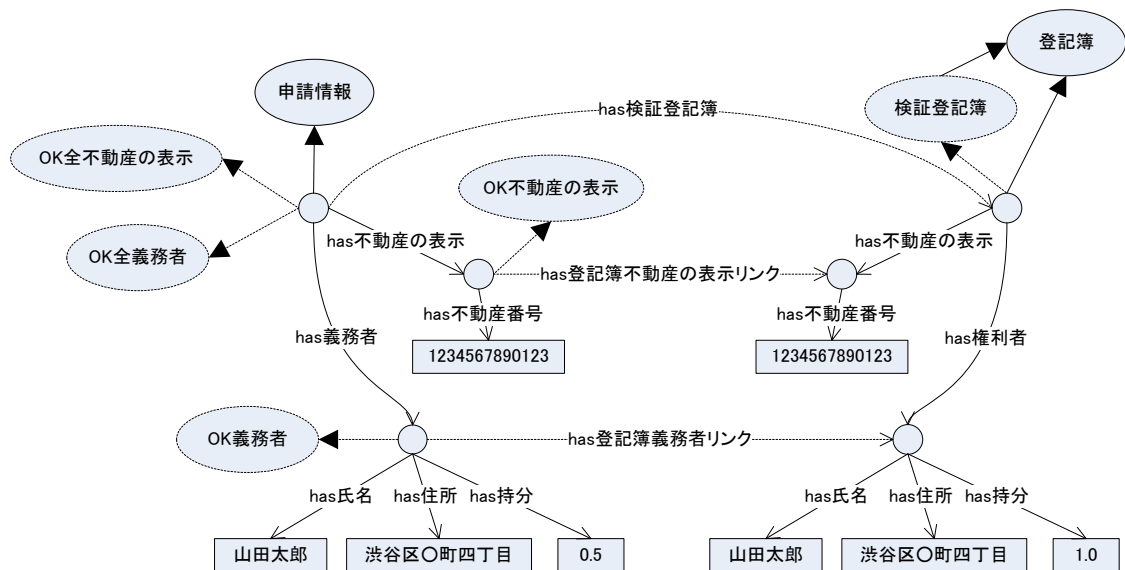


図 4-10 拡張ルール推論の例

拡張ルールを Java 言語と OWLAPI を用いて、申請情報のインスタンスから得られる has 不動産の表示の個体を列挙し、その個体全てが OK 不動産の表示クラスに属しているとき、申請情報のインスタンスを OK 全不動産の表示のインスタンスであると宣言するように実装する。推論は、まず OWL, SWRL による推論が実行され、先の SWRL の検証ルールにより、不動産の表示のインスタンスが OK 不動産の表示のインスタンスであることが導出される。その後拡張ルールを適用し、その実装から申請情報のインスタンスが OK 全不動産の表示のインスタンスであると推論される。この処理は開世界仮説を採用せず、与えられた不動産の表示のみに閉じて判定を行っている。

#### 4.6 妥当性検証システムの動作

検証システムの動作を以下のように定義する。

- ① 前処理として、入力となる申請文書セットをオントロジーコンバーターにより申請データ（OWL 形式）へ変換する。

- ② 申請オントロジー，検証オントロジー，申請データを読み込み，この状態のオントロジーを **O** とする．
- ③ **O** に対して **OWL** と **SWRL** の推論を実行し，その結果のオントロジーを **OR** とする．
- ④ **OR** に対して全ての拡張ルールを適用し，その結果のオントロジーを **OE** とする．
- ⑤ **OR**  $\neq$  **OE** のとき，新たな知識の追加があったので，**O=OE** として，③に戻る．
- ⑥ **OR**  $=$  **OE** のとき，新たな知識追加がないので，**OE** を結果オントロジーとして保存して終了．このとき，申請全体を表す個体が **OK** 申請情報クラスのインスタンスであれば，“妥当な申請情報”と判定して表示する．そうでない場合は“妥当とはいえない”と表示する．

これまでの例を元に申請情報のインスタンスが妥当と判定されるイメージを図 4-11 に示す．

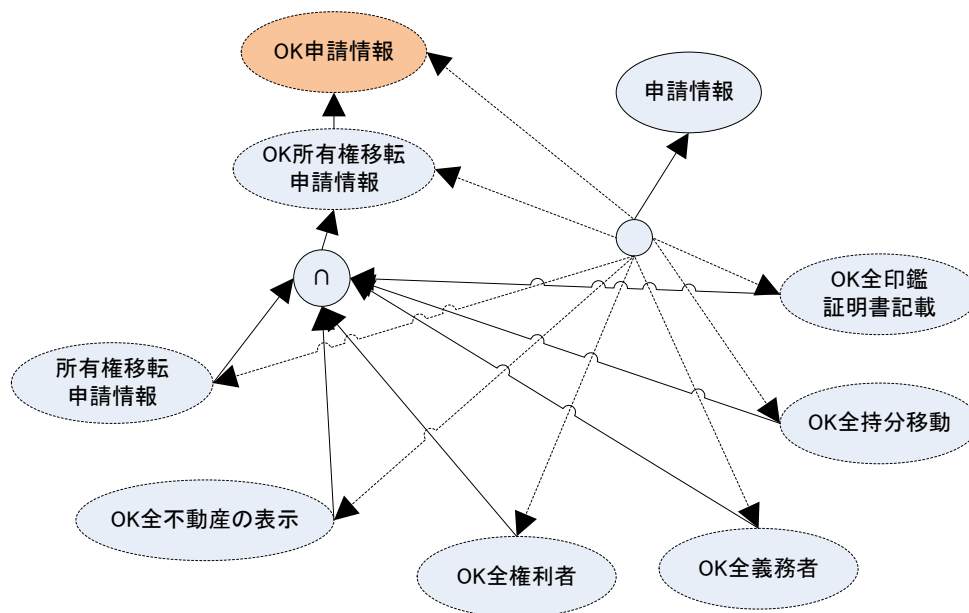


図 4-11 検証結果の例

OWL, SWRL, 拡張ルール の推論により, 申請情報のインスタンスが所有権移転申請情報, OK 全不動産の表示, OK 全権利者, OK 全義務者等の OK 所有権移転申請情報の前提となるクラス全てに属することが導出された場合, 定義により申請情報のインスタンスは OK 所有権移転申請情報のインスタンスとなる. また OK 申請情報は OK 所有権移転申請情報のサブクラスであるので, 申請情報のインスタンスは OK 申請情報のインスタンスでもあることになり, 結果, 与えられた申請情報は妥当であると判定される.

妥当性検証システムが終了したとき, 推論途中に導出された妥当性検証結果の根拠情報を含む結果オントロジーも出力される. 図 4-12 に結果オントロジーのイメージを示す. 結果オントロジーは推論により導出された各記載項目の関係も明示されるため, 検証目的以外にも有用な情報を含んでいる. また申請情報が妥当と判断されなかった場合にも, 問題が無い個所と問題がある個所を特定するための情報としても利用することができる.

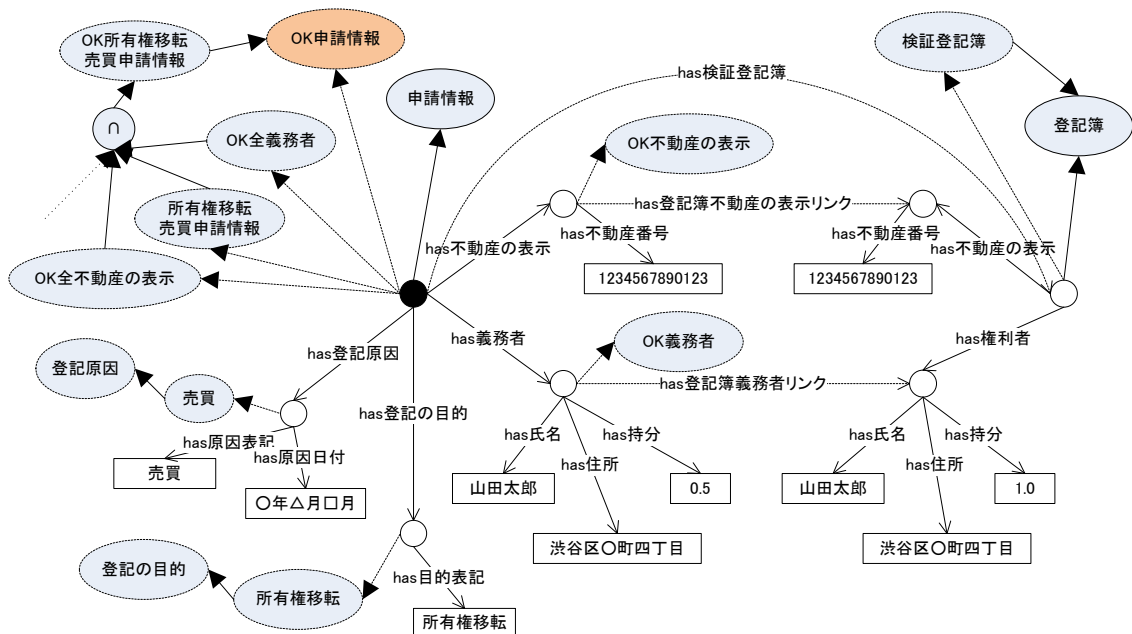


図 4-12 結果オントロジーのイメージ

## 4.7 評価と考察

### 4.7.1 動作評価

本プロトタイプシステムでは、よく行われる売買による所有権移転登記を中心に、申請オントロジーと検証オントロジーの構築を行っている。これに対して実例に即した登記申請サンプルを用意し、これを申請オントロジーの語彙を用いて OWL 化した申請データを検証システムへ与えることで動作確認を行った。

記載内容と添付情報に齟齬のない妥当な申請データサンプルを検証システムに与えた場合、各記載項目の個体が妥当であることを表すクラスのインスタンスとなり、その結果、申請情報の個体が OK 申請情報のクラスのインスタンスとなり、検証システムの出力結果は妥当となった。

妥当な場合の結果オントロジー例を Protégé に読込んだ画面を図 4-13 示す。画面左側に個体のリストがあり、個体を選択すると中央のリストにその個体が属するクラスが一覧される（クラスのインスタンスであることが分かる）。画面右のリストにはその個体のプロパティ（推論により導出されたプロパティを含む）が表示される。

ここで画面左のリストから申請情報を表す個体 sjh を選択すると、画面中央のリストに OK 申請情報クラスが表示されることから、sjh が OK 申請情報クラスのインスタンスとなっていることが分かる。また OK 全義務者などの、OK 申請情報の前提となる検証項目のインスタンスとなっていることもわかる。

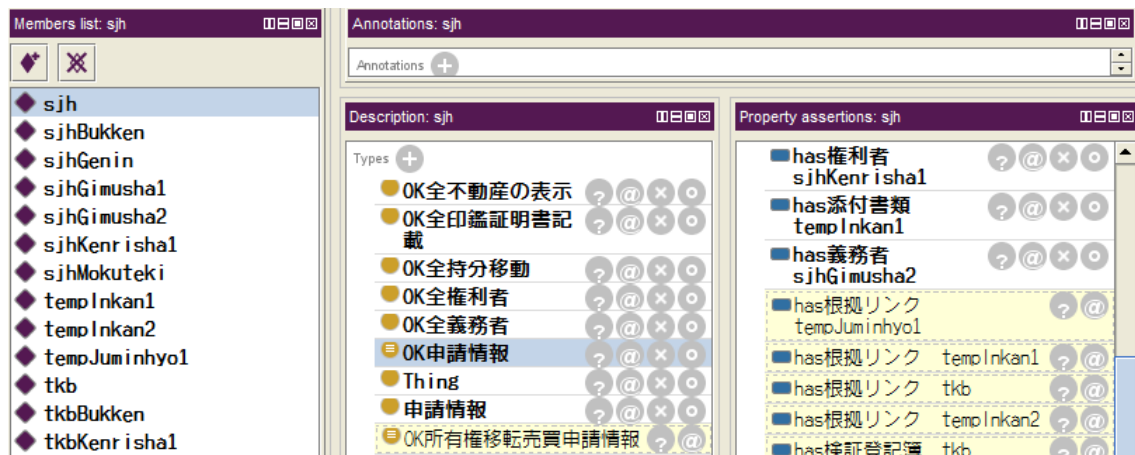


図 4-13 妥当な場合の結果オントロジー取込例

これに対し、妥当な申請データサンプルの一部の記載を変更して記載内容と添付情報に齟齬がある状態のデータを与えた場合、妥当と判断されなかった。この妥当とならなかった検証結果オントロジーをオントロジーエディターである Protégé へ取り込んで確認したところ、申請情報の個体が OK 申請情報のクラスのインスタンスとなっていなかった。また各検証項目については齟齬が無い検証項目はそれが妥当であることを表すクラスのインスタンスとなっているが、齟齬がある項目については妥当であることを表すクラスのインスタンスとならないことが確認できた。

例えば、ある義務者の住所記載を登記簿の住所と異なるものに変更した場合、申請情報の個体は OK 申請情報のインスタンスとはならず、OK 全義務者のインスタンスにもならない。

この妥当でない場合の結果オントロジー例を Protégé に読込んだ画面を図 4-14 示す。画面左のリストから申請情報を表す個体 sjh を選択すると、画面中央のリストに OK 申請情報クラスが表示されていないことから、sjh が OK 申請情報のインスタンスではないことが分かる。同様に OK 全義務者クラスも表示されていないことから、OK 全義務者のインスタンスでもないことがわかる。

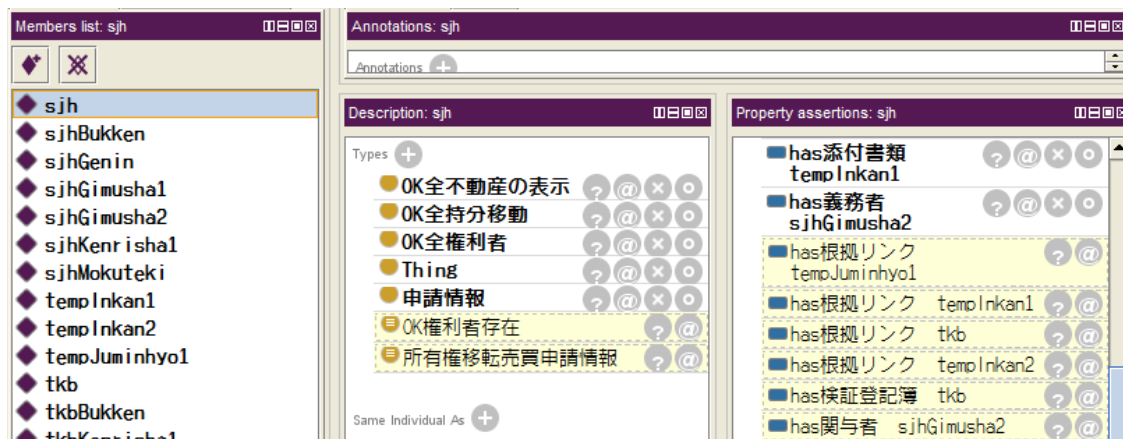


図 4-14 妥当とならない場合の結果オントロジー取込例

このことから義務者の何らかの情報に齟齬があることが分かる。また義務者が二人以上いる場合、齟齬が無い義務者は OK 義務者のインスタンスとなるが、齟齬がある義務者は OK 義務者のインスタンスとならないので問題がある義務者が特定できる。さらに、オブジェクトプロパティである has 登記簿義務者リンクもないことから登記簿との関係に問題があることがわかり、その義務者の記載データと登記簿データを見比べることで住所記載の違いを特定できる。このように妥当とならない場合においても問題が無い項目と問題がある項目を識別することが可能であることが確認できた。

義務者に問題が無い場合の結果オントロジー例を図 4-15 に示す。画面左のリストから一人目の義務者を表す个体 sjhGimusha1 を選択すると、画面中央のリストに OK 義務者クラスが表示されていることから、sjhGimusha1 が OK 義務者クラスのインスタンスであることが分かる。また has 登記簿項目リンク等のオブジェクトプロパティが存在し、OK 義務者クラスを導出した根拠となる登記簿の記載項目等との対応関係もわかる。

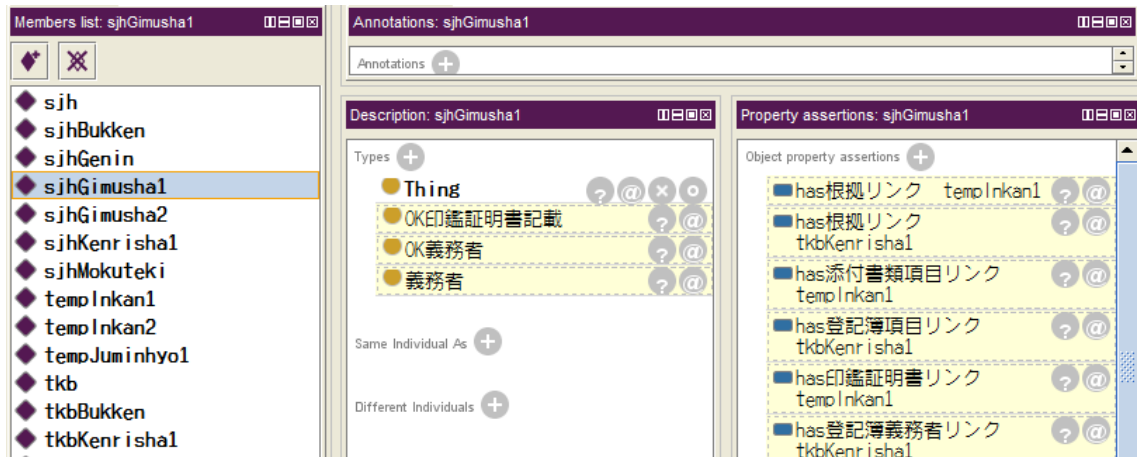


図 4-15 問題が無い義務者情報の例

義務者に問題がある場合の結果オントロジー例を図 4-16 に示す。画面左のリストから二人目の義務者を表す個体 sjhGimusha2 を選択すると、画面中央のリストに OK 義務者クラスが表示されていないことから、sjhGimusha2 が OK 義務者クラスのインスタンスではないことが分かる。また has 登記簿項目リンク等のオブジェクトプロパティが存在しないことから、登記簿の記載項目等との齟齬があることがわかる。



図 4-16 問題がある義務者情報の例

本システムにより申請データが妥当である（記載内容に齟齬がなく却下にならない可能性が高い）ことと、妥当ではない（記載内容等に齟齬があり却下になる可能性がある）ことがわかり、また問題の所在を追跡できることがわかる。しかし本システムでは妥当



でない個所や原因を特定して明示することはできない。これは妥当であるかどうかの判断を、申請情報の記載要素である個体が検証項目に対して妥当であることを表すクラスのインスタンスかどうかを判定することにより行っており、従って SWRL と拡張ルールによる検証ルールも、個体が妥当となる条件を満たせば妥当を表すクラスに属すると帰結するものとなっていることから、妥当とならない問題点を検出する仕組みをもっていないことによる。一方、登記は先願主義であるため申請した登記が却下になると不利益（抵当権の順位が確保できない等）を被る可能性があり、また代理人が申請を行う場合は信用問題になる場合もあることから、本システムにより申請前に検証を行い申請データが妥当であることを示すことは有用と考えられる。また直接問題点を明示する仕組みはないものの、本システムの検証後出力される結果オントロジ (OWL ファイル) は、推論結果だけでなく推論過程で導出されたクラスやプロパティを含んでいるため、Protégé 等のオントロジーエディターを用いて記載項目の何が OK で何が OK となっていないかを把握することが可能である。先の例のように、結果オントロジを利用することで、申請者は比較的容易に OK となっていない原因を調査して申請データを修正できることから、却下とならない登記申請データの作成に役立てることができると考えられる。さらに専門家であれば OK とならない本質的な原因の調査を行い、より高度な判断に役立てることが可能である。例えば先の例においては、単なる住所の記載ミスなのか、あるいは引越等により住所が変わっており本申請の前に名義変更登記が必要であるか等の判断を行うことができる。

#### 4.7.2 オントロジの詳細化と拡張

登記官による形式審査は、省令や先例を含む法律等により決められた記載事項や添付書類について齟齬がないかどうかを確認する作業であることから、基本的に登記官により審査結果が異なることはない。従って本システムによる妥当性検証が行える範囲は、

本システムのオントロジーとルールが実装する法律要件の範囲により決まることになる。本システムはプロトタイプとして典型的な登記に関するオントロジー構築を行った段階であり、今後さらに本オントロジーの詳細化と拡張を行う必要がある。

一方、不動産登記法は手続法であり、省令や先例等を合わせると細かく規定されているにもかかわらず、登記全般をカバーする抜けのない完全なオントロジーとルールを構築するのは困難である。例えば不動産登記法には記載事項に対して添付すべき具体的な情報を記載しておらず、受理される添付情報が何かを知るには省令や先例の他、広い法律知識が必要である。また法改正等もあることから作成済みのオントロジーを必要に応じて更新する必要もある。これらのことから様々な事例に対応できるようにオントロジーの詳細化を行うためには法律専門家との継続的な共同作業が必要となる。ここで一般に法律関係のシステム構築を行う際に問題となるのが、情報技術者は高度な法律知識をもっておらず、法律専門家も高度な情報技術知識をもっていないことから、共同作業が難しいことである。しかし本システムではオントロジー構築を基本とすることで、法律用語の概念定義の問題として考えることができ、その概念の性質や関係を表現することは法律専門家が得意とするところであるので、共同作業の基盤とすることは本システムのアプローチの利点と考えられる。

実用面では全ての登記について対応できなくても、適用範囲が明確であれば重要度の高い登記に限定したものであっても十分有用と考えられる。今後は本システムのオントロジーを専門家と協力して重要度が高く典型的な登記から継続的に詳細化していき、順次対応する登記の種類等を増やしていくように拡張を行うことで有用なオントロジーを構築できると考えられる。

#### 4.8 まとめ

本章では、不動産登記申請情報の妥当性検証を行うことを目的として、オントロジーを利用した構造化文書内容の妥当性検証を不動産登記申請の妥当性検証へ適用した。

オントロジーを用いた構造化文書の妥当性検証システムを不動産登記申請内容の妥当性検証へ適用するために、申請オントロジー（文書ドメインオントロジー）と検証オントロジーの構築方法を示し、具体的な不動産登記申請に対する妥当性検証システムの動作評価を行った。この結果、与えられた不動産登記申請情報に対して妥当性の判断を行えることを示した。また、推論過程で導出された情報を含む結果オントロジーを利用することで、妥当でない場合のエラー原因の調査目的だけではなく、妥当である場合にも人や物件、添付情報とのリンク情報、クラス情報等を含むため、利用者に対して有用な情報を提供できることを示した。また、全ての不動産登記について対応できなくても、重要度の高い登記に限定したものであっても十分有用と考えられるため、本手法により、順次対応する登記の種類等を増やしていくように拡張を行うことで有用なオントロジーを構築することが可能である。

以上により、これまで登記申請に関するシステム支援は、送信データのメッセージ形式やプロトコル仕様のチェックにとどまっていたのに対し、申請内容の検証を行うことができることを示した。本論文で提案した妥当性検証の枠組みと、既存のメッセージ形式やプロトコル仕様のチェックを組み合わせることにより、間違いのない登記申請情報の作成が可能となる。

不動産登記申請内容の妥当性検証システムの今後の課題として、本プロトタイプシステムを実用システムに適用できるようにするために、登記の目的、原因をさらに詳細化し、記載内容に対する検証ルールを増やしていく必要がある。

## 第5章 結論

本章では、本論文のまとめと今後の課題と展望について述べる。

本論文では、専門的知識を必要とする分野における構造化文書内容の妥当性検証を行うことを目的として、オントロジーを利用した推論を行うことにより構造化文書内容の妥当性検証を行うシステムを提案した。さらに提案手法を応用して不動産登記申請の妥当性検証へ適用した。

妥当性検証システムは、構造化文書の内容を OWL で表現するための語彙を定義する文書ドメインオントロジーと、文書内容に対する検証項目が妥当であることを表すクラスとそれを導出する推論規則である検証ルールを定義する検証オントロジーの二つのオントロジーを利用した推論を行うことにより構造化文書内容の妥当性を判断する。このような推論を行うためのオントロジーを、専門家と情報処理技術者が共同して構築するための構築方法と、妥当性検証システムの実装方法を示した。

妥当性検証システムは、構築した文書ドメインオントロジーと検証オントロジーの範囲において、検証対象の構造化文書が妥当であるかどうかを判断することができる。また妥当かどうかの判断だけでなく、推論途中に導出される妥当と判断した根拠情報を含む結果オントロジーを検証結果と併せて出力する。妥当でない場合には、結果オントロジーを利用することで何が OK で、何が OK でないかを把握できる。さらにエラー原因の調査にも利用できるなどの有用性を確認できた。オントロジーの構築という観点からは、専門家と情報技術者が知識共有を行い、共同作業を行いながら継続的にオントロジーを構築する具体的な手順を示した。この手順を用いて重要度の高いものから適用範囲を明確にしてオントロジーを構築し、段階的に利用と拡張を繰り返すことで、実用性の

高い妥当性検証が行えることを示した。またオントロジーを OWL と SWRL の記述に加えて、OWL と SWRL では記述できない拡張ルール仕組みを導入することで、複雑な妥当性検証ルールを記述することができることを示した。最後に提案システムのコンポーネントの構成は、カスタマイズが容易で、かつ既存のツールの再利用により構築できることを示した。これにより対象分野に特化したオントロジーの作成に注力できるだけでなく、妥当性検証システムの開発コストとメンテナンスコストを小さく抑えられるため有用である。

オントロジーを用いた構造化文書の妥当性検証システムを不動産登記申請内容の妥当性検証へ適用したシステムについて、申請オントロジー(文書ドメインオントロジー)と検証オントロジーの構築方法を示し、具体的な不動産登記申請に対する妥当性検証システムの動作評価を行った。この結果、与えられた不動産登記申請情報に対して妥当性の判断を行えることを示した。また、推論過程で導出された情報を含む結果オントロジーを利用することで、妥当でない場合のエラー原因の調査目的だけではなく、妥当である場合にも人や物件、添付情報とのリンク情報、クラス情報等を含むため、利用者に対して有用な情報を提供できることを示した。また、全ての不動産登記について対応できなくても、重要度の高い登記に限定したものであっても十分有用と考えられるため、本手法により、順次対応する登記の種類等を増やしていくように拡張を行うことで有用なオントロジーを構築することが可能である。オントロジーを用いた構造化文書の妥当性検証システムを不動産登記申請内容の妥当性検証へ適用することにより、これまで登記申請に関するシステム支援は、送信データのメッセージ形式やプロトコル仕様のチェックにとどまっていたのに対し、申請内容の検証を行うことができることを示した。本論文で提案した妥当性検証の枠組みと、既存のメッセージ形式やプロトコル仕様のチェックを組み合わせることにより、間違いのない登記申請情報の作成が可能となる。

以上により、提案システムはオントロジーを主体とすることで、複雑な妥当性検証ル

---

ールを記述でき、専門家との継続的な共同作業を可能とすることができた。また提案システムを用いることにより、政府への申請や企業間取引のデータ内容を送信する前に、送信内容が妥当であることを検証できるため、より安全な申請や取引が可能となる。

妥当性検証システムの今後の課題を述べる。提案手法では、構造化文書内容が妥当か妥当でないかの判断はできるが、妥当と判断されない場合に、妥当でない個所やその原因を特定して明示することはできない。そこで、文書内容が妥当かどうかの結果だけでなく、妥当でない場合にエラー原因の提示や妥当なデータへ修正するためには何が必要か等の情報を併せて提示できる手法の開発が課題となる。

不動産登記申請内容の妥当性検証システムの今後の課題として、本プロトタイプシステムを実用システムに適用できるようにするために、登記の目的、原因をさらに詳細化し、記載内容に対する検証ルールを増やしていく必要がある。

最後に今後の展望として、実務支援の観点からは、申請内容が妥当かの結果だけでなく、妥当でない場合、妥当な登記とするためには何が必要かも併せて提示することができれば、申請内容の検証だけでなく、法律関係の実体判断の支援も可能となることから、エラー検出などの問題に取り組んでいきたい。また、不動産登記以外の商業登記などの申請情報の妥当性検証にも本手法は有効であると考えられるため、不動産登記以外の登記に関するオントロジー構築も取り組んでいきたい。

## 参考文献

1. Tim Bray; Jean Paoli; C. M. Sperberg-McQueen; François Yergeau. Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation. 2008.  
<http://www.w3.org/TR/REC-xml/>.
2. Eric van der Vlist. XML Schema.オライリージャパン, 2003.
3. 今村誠; 渡邊圭輔; 増塩智宏; 渡部明洋;. 素性論理に基づく XML 文書ルール記述言語 DRDL とインターネット文書交換システムへの応用.情報処理学会論文誌. 2006, vol.47, no.3, p.751-764.
4. Jesús M. Almendros-Jiménez. Validation of XML Documents with SWRL. Multidisciplinary Research and Practice for Information Systems. 2012, vol.7465, p. 44-57.
5. Tim Berners-Lee; James Hendler; Ora Lassila. The Semantic Web. Scientific American. 2001, p. 29-37.
6. Nigel Shadbolt; Wendy Hall; Tim Berners-Lee. The Semantic Web Revisited. IEEE Intelligent Systems Journal, 2006, p. 96-101.
7. 溝口理一郎. オントロジー工学. オーム社, 2005.
8. Shinji Norimatsu; Kenji Murakami; Hiroshi Takahashi. “Validation of XML Document Content Using Ontology”, Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition. Kuala Lumpur, Malaysia, 2014, p. 152-158.
9. 乗松真二, 村上研二. OWL と SWRL を用いた不動産登記申請妥当性検証システムの提案. 情報システム学会誌. 2014, vol.10, no.1, p. 1-13.
10. 神崎正英. セマンティック・ウェブのための RDF/OWL 入門.森北出版, 2005.
11. 斎藤信男, 萩野達也. セマンティック Web 入門.オーム社, 2004.
12. “RDF 1.1 Primer”, W3C Working Group Note, 2014.  
<http://www.w3.org/TR/rdf11-primer/>.
13. トム ヒース, クリスチャン バイツァー, 武田英明 (訳). Linked Data: Web をグローバルなデータ空間にする仕組み. 近代科学社, 2013.
14. “RDF Schema 1.1”, W3C Recommendation. 2014.  
<http://www.w3.org/TR/rdf-schema/>.
15. ディーン アルマン, ジェームス ヘンドラー , 萩野達也 (監修), セマンティック Web 委員会 (翻訳). 実践セマンティック Web—RDF/RDFS/OWL によるオントロジー設計ガイド.ジャストシステム, 2010.
16. Dean Allemang; James Hendler. Semantic Web for the Working Ontologist, Second

- Edition: Effective Modeling in RDFS and OWL. Morgan Kaufmann, 2011.
17. OWL 2 Web Ontology Language Document Overview (Second Edition), W3C OWL Working Group. 2012. <http://www.w3.org/TR/owl2-overview/>.
  18. OWL 2 Web Ontology Language Primer (Second Edition), W3C Recommendation. 2012. <http://www.w3.org/TR/owl2-primer/>.
  19. 兼岩憲. 記述論理と Web オントロジー言語. オーム社, 2009.
  20. RuleML. RuleML Inc. <http://wiki.ruleml.org/>.
  21. Ian Horrocks; Peter F. Patel-Schneider; Harold Boley; Said Tabet; Benjamin Grosz; Mike Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. 2004. <http://www.w3.org/Submission/SWRL/>.
  22. 來村徳信, 人工知能学会. オントロジーの普及と応用. オーム社, 2012.
  23. 溝口理一郎, 人工知能学会. オントロジー工学の理論と実践. オーム社, 2012.
  24. Mike Uschold; Michael Gruninger. Ontologies: Principle, Methods and Applications. Knowledge Engineering Review. 1996.
  25. Michael Gruninger; M. Fox. Methodology for the Design and Evaluation of Ontologies. Proceedings of Int. Joint Conf. AI 1995, Workshop on Basic Ontological Issues in Knowledge Sharing.
  26. 石川誠一, 久保成毅, 古崎晃司, 來村徳信, 溝口理一郎. タスク・ドメインロールに基づくオントロジー構築ガイドシステムの設計と開発. 人工知能学会論文誌, 2002, vol.17, no.5, p. 585-597.
  27. Natalya F. Noy; Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.
  28. W3C. Semantic Web Best Practices and Deployment Working Group. <http://www.w3.org/2001/sw/BestPractices/>.
  29. Rinke Hoekstra. Ontology Representation - Design Patterns and Ontologies that Make Sense. IOS Press, 2009.
  30. 古崎晃司, 來村徳信, 池田満, 溝口理一郎. 「ロール」および「関係」に関する基礎的考察に基づくオントロジー記述環境の開発. 人工知能学会論文誌. 2002, vol.17, no.3, p.196-208.
  31. 吉野一. 法律人工知能—法的知識の解明と法的推論の実現. 創成社, 2005.
  32. 山口高平, 樽松理樹. 法律オントロジー. 人工知能学会誌. 1998, vol.13, no.2, p.189-196.
  33. ESTRELLA. Deliverable 1.4 OWL Ontology of Basic Legal Concepts(LKIF-Core). 2007.
-



<http://www.estrellaproject.org/doc/D1.4-OWL-Ontology-of-Basic-Legal-Concepts.pdf>

.

34. 樽松理樹, 山口高平. 法律知識の体系的定義としての法律オントロジー. 人工知能学会誌. 2004, vol.19, no.2, p.144-150.
35. W3C Recommendation. W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures. 2012. <http://www.w3.org/TR/xmlschema11-1/>.
36. W3C Recommendation. W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. 2012. <http://www.w3.org/TR/xmlschema11-2/>.
37. James Clark; MURATA Makoto. RELAX NG Specification. 2001.  
<https://www.oasis-open.org/committees/relax-ng/spec-20011203.html>.
38. Matthew Horridge; Peter F. Patel-Schneider. OWL 2 Web Ontology Language Manchester Syntax (Second Edition). 2012.  
<http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/>.
39. JXML2OWL Project. <http://jxml2owl.projects.semwebcentral.org/>.
40. Toni Rodrigues; Pedro Rosa; Jorge Cardoso. MAPPING XML TO EXISTING OWL ONTOLOGIES. International Conference WWW/Internet. 2006, p.72-77.

## 謝辞

本研究を遂行するに当たり，終始熱心なご指導を賜りました愛媛大学名誉教授 村上研二先生に心より感謝申し上げます．また本研究を遂行するに当たり，終始熱心なご指導を賜りました愛媛大学大学院理工学研究科 高橋寛教授に心より感謝申し上げます．本研究の過程で多くの貴重なご教示を頂きました愛媛大学大学院理工学研究科 藤田欣裕教授，樋上喜信准教授に厚く御礼を申し上げます．

研究テーマとした登記に関しては司法書士の先生方に貴重なご教示を賜りました．特に元日本司法書士会連合会会長 佐藤純通先生の前向きで，かつ示唆に富んだ貴重なお話を多数頂戴したことは，登記の知的支援を研究するきっかけとなりました．心より感謝申し上げます．

株式会社リーガルの皆様には社会人として研究することにご理解とご配慮を賜りました．まず故山岡昭子 前社長には入学を快諾して頂きました．そして大塚至正 社長には多忙な業務の中で研究を行うことへのご理解とご支援を頂きました．重松学 法務部長には登記の知識をわかりやすくご教示頂きました．津田孝昭氏には英語に関する貴重なアドバイスを頂きました．そして多忙な仕事の中，テクノプラザ分室のメンバーにご配慮を頂きながら研究を継続できたことは私にとって一生の思い出となりました．皆様に心より感謝申し上げます．

最後に，陰ながら見守ってくれる両親，そして最高の笑顔と愛妻弁当で応援してくれる妻 良美へ感謝の言葉を贈りたいと思います．

## 研究業績一覧

### 学術論文（査読あり）

1. 乗松真二, 村上研二. OWL と SWRL を用いた不動産登記申請妥当性検証システムの提案. 情報システム学会誌. 2014, vol.10, no.1, p. 1-13.

### 国際会議論文（査読あり）

2. Shinji Norimatsu; Kenji Murakami; Hiroshi Takahashi. Validation of XML Document Content Using Ontology, Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition. Kuala Lumpur, Malaysia. 2014, p. 152-158.

### 国内学会発表

3. 乗松真二, 村上 研二. “不動産登記の妥当性検証のためのオントロジの構築”. 第 28 回人工知能学会全国大会, 2014.

### 研究会発表

4. 乗松真二, 村上研二. “OWL と SWRL を用いた不動産登記申請の妥当性検証システムの提案”. 人工知能学会 第 31 回セマンティックウェブとオントロジー研究会, 2013.