

[論文]

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

乗松 真二[†]、村上 研二[‡]

要旨

本論文は、不動産登記申請の妥当性検証を行うシステムを提案する。本システムは、OWL と SWRL を用いて、申請データを記述するための申請オントロジと、検証ルールを記述した検証オントロジの二つのオントロジを構築している。このオントロジを利用して、申請データに対して、検証システムが推論を行うことで妥当性を判定する。検証システムの推論は、OWL と SWRL の推論に加えて、OWL, SWRL では記述できない拡張ルールの適用を行う。本論文では、これらオントロジの構築手順と検証システムの詳細、それらの特徴について述べる。

Abstract

This paper proposes a validation system of a real property registration application. The system is constructed from two ontologies, the application ontology for describing application data, and the validation ontology for describing validation rules, using OWL and SWRL. The validation system judges validity by reasoning to application data using these ontologies. In addition to reasoning of OWL and SWRL, the system applies extended rules which cannot be described by OWL and SWRL. This paper also gives the construction procedure of these ontologies, details of the validation system, and its significant properties of the system.

1. はじめに

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。

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

Proposal of a Validation System of Real Property Registration Application Using OWL and SWRL

Shinji Norimatsu, Kenji Murakami

[†]株式会社リーガル、[‡]愛媛大学大学院

[†]Legal Co., Ltd.; [‡]Graduate School, Ehime University

[論文] 2013年11月29日受付

2014年 7月 7日受理

© 情報システム学会

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オンラインによる申請も可能となり、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、OWL (Web Ontology Language)^{[1][2]}と SWRL (Semantic Web Rule Language)^[3]を用いた不動産登記申請の妥当性検証システムを提案する。まず登記の知識や検証ルールに関するドメインオントロジ^[4]を定義して W3C (World Wide Web Consortium)で勧告されている OWL, SWRL, 拡張ルールを用いて記述する。そしてそれらの推論を実行することにより申請内容の妥当性を検証するシステムを提案し、そのプロトタイプシステムを構築する。

本論文の構成は以下の通りである。まず2章で関連研究について述べた後、3章でシステム概要として不動産登記例と提案のシステムを概観し、4章でシステムの詳細を述べる。5章で本システムの評価と考察について述べ、最後に6章でまとめと今後の課題を述べる。

2. 関連研究

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^[5]。ここでは法律オントロジの構築^[6]に関する研究が行われているが、法律知識の構造解明が中心課題とされ、オントロジ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、

登記に関する具体的なオントロジ構築は示されていない。そこで本論文では不動産登記法の概念を出発点とした不動産登記申請の妥当性検証に特化した軽量ドメインオントロジを構築し、その推論により妥当性検証を行うシステムを提案している。

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^{[7][8]}。ここで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上位概念語彙定義と設計原理を示しているが、具体的な法律の語彙定義は行っていない。本論文では日本の不動産登記申請のオントロジを作成することで、その検証に必要な語彙や概念定義を行っている。一般に法律概念は各国で異なるものが多く、日本の法律制度にそのままあてはめることは困難であるが、本論文のように具体的なオントロジを作成することにより、その関係性を具体的に考えることが可能となる。

電子申請システムへ応用可能な申請データ検証に関する研究がある^[9]。ここでは XML 文書に対して内容間制約を記述できる言語を作成し制約チェックを行う手法が提案され XML スキーマ等ではチェックできない高度な XML 文書内容チェックが可能となっている。しかしこれら手法は XML データ間の制約ルールが基本となっており、データ内容により制約が変化するような場合にはルール記述量が増大する。また詳細なルールの作成には法律知識が必要となることから法律専門家との共同作業が不可欠であるが、法律概念を明確にするものではないことから法律専門家との共同作業による制約ルール記述は困難と思われる。本論文の手法はオントロジを基本として法律概念を明確にして検証ルールを記述することにより法律専門家との共同作業を可能とし、さらに OWL のもつ概念間の継承や制約等の推論能力を利用することで記述量の削減を可能としている。一方、本論文の手法は XML データ構造を直接検証するものではないことから、これらの手法を用いた XML データの構造検査と本論文のオントロジによる

内容検証を組み合わせて利用することが有用である。

3. システム概要

3.1. 不動産登記の例

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。

3.2. 情報システム概要

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。

本システムの概要を図1に示す。

本システムは、OWLファイルとして保存されている申請オントロジ、検証オントロジ、申請データを入力として検証システムを実行し、推論により申請データの妥当性を判定して結果を表示するとともに、結果オントロジとしてOWLファイルを出力する。

ここで、申請オントロジとは、登記申請に関する知識（クラス、プロパティ等）をOWL、SWRLを用いて記述、構築したものである。申請データは、検証対象となる申請情報、添付情報等の実データであり、個体と個体間の関係等を申請オントロジのプロパティを用いてOWLで記述したものである。

検証システムは、申請オントロジと申請データに加えて検証オントロジを読み込む。検証オントロジは、妥当性検証項目とその検証ルールで構成され、OWL、SWRL、拡張ルールで記述

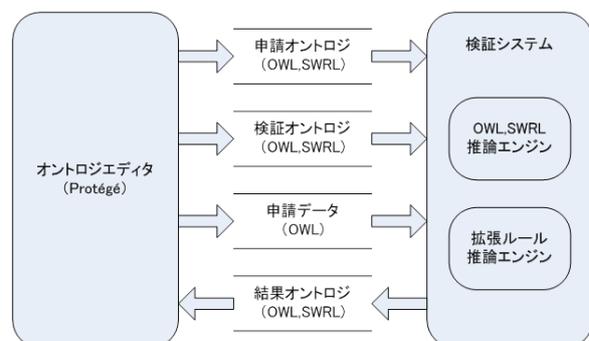


図1 システム概要

するものである。検証システムは、OWL, SWRLの推論に加え、拡張ルールの推論を行い、与えられた申請データの妥当性を判定して結果を表示するとともに、検証結果を結果オントロジとして出力する。

本システムの検証システムの実装には Java 言語を用い、OWL, SWRLの解釈には OWLAPI を利用している。また、OWL, SWRL の推論を行う Reasoner として、Pellet を用いた。OWL ファイルの表示、編集には、オントロジエディタとして Protégé を利用している。

4. システム詳細

OWL を用いてオントロジを作成する利点の1つとして推論ができることがある。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^{[10][11]}。

本システムの特徴は、この性質を利用し、個体と個体間のプロパティを申請データとして与えて推論を行い、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を構築する必要がある。本システムでは、与えるデータを記述するための申請オントロジと、検証項目が妥当であることを表すクラスやそれを導出する推論規則にあたる検証ルールを記述する検証オントロジに分けて構成する。

以下にこれらのオントロジ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。

4.1. 申請オントロジと申請データ

本システムの土台として申請オントロジを構築する。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, SWRL を用いて記述したものである。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で定義されるクラス等が導出されることになる。ただし、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で定義し、申請オントロジには含めない。

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	平成〇年△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持つ. 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^[12]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

上記例のイメージを図2に示す. ここで, **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.

なお, 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. また, 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.

その他, 本システムの申請オントロジの例を図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)

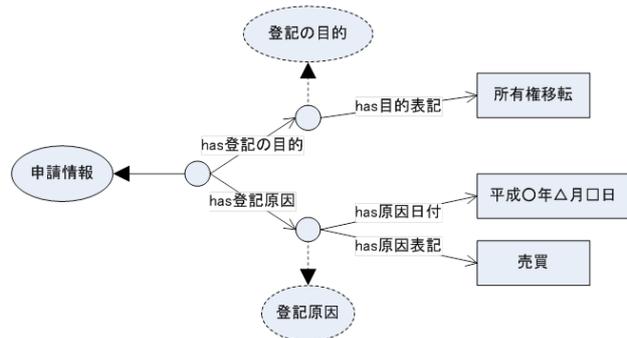


図2 申請データ例

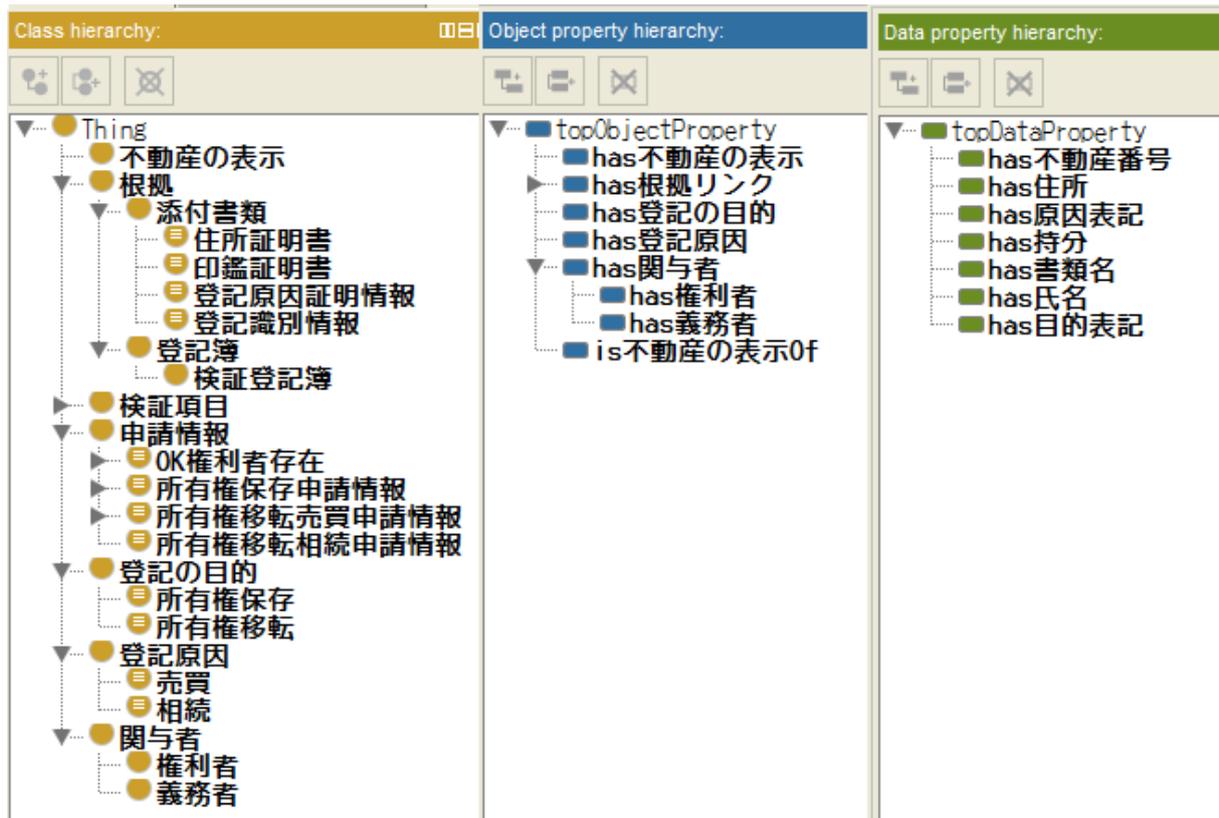


図3 申請オントロジの例

4.2. 検証オントロジ

検証オントロジは、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。

本システムでは、与えられた個体 X が検証項目 Y に対して妥当かどうかは、検証項目 Y が妥当であることを表すクラスを YC とすると、X が YC のインスタンスであるとき、妥当と判定する。

こ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。)

一方、検証項目 Y が妥当であることを確認する検証処理は、

IF 前提条件を満たす

THEN 検証項目は妥当である (YC)

という検証ルールとして考えることができる。ここで、帰結にあたるものは、本システムにおいては、検証項目 Y が妥当であることを表すクラス YC である。検証ルールの具体的な記述方法は後述するが、何れの記述方法においても検証ルールの考え方は同じである。

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持つ。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で定義する。

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。

4.2.1. 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。

OK 申請情報 \Leftrightarrow OK 所有権保存申請情報
 \vee OK 所有権移転売買申請情報 ...

OK 所有権保存申請情報 \Leftrightarrow 所有権保存申請
情報 \wedge OK 全不動産の表示 \wedge OK 全権
利者 \wedge ...

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

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 ...

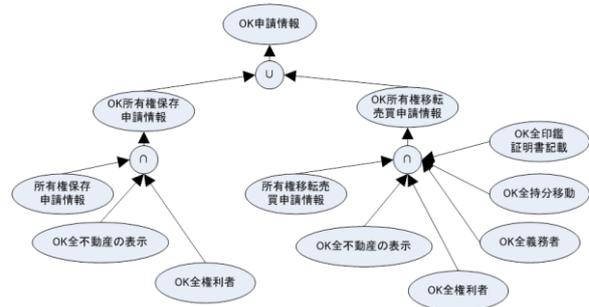


図 4 OK 申請情報の定義例

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 に示す。

4.2.2. 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:
    登記の目的 and
    (has 目的表記 value "所有権移転")
Class: 売買 EquivalentTo:
    登記原因 and
    (has 原因表記 value "売買")
```

これを用いて、検証パターンである所有権移転売買申請情報クラスは、登記の目的が所有権移転、登記原因が売買である申請情報の派生クラスとして、以下のように定義できる。

```
Class: 所有権移転売買申請情報
EquivalentTo:
    申請情報 and
    (has 登記の目的 some 所有権移転)
    and (has 登記原因 some 売買)
```

図2の例に対し、上記定義を与えた状態での検証パターンの推論のイメージを図5に示す。ここで、クラス(楕円)間の矢印は、クラスの継承関係を表す。

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

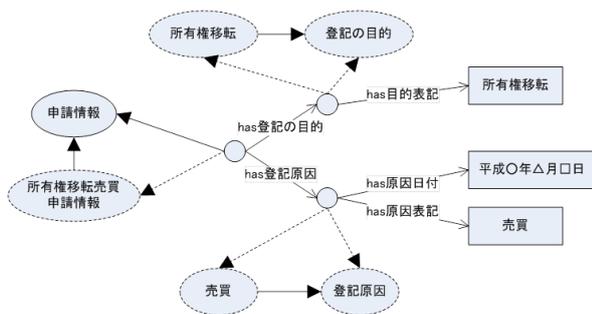


図5 検証パターンの決定

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。

4.2.3. 記載内容等に関する検証規則の記述

個々の記載内容に関する妥当性検証のルールも

```
IF 前提条件を満たす
THEN 検証項目は妥当である
```

の形で記述し、推論を行い、最終的に妥当であることを表すクラスを導出する。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)

また、OWL, SWRL の双方で記述可能なルールは、OWL で記述する。以下にそ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

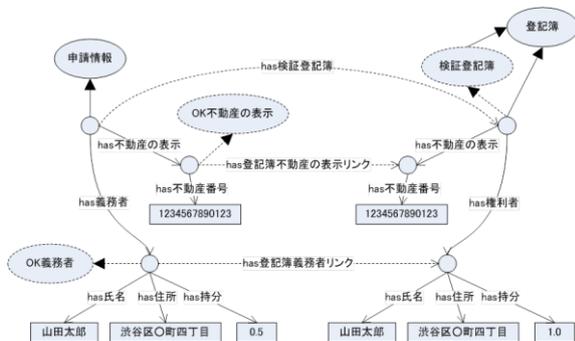


図 6 不動産の表示と義務者の検証例

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 6 に示す。

(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) [13]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ルール的前提条件は“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 全持分移動の検証ルール的前提条件は“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、先の妥当性検証ルー

ルと同じ考え方であり、拡張ルールに関する推論 (Java 言語によるルールの適用) 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、拡張ルール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。

拡張ルールの実行、実装について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。

先の OK 全不動産の表示を拡張ルールにより推論する例を図 7 に示す。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。

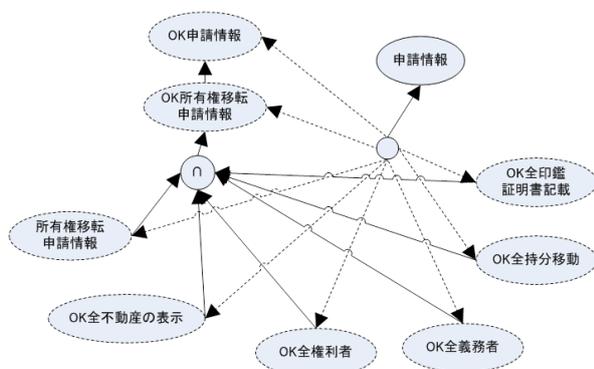


図 7 拡張ルール推論の例

4.3. 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. 申請オントロジ, 検証オントロジ, 申請データを読み込んだ状態のオントロジを O とする.
2. O に対し, OWL, SWRL の推論を実行し, その結果のオントロジを OR とする.
3. OR に対し, 全ての拡張ルールを適用し, その結果を OE とする.
4. $OR \neq OE$ のとき, 新たな知識の追加があったので, 2に戻る.
5. $OR == OE$ のとき, 新たな知識追加がないので, OE を OWL ファイルとして保存して終了. このとき, 申請情報のインスタンスが“OK 申請情報”クラスに属する場合, “妥当な申請情報”と判定して表示する.

ここで申請データは, 申請情報を構成する個体と申請オントロジで定義するオブジェクトプロパティ, データプロパティで記述するが, 本プロトタイプ実装においては, 検証システムが検証対象である申請情報の個体を特定するために, 申請情報の個体が申請情報クラスのインスタンスであることを宣言することとしている.

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 8 に示す. 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

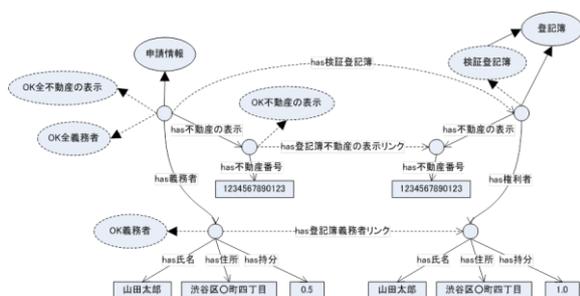


図 8 検証結果の例

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.

5. 評価と考察

5.1. 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, 申請オントロジと検証オントロジ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. また, 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をオントロジエディタである 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。

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持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 等のオントロジエディタ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。

なお、本論文ではサンプル申請データの入力にオントロジエディタである Protégé を用いて申請データの作成を行ったが、法務省や民間ベンダが提供する登記申請データ作成ソフトウェアを用い、添付ファイルを含めて申請情報が電子化されている場合は、申請内容である XML ファイル等と申請オントロジ語彙の変換プログラムを作成することで本システムの申請データへ機械的に変換可能であることから、利用者は妥当性検証用の申請データ作成の負担無く申請内容の妥当性を検証できるようになる。

5.2. 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構築を行った段階であり、今後さらに本オントロジ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

系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持っておらず、法律専門家も高度な情報技術知識を持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を構築できると考えられる。

6. むすび

本論文では、不動産登記申請の妥当性検証を OWL, SWRL と拡張ルールの推論を用いて行うシステムの提案を行った。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。

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、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。また、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できると、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が期待できる。

参考文献

- [1] W3C, <http://www.w3.org/TR/owl-overview/>, 2012-12-11.
- [2] 神崎正英, “セマンティック・ウェブのための RDF/OWL 入門”, 森北出版株式会社, 2005 年 1 月.
- [3] W3C, <http://www.w3.org/Submission/SWRL/>, 2004-05-21.
- [4] 溝口理一郎, “オントロジー工学”, 株式会社オーム社, 2005 年 1 月.
- [5] 吉野 一, “法律人工知能”, 株式会社創成社, 2000 年 2 月.
- [6] 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, 19, 2, pp. 144-150, 2004-03.
- [7] ESTRELLA, OWL Ontology of Basic Legal Concepts(LKIF-Core), <http://www.estrellaproject.org/doc/D1.4-OWL-Ontology-of-Basic-Legal-Concepts.pdf>, 2007-01-22.
- [8] Rinke Hoekstra, “ONTOLOGY REPRESENTATION -- Design Patterns and Ontologies that Make Sense,” IOS Press, 2009.
- [9] 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, 47, 3, pp. 751-764, 2006-03.
- [10] W3C OWL Web Ontology Language Guide, <http://www.w3.org/TR/owl-guide/>, 2004-02-10.
- [11] Dean Allemang James Hendler セマンティック Web 委員会訳, “実践セマンティック Web”, 株式会社ジャストシステム, 2010 年 3 月.
- [12] W3C OWL 2 Web Ontology Language Manchester Syntax (Second Edition), <http://www.w3.org/TR/owl2-manchester-syntax/>, 2012-12-11.
- [13] 兼岩憲, “記述論理と Web オントロジー言語”, 株式会社オーム社, 2009 年 8 月.