

# OWL を用いたロール概念表現に関する考察

## A Consideration of Role Representation using OWL

周 俊\*<sup>1</sup>  
Jun ZHOU

国府 裕子\*<sup>1</sup>  
Hiroko KOU

古崎 晃司\*<sup>1</sup>  
Kouji KOZAKI

溝口 理一郎\*<sup>1</sup>  
Riichiro MIZOGUCHI

\*<sup>1</sup> 大阪大学 産業科学研究所  
The Institute of Scientific and Industrial Research, Osaka University

OWL(Web Ontology Language) is one of the standard languages for ontology representation. Although many applications using OWL have been developed and expected, few of them provide a higher-level framework for conceptualization of the target world supported by fundamental investigation, which would cause to decrease semantic interoperability of ontologies because developers need to devise idiosyncratic patterns for building their own ontologies for themselves and such patterns will lack compatibility with others. In this article, we focus on roles as one of the common and typical semantic primitives in ontology development, and investigate representation model for dealing with characteristics of roles in OWL justified by fundamental investigation. It contributes to increasing semantic interoperability of roles by providing an infrastructure for role representation.

### 1. はじめに

近年、次世代の Web として期待される Semantic Web に注目が集まっている。Semantic Web を実現するために基盤技術はオントロジーに基づいて Web 上のリソースにメタデータを付与することにより、膨大な情報が存在する Web を効率よく情報を収集・検索・活用を可能にした点にある。オントロジーは「データに(計算機処理可能な)意味を与える」という点で Semantic Web 理念の根幹に関わっている。Semantic Web で用いられるオントロジー記述言語の一つに OWL(Web Ontology Language)がある。OWL は W3C によって Semantic Web におけるオントロジー記述言語の標準として定められ、Web 情報に内在するデータ(概念)やデータ間の関係を記述するための語彙や規約を定義するために使用される。OWL で定義したオントロジーに基づいて、様々な領域の知識を記述することで、プログラムが自動的に知識を処理し、知識の相互運用が可能となる。しかし OWL が提供しているのはオントロジーを記述するための基本要素のみで、ロール概念と言った概念レベルの高い要素を直接扱う枠組みを持っていない。そこで筆者らはオントロジー工学の理論的考察に基づき、ロール概念を OWL を用いて表現するモデルを提案してきた[砂川 06, Kozaki 07]。

本論文では、これらの考察を進め、オントロジーの利用目的に応じた様々な詳細度によるロール概念の OWL 表現について議論する。さらに、これらの OWL 表現を、ロール理論に基づいて開発されたオントロジー構築・利用環境「法造」[古崎 02]のロール概念表現と比較する。また、「法造」で構築したオントロジーを OWL 形式でエクスポートする機能の実装について述べる。この OWL エクスポート機能は、外部システムから法造の提供するオントロジーに関する操作機能の一部を利用可能にした API である HozoCore を用いて開発した。これにより法造が扱うロール概念を OWL を用いて記述し、高度な Semantics を規定する枠組みに基づいて取り扱うことが可能になる。

### 2. 法造におけるロール概念の扱い

法造は、ロール概念に関する理論[溝口 99, 古崎 02, Mizog

連絡先: 周俊, 大阪大学産業科学研究所 能システム分野,  
茨木市美徳ヶ丘 8-1, Tel:06-6879-8416, Fax:06-6879-2123, shuu@ei.sanken.osaka-u.ac.jp

uchi 07]を中心としたオントロジーの基礎理論に基づいて設計・開発がなされたオントロジーを構築ツールである。ここでは、ロール概念の理論を簡単に述べる。

#### 2.1 ロール概念

本論文が扱う「ロール概念」とは、「状況に依存して変化する個物の呼ぶ名」、また「あるものが特定のコンテキストのもとで果たす役割を概念化したもの」などと定義され、用いられるものである。ロール概念の本質はコンテキストの依存性である。例えば、「妻としての役割(妻 role)や「看護婦としての役割(看護婦 role)」など、あるものが特定のコンテキストのもとで果たす役割を捉えて概念化したものである[溝口 99, 古崎 02]。さらに「妻」や「看護婦」など、基本概念が「ロール概念」で定義された役割を担った状態にあるインスタンスを「ロールホルダー」と呼ぶ。担い得るもの(Role Playable Thing または Potential Role Player)が所属するクラスに対する制約を「クラス制約」と呼ぶ。

一方、ロール概念とは異なり、他の概念へ依存せずに定義可能な概念、例えば、「人間」、「学校」など他の概念に依存せずに定義される概念を「基本概念」という。

図 1 においては、ロール概念は「コンテキストとなる概念を表すノードに関連付けられたスロット」として表現される。ロール概念を表すスロットは、コンテキストとなる概念が全体概念である場合は全体一部分(part-of)関係を表すリンク(p/o で示す)または属性(attribute-of)関係を表すリンク(a/o で示す)で関連付けられ、スロットの数を規定する個数制約(Cardinality)が数字で表

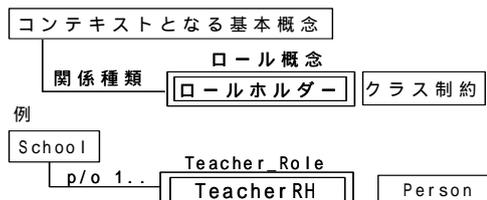


図 1 ロール概念の表現形式



図 2 ロール概念の階層表現

表 1 法造のオントロジー表現と OWL を用いた表現モデルの対応

概念	法造の表現	簡易モデル(a)	拡張モデル(b)	詳細モデル(c)
基本概念		owl:class	owl:class	owl:class
ルール概念		owl:ObjectProperty または owl:DatatypeProperty	owl:ObjectProperty または owl:DatatypeProperty	owl:class
Part-of関係 Attribute-of関係				owl:ObjectProperty
ルールホルダー		対応不可	owl:class (クラス制約として参照している概念の下位概念 (rdfs:subClassOf)として定義する)	owl:class
ルール概念がもつ スロット		対応不可	ルールホルダーの プロパティ	owl:class および owl:ObjectProperty (ルール概念がもつ プロパティ)
基本概念間の is-a 関係		rdfs:subClassOf	rdfs:subClassOf	rdfs:subClassOf
ルール概念間の is-a関係		rdfs:subPropertyOf	rdfs:subPropertyOf	rdfs:subPropertyOf rdfs:subClassOf

される。例えば図 1 のように「School というコンテキストの元で定義されるルール概念 Teacher\_Role の役割を、基本概念 Person が担い、School に対する部分である TeacherRH になる」という内容を表現している。

## 2.2 ルール概念の is-a 関係

ルール概念のコンテキストとなる概念に is-a 関係が成り立つとき、上位概念で定義されたルール概念は下位概念に継承される。例えば図 2 において、コンテキスト「School」において定義された「Teacher\_Role」は、「School」の下位概念である「University」に継承される。さらに、継承されたルール概念の定義を特殊化することで、ルール概念の下位概念を定義することができる。このとき、ルール概念間の is-a 関係が成り立つ。図 2 における「Professor\_Role」、「assistantProfessor\_Role」は、「Teacher\_Role」を継承・特殊化したもので「Professor\_Role is-a Teacher\_Role」と「assistantProfessor\_Role is-a Teacher\_Role」の関係が成り立つ。

## 3. OWL を用いたルール概念の表現モデル

法造が提供するルール概念に関する理論的枠組みと、OWL が提供するオントロジー記述の枠組みとの間には概念的なギャップが存在する。そのため、法造で構築したオントロジーの概念定義を、OWL で表現する際には、ルール概念の性質をどこまで正確に表現するかによって、様々な詳細度の表現モデルを考えることができる。本研究では、ルール概念を OWL 形式で表現する方法について、以下の 3 通りモデルを検討した。

- (1) OWL のオントロジー表現に合わせてルール概念の枠組みに制限を加え、ルール概念をプロパティとして表した簡易表現(簡易モデル(a)).
- (2) ルールホルダーをクラスとして扱うことなどにより簡易モデル(a)を拡張し、OWL を用いてルール概念の性質をより詳細に扱えるようにした拡張簡易表現(拡張モデル(b)).
- (3) 法造の提供する枠組みを、OWL と SWRL(Semantic Web Rule Language)を用いて忠実に反映した詳細表現(詳細モデル(c)).

表 1 は法造におけるオントロジーの表現と、上述の各表現モデルの対応をまとめたものである。3 つの表現モデルの相違点を比較することにより、法造の提供するオントロジー定義の枠組みの特徴がより明らかになる。また、法造で構築したオントロジーを OWL 形式にエクスポートし、Semantic Web アプリケーション開発に利用するときの指針を与えることができる。次節より、各表現モデルの詳細について述べる。なお紙面都合上、本論文では詳細モデル(c)の説明は割愛する。詳細は参考文献 [砂川 06, Kozaki 07]を参照いただきたい。

### 3.1 簡易モデル(a)

簡易モデル(a)では、法造における基本概念およびルール概念(基本概念が持つスロット)が持つ定義内容の一部が表現される。

#### (1) 基本概念の OWL クラスによる表現

このモデルでは、法造の基本概念は OWL のクラスとして owl:Class 要素を用いて表現され、基本概念間の is-a(上位-下位)関係は rdfs:subClassOf によって表される。例えば、図 2 の「School」と「University」の間の is-a 関係は <owl:Class rdf:ID="University"> の定義中に <rdfs:subClassOf rdf:resource="#School"/> と記述される。

次に、基本概念が持つ part-of 関係, attribute-of 関係およびそれらの関係に伴うルール概念の OWL 表現について説明する。法造では、基本概念の定義内容は基本概念を表すノードが持つスロット(図1)により表現される。

ここで図 2 の例を用い、ルール概念を中心にして、簡易モデル(a)による表現方法について説明する。

#### (2) ルール概念の OWL プロパティによる表現

##### ルール概念の表現:

ルール概念はOWLのプロパティとして表現される。OWLの基本的なプロパティには、個体同士の関係を記述する個体値型プロパティowl:ObjectPropertyと個体をデータ値と関係づけ

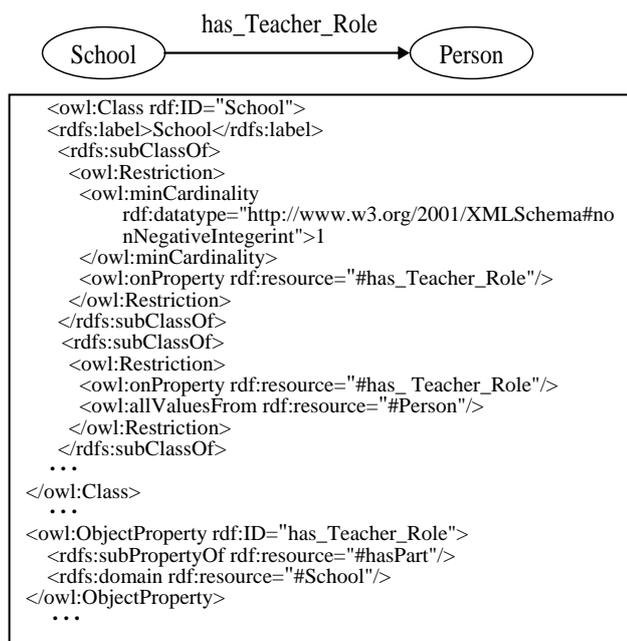


図3 簡易モデル(a)による「Teacher\_Role」の定義

るデータ値型プロパティowl:DatatypePropertyがある。これらの区別は、ロール概念のクラス制約が通常概念であるかDatatypeであるかの違いにより判断する(詳細は後述する)。例えば、図2におけるロール概念「Teacher\_Role」のクラス制約はPersonという通常概念であるので、この簡易モデル(a)では<owl:ObjectProperty rdf:ID="has\_Teacher\_Role">と表現される。なお、ロール概念をOWLのプロパティ(関係)で表す場合、関係としての名前とするためにロール概念名の前に「has\_」を追加する。

**コンテキストの表現:**

ロール概念が依存するコンテキストに相当する全体概念はロール概念を表すOWLプロパティの domain として記述する。図2において、「Teacher\_Role」は「School」をコンテキストとして定義されているので、「has\_Teacher\_Role」の domain は「School」となり、<owl:ObjectProperty rdf:ID="has\_Teacher\_Role">の定義中に<rdfs:domain rdf:resource="#School"/>と記述される。なお、同じ名前のロール概念が複数あり、各ロール概念の間に is-a 関係がない場合、domain にロール概念のコンテキストになるクラスを owl:unionOf を用いて列挙する。

**part-of/attribute-of 関係の表現:**

ロール概念とコンテキストの関係は、ロール概念を表すプロパティの上位プロパティとして表される。part-of 関係を表す hasPart は、owl:ObjectProperty の上位プロパティとして定義される。図3において、「Teacher\_Role」と「School」の関係は part-of 関係であるので、<owl:ObjectProperty rdf:ID="has\_Teacher\_Role">は、part-of 関係を表すプロパティ hasPart のサブプロパティ<rdfs:subPropertyOf rdf:resource="#hasPart"/>として記述される。なお、ロール概念とコンテキストの関係が attribute-of 関係の場合は、attribute-of 関係を表す hasAttribute プロパティのサブプロパティとして定義される。

**基本概念がもつロール概念の表現**

ロール概念はコンテキストに依存して定義されるので、ロール概念と基本概念間の関係(法造ではスロットで表現される)の

定義は、コンテキストとなる OWL クラスのプロパティ制約 owl:Restriction を用いて表現する。owl:Restriction の制約対象(owl:onProperty)で対象とするロール概念に対応するプロパティを指定し、クラス制約は owl:allValuesFrom として記述される。図2における「School」と「Teacher\_Role」の関係は、<owl:Class rdf:ID="School">の定義中の<owl:Restriction>において<owl:onProperty rdf:resource="#has\_Teacher\_Role"/>および<owl:allValuesFrom rdf:resource="#Person"/>として記述される。

さらに、関係の数に関する制約は、owl:maxCardinality, owl:minCardinality, owl:cardinality 三種類を用いて記述する。図2の例における「School」の「has\_Teacher\_Role」の最小値は1であるので(法造では"1."と示される)、<owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:minCardinality>と記述される。

図3に図2の例を簡易モデル(a)で記述したものを示す。

**データタイプの扱い**

ロール概念のクラス制約が Datatype(整数(integer), 文字列(string)...)などの場合がある。例えば、表1の「Person」基本概念のロール概念「Name」のクラス制約は「string」である。このときの OWL では owl:allValuesFrom の対象として Datatype を指定することができないので、ロール概念を表すプロパティの Range として記述する。上述の例では、<owl:DatatypeProperty rdf:about="#has\_Name">の定義の中に<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>と記述される。

**(3) ロール概念の is-a の扱い**

このモデルでは、法造のロール概念は OWL プロパティで表されるので、ロール概念間の is-a 関係は、下位ロール概念と上位ロール概念の間の rdfs:subPropertyOf 関係によって記述する。2.2 節で述べたように、ロール概念は常に依存するコンテキストとなる概念に付随する形で定義されるので、ロール概念の is-a 関係による階層化はコンテキストとなる概念の is-a 階層に沿って行われる[砂川 05]。例えば、図2において、「Professor\_Role」は「Teacher\_Role」の下位ロール概念であり、両者の関係は、<owl:ObjectProperty rdf:ID="has\_Professor\_Role">の定義中に<rdfs:subPropertyOf rdf:resource="#has\_Teacher\_Role"/>と記述される。

なお法造では、下位ロール概念と上位ロール概念の概念名が変わらないことを許容しているが、OWL では rdf:ID 属性の重複が認められないので、簡易モデル(a)では同じ概念名のロール概念は同一の OWL プロパティとして扱われる。

**(4) 簡易モデル(a)で表現できない内容**

簡易モデル(a)では、法造においてロール概念が持つスロットとして表されるロール概念の定義内容の詳細や、ロールホルダーを表現することができない。

**3.2 拡張モデル(b)<sup>2</sup>**

前節で述べた簡易モデル(a)のようにロール概念を OWL プロパティを用いて表現すると、法造がサポートしているロール概念の性質の一部しか表すことができない。この問題を解決するため、ロールホルダーをロール Player クラスのサブクラスとして表した表現モデルが拡張モデル(b)である。このモデルでは、法造でロール概念が持つスロットとして表される定義内容を、ロールホルダーを表す OWL クラスの定義内で表すことができる。図4

<sup>1</sup>OWLのプロパティは「関係」であるのに対しロール概念は「関係」ではない。

<sup>2</sup>これも簡易版であり、ロール理論的には不完全なモデルである。

に、このモデルを用いた OWL 表記における、ロールホルダーと各概念間の関係を示す。

(1) ロールホルダーの OWL クラスとしての表現

ロールホルダーとコンテキストの関係の表現:

このモデルでは、ロールホルダーはロール概念で定義される役割を担った状態にある基本概念のインスタンスであるため、ロール概念とコンテキストの関係は、ロールホルダーとコンテキストの間でも成立する。そのため、簡易モデル(a)で導入したロール概念およびロール概念とコンテキストの間関係の両方を表しているプロパティを対象 (owl:onProperty としての制約(owl:Restriction)に owl:allValuesFrom としてロールホルダーを加える。図 4 の例において、「School」は<owl:Class rdf:ID="School">として記述され、そのプロパティに関する制約<owl:Restriction>に<owl:onProperty rdf:resource="#has\_Teacher\_Role"/> <owl:allValuesFrom rdf:resource="#TeacherRH"/>と表される。

ロールホルダーとプレイヤーの関係の表現:

ロールホルダーはロール概念そのものが持つ性質(スロット)と、ロール概念によって参照されるプレイヤー(クラス制約)の性質を併せもった存在である。よって、このモデルではロールホルダーはクラス制約として参照している概念のサブクラスとして表す。図 4 の例においては、「TeacherRH」を「Person」のサブクラスとして<owl:Class rdf:ID="TeacherRH">の定義中に<rdfs:subClassOf rdf:resource="#Person"/>と記述される。

ロール概念が持つスロットの表現:

ロールホルダーがスロット(ロール概念)を持つ場合、それらのスロットが表すロール概念が定義されるコンテキストはそのロールホルダーとなる。よってそれらのロール概念に対応する OWL プロパティの domain はロールホルダーとなり、コンテキストとなるロールホルダーに対応する OWL クラスの定義にそれらプロパティに関する制約が追加される。図 4 の例において「TeacherRH」は<owl:Class rdf:ID="TeacherRH">で定義されその<owl:Restriction>の中に<owl:onProperty rdf:resource="#has\_Subject"/>, <owl:allValuesFrom rdf:resource="#Subject"/>などが記述される。

クラス制約からの継承・特殊化の表現:

ロールホルダーを表す OWL クラスにおけるプロパティの制約には、そのロールホルダーで定義された制約とプレイヤー

となるクラスから継承・特殊化した制約がある。これによりロール概念においてクラス制約から継承・特殊化したスロットを表現することができる。図 4 の例では「Person」が持つ「has\_Name」、「has\_Height」および「has\_Weight」プロパティは、ロールホルダー「TeacherRH」に継承されている。さらに「TeacherRH」においては「has\_Name」がクラス制約「Person」から継承されていることが明示化されると共に、「has\_Subject」プロパティが追加されていることが表されている。

(2) 拡張モデル(b)で表現できない内容

拡張モデル(b)ではロール概念、ロールホルダーの定義がより詳細に表現できる。しかし、ロール概念のインスタンスの生成問題やカウンティング問題など、ロール概念の性質を完全には表現できない。これらのすべてを表したい場合は、詳細モデル(c)[砂川 06, Kozaki 07]を用いる必要がある。

3.3 各表現モデルに基づく OWL エクスポート機能

現在、Web サイト(<http://www.hozo.jp>)上で公開している「法造」のオントロジーエディタ(ver.5.0)では、簡易モデル(a)に基づく OWL エクスポート機能のプロトタイプが実装されている。この出力結果は Protégé[Protégé]といったオントロジー構築ツールでの読み込みが可能であることは確認されている。また、オンラインでオントロジーの内容検証ツールである OWL Ontology Validator[Ontology Validator]での利用も可能であることも確認した。拡張モデル(b)および詳細モデル(c)による OWL エクスポート機能についても現在開発を進めており近日中に公開を予定している。

4. まとめ

本論文では、法造におけるオントロジーで扱うロール概念の性質を、OWL の標準形式で表現するモデルについて考察し、各モデルにおけるロール概念表現の特徴を比較した。そして、これらの考察に基づいて「法造」で構築したオントロジーの OWL のエクスポート機能を開発した。これにより、「法造」のロール概念の特徴を必要に応じて活かした Semantic Web アプリケーション開発を進めることができる。今後、「法造」の拡張を進めると共に、応用システム開発への適用を進めていきたい。

参考文献

[溝口 99] 溝口理一郎: オントロジー工学, オーム社, 2005.  
 [古崎 02] 古崎, 来村, 池田, 溝口: 「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, 人工知能学会誌, Vol.17, No.3, pp.196-208, 2002.  
 [砂川 05] 砂川, 古崎, 来村, 溝口: コンテキスト依存性に基づくロール概念組織化の枠組み, 人工知能学会論文誌, Vol.20, No.6, pp.461-472, 2005.  
 [砂川 06] 砂川, 古崎, 来村, 溝口: OWL と SWRL を用いたロール概念の取り扱いに関する一考察, 人工知能学会第 20 回全国大会(JSIAI2006), 3B4-2, 2006.  
 [OWL] <http://www.w3.org/TR/webont-req/>  
 [Protégé] <http://protege.stanford.edu/index.html>  
 [Ontology Validator] <http://www.mygrid.org.uk/OWL/Validator>  
 [Mizoguchi 07] Mizoguchi, R., et al., A Model of Roles within an Ontology Development Tool: Hozo, J. of Applied Ontology, Vol.2, No.2, pp.159-179. Sep.2007  
 [Kozaki 07] Kozaki K., et al., Role Representation Model Using OWL and SWRL, Proc. of 2nd Workshop on Roles and Relationships in Object Oriented Programming, Multiagent Systems, and Ontologies, Berlin, July 30-31, 2007

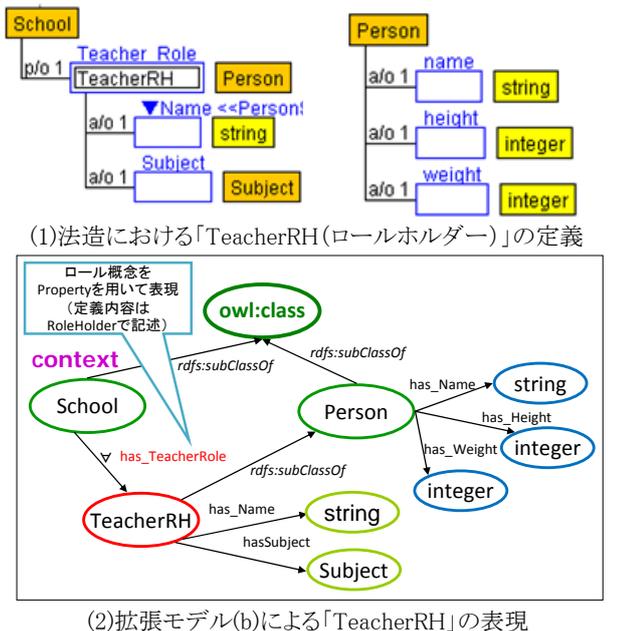


図 4 拡張モデル(b)によるロール概念の表記