

Semantic Web アプリケーションの開発に向けた オントロジー構築・利用環境「法造」の拡張 An Extension of Hozo for Development Semantic Web Applications

古崎 晃司*¹ 太田 衛*² 砂川 英一*¹ 來村 徳信*¹ 溝口 理一郎*¹
Kouji Kozaki Mamoru Ohta Eiichi Sunagawa Yoshinobu Kitamura Riichiro Mizoguchi

*¹ 大阪大学産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

*² 株式会社エネゲート
ENEGATE Co., Ltd.

Recently, the semantic web attracts many researchers, and ontology is placed among important technology to realize it. Building an ontology requires a clear understanding of what can be concepts with what relations to others. An ontology focuses on “concepts” themselves rather than “vocabulary”, and its design is not the problem of how to represent but that of identifying the inherent conceptual structure. We discuss mainly “role concept” and “relationship”, and consider how these ontologically important concepts should be treated in our environment. On the basis of the consideration we have designed and have developed an environment for building and using ontologies, named “Hozo”. This paper presents an outline of the functionality of Hozo. And we consider how to apply our system to use for developing semantic web applications.

1. はじめに

近年、オントロジーの応用分野として注目されている研究課題の1つに、Semantic Webがある。Semantic Webは、Webの創始者でもあるTim Berners-Leeにより提唱された次世代のWebのビジョンで、現状のWebコンテンツに計算機が理解可能な意味情報を付加することにより、Webの有用性を飛躍的に高めようとしている[W3C SW]。オントロジーは、この意味情報を表現する方法の1つとして位置付けられ、Semantic Web実現に向けた基礎技術の研究、標準化への取り組み、Semantic Webに用いるオントロジーを構築するためのツールの開発など、様々な研究が進められている。

このような背景のもと筆者らは、オントロジーの構築・利用を支援する計算機環境として、オントロジー構築・利用環境「法造」の開発を進めると共に、様々なプロジェクトにおいてオントロジーやオントロジーベースのアプリケーションの開発を行ってきた[古崎 02a,02b, 石川 02]。本論文では、「法造」をSemantic Webアプリケーション開発に向けた、法造の拡張について、Semantic Webの標準技術仕様であるOWLへの対応を中心に考察する。

2. オントロジー構築利用・環境「法造」

ここでは筆者らがこれまでに開発を進めてきた、オントロジー構築・利用環境「法造」の概要を述べる。法造の特徴は、全体概念・関係概念に関する理論およびロール概念に関する理論を中心とした、オントロジーの基礎理論に基づいて設計・開発がなされている点にある。以下、それらの理論の概要を述べる。

2.1 全体概念および関係概念

全体概念とは、あるものをその一部分をなす複数の概念から構成される全体としてとらえ概念化したものであり、関係概念とは、複数の概念間に成り立つ関係を概念化したものをいう[溝口 99, 古崎 02a]。例えば、「夫婦」という概念は全体概念に相当し、夫婦間の関係性に注目して概念化すると、関係概念「夫婦関

係」となる。法造では、全体概念および関係概念の対応関係を管理しながら記述する環境を提供する(図1)。

2.2 ロール概念

「ロール概念」とは、「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 02a]。それに対し「人間」、「男性」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。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。例えば「自転車」の部品である基本概念「車輪」のインスタンスが、「自転車」において「前輪としての役割」(前輪 role)で定義された役割を果たしているとき、その「車輪」のインスタンスはロールホルダー「前輪」となる。この際、ロール概念が定義する役割を担うことができる基本概念に関する制約を「クラス制約」と呼ぶ。

法造ではこのようなロール概念は図2のように表示される。図の例では、自転車というコンテキストの下、前輪 role および後輪 role という2つのロール概念が定義されている。図中の p/o は全体部分(part-of)関係を表し、その横にかかれた“1”という数

連絡先: 古崎晃司, 大阪大学産業科学研究所ナノバイオ知能システム分野, 茨木市美穂ヶ丘 8-1, Tel:06-6879-8416, Fax:06-6879-2123, kozaki@ei.sanken.osaka-u.ac.jp

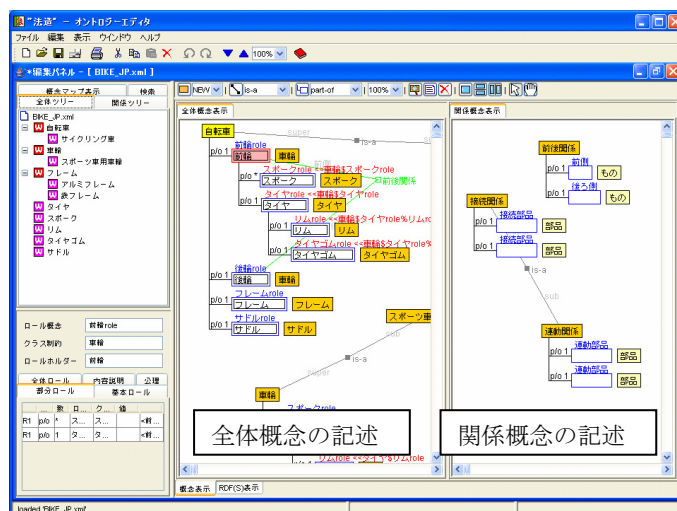
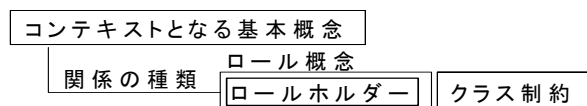
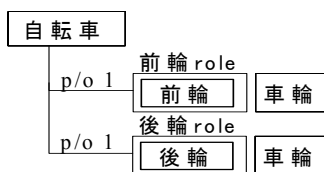


図1 法造のオントロジーエディタの画面例



a) ロール概念の表示形式



b) 「自転車」におけるロール概念の表示例

図2 法造におけるロール概念の表示例

字は、自転車において前輪 role, 後輪 role のそれぞれを果たす車輪が1つずつ存在するという数に関する制約を表す。

2.3 「法造」を用いたアプリケーション開発

法造を用いたアプリケーション開発の手法は、

- (1) 構築したオントロジーやモデルを、RDF(S)や OWL といった標準フォーマットへ変換し、それらのフォーマットに対応したツールで変換結果を読み込んで利用する方法
- (2) 外部システムから法造の提供する機能の一部を利用可能にした HozoAPI を用いる方法

があげられる。特に近年は、Semantic Web の標準技術として、OWL[OWL]を利用するアプリケーションが多く開発されており、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に対する要望が高まっている。そこで次章では、「法造」の OWL エクスポート機能に関する仕様を検討する。

3. 「法造」の OWL エクスポート仕様

法造が提供するロール概念および全体概念・関係概念に関する基礎理論の考察に基づくオントロジー構築の枠組みと、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。その為、法造で構築したオントロジーの概念定義を、そのまま OWL で表現する際には、その概念的なギャップを如何に埋めるかが問題となる。

ここでは、OWL 形式でのエクスポート仕様を検討するに辺り

- (1) OWL のオントロジー表現に合わせて法造が提供する枠組みに制限を加えた簡易表現によるエクスポート
- (2) 法造の提供する枠組みを、OWL を用いて可能な限り忠実に反映した詳細表現によるエクスポート

の2つのアプローチをとった。この2つの相違点を比較することによって、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と共に、法造を用いた Semantic Web アプリケーション開発時の指針を与えることができる。

2章で述べたように、法造の特徴は、「ロール概念」および「全体概念・関係概念」の扱いにある。以下、それぞれについて OWL 形式によるエクスポート時における扱いを検討する。

3.1 全体概念・関係概念の扱い

ここで注意すべき点は、法造における全体概念および関係概念は、共に OWL におけるクラス(owl:Class)に対応することである。すなわち、法造の関係概念は、OWL のプロパティとは対応しない。なぜなら、関係概念は複数の概念間の関係を捉えて概念化したものであるが、それは単なる関係性を表すだけではなく、その関係自身の性質を規定する定義も含んでいるからである。例えば、「夫婦関係」の関係概念として定義には、そ

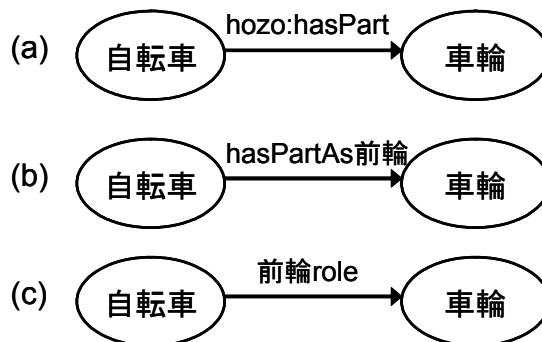


図3 簡易表現によるロール概念の OWL 表記

れが夫と妻の間の関係であるということだけではなく、その関係が何年続いているかといった属性も記述することができる。また関係概念は、2項関係のみならず3項以上関係も含まれる点も、OWL のプロパティとは異なる点である。さらに、すべての関係概念には、それらの関係に参加している概念を含む全体を概念化した全体概念を考えることができ¹、その対応を表現できる必要がある。

これらの考察により、OWL 形式でエクスポートする際には、OWL クラス(owl:Class)のサブクラスとして、全体概念クラス(hozo:WholenessConcept)および関係概念クラス(hozo:RelationalConcept)を定義し、法造で定義した全体概念および関係概念は、それぞれのサブクラスとして表す(hozo:は法造から OWL にエクスポートする為に定義したクラスであることを示す namespace)。この全体概念・関係概念の OWL エクスポート時の扱いは、前述の2つのアプローチに共通して利用される。

3.2 ロール概念の扱い

ここでは、図2の自転車の例において車輪が前輪 role を果たしロールホルダー前輪となる例を用いて、OWL 形式での表現を考察する。

(1) 簡易表現による OWL エクスポートにおける扱い

まず初めに、法造における part-of 関係を OWL のプロパティ(owl:ObjectProperty)のサブプロパティ hozo:hasPart プロパティとして定義することを考える。その際、この例における hozo:hasPart の定義域(domain)は、自転車クラスとなるが、値域(range)の候補としては、図2における車輪と前輪のいずれをとればよいか問題となる。しかし、法造におけるロールホルダー「前輪」は、あくまで車輪のインスタンスが前輪 role で定義される役割を担っている状態を表しており、クラスとしては認定されない[古崎 02a]。よって、rangeには車輪クラスをとることにする(図3-(a))。

しかし、それでは、この車輪が前輪 role を果たしていることが表すことができない。そこで、hozo:hasPart のサブプロパティとして、「前輪としての役割(前輪 role)を果たす部分をもつ」という意味を表す“hasPartAs 前輪”というプロパティを定義し、それを用いて図3-(b)のように表現する。ここで、“hasPartAs 前輪”の省略表現として、“前輪 role”をプロパティ名とすることも考えられる(図3-(c))。ただし、この表記における“前輪 role”はあくまでもラベルの省略表記であり、ロール概念の定義を正確に反映するこ

¹ すべての概念について全体概念・関係概念の両方として定義する必要はない。法造では、対応する全体概念・関係概念の両方が定義されたときのみ、その対応を管理している。

```

<owl:Class rdf:ID="hozo:WholenessConcept">
  <rdfs:label>WholenessConcept</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="hozo:RelationalConcept">
  <rdfs:label>RelationalConcept</rdfs:label>
</owl:Class>
...
<owl:Class rdf:ID="自転車">
  <rdfs:label>自転車</rdfs:label>
  <rdfs:subClassOf rdf:resource="hozo:WholenessConcept" />
  <rdfs:subClassOf>
    <owl:Restriction owl:cardinality="1" >
      <owl:onProperty rdf:resource="#前輪role"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction owl:cardinality="1" >
      <owl:onProperty rdf:resource="#後輪role"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
...
<owl:ObjectProperty rdf:ID="hozo:hasPart">
  <rdfs:domain rdf:resource="hozo:WholenessConcept" />
  <rdfs:range rdf:resource="hozo:WholenessConcept" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="前輪role">
  <rdfs:subPropertyOf rdf:resource="hozo:hasPart"/>
  <rdfs:domain rdf:resource="#自転車"/>
  <rdfs:range rdf:resource="#車輪"/>
</owl:ObjectProperty>
...

```

図4 簡易表現によるOWL出力のXML表記(一部)

とはできない。具体的には、この形式でエクスポートした際には、法造で記述された前輪 role に関する定義(例えば、操舵機能を持つ、フォークと接続されている、前照灯に電力を供給する、など)はすべて失われる。

図4に、簡易表現によるOWL形式でエクスポートした例のXML表記の一部を示す。3.1節で述べたように全体概念を表す `hozo:WholenessConcept` クラス、および関係概念を表す `hozo:RelationalConcept` クラスが定義され、「自転車」クラスは `hozo:WholenessConcept` のサブクラスとして定義されている。part-of 関係を表す `hozo:hasPart` は、`owl:ObjectProperty` のサブプロパティとして定義され、前輪 role はさらにそのサブプロパティとして定義されている。ただし、ここでの前輪 role は、前輪の役割を果たす部分を持ち(`hasPartAs` 前輪)の省略表記であることに注意が必要である。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(ここでは自転車)と、ロール概念で定義された役割を果たす基本概念のクラスに関する制約(ここでは車輪)は、それぞれ domain および range を用いて記述され、part-of 関係の数に関する制約は、`owl:Cardinality` を用いて記述される。

法造のオントロジーエディタ ver.4.0.5 におけるOWLエクスポートは、この簡易表現を実装している。

(2) 詳細表現によるOWLエクスポートにおける扱い

前節で述べたように、簡易表現によるOWLエクスポートにおいてはロール概念を `owl:ObjectProperty` を用いて定義すると、法造がサポートしているロール概念の特徴を正しく反映することはできない。この問題を解決し、法造で記述されたロール概念

の定義内容をできるだけ忠実にOWL形式で表現するには、ロール概念をクラスとして定義し基本概念との関係を明確にする必要がある。この点に、法造におけるロール概念と、RDF(S)やOWLなどオントロジー表現言語におけるプロパティの概念的な相違点が現れている。

ここで、ロール概念はコンテキスト依存性など、基本概念と異なるセマンティクスを持つので、OWL表現の際にも基本概念を表すクラス(3.1節で述べた `hozo:WholenessConcept` クラス)とは別に、ロール概念を表すクラス(`hozo:RoleConcept`)を定義する。さらに、ロール概念のコンテキスト依存性を表すプロパティとして `hozo:contextOf`、ロール概念で定義される役割を担う基本概念との関係を表すプロパティとして `hozo:playRole` を定義する。またこれらのプロパティとの反対関係(`owl:inverseOf`)として、`hozo:hasRole` および `hozo:holderOf` というプロパティを定義することができる。これらの法造のOWLエクスポートの為に拡張した定義を用いると、図2の例は図5の様に表される。

これらの定義に基づき、詳細表現によるOWLエクスポートを行った結果のXML表記を図6に示す。ロール概念を表す `hozo:RoleConcept` クラスは、必須(`Cardinality=1`)のプロパティとして `hozo:contextOf` および `hozo:holderOf` をもつ `owl:Class` のサブクラスとして定義されている。前輪 role は、そのサブクラスにおいてさらに、ロール概念が依存するコンテキストを表す `hozo:contextOf` の値域が自転車、ロール概念を担うことができるクラスに関する制約を表す `hozo:holderOf` の値域が車輪となるように、`owl:allValuesFrom` を用いた制約を加えることで定義される。自転車が持つプロパティ `hasPartAs` 前輪 および `has` 前輪 role は、それぞれ、part-of 関係を表す `hozo:hasPart` およびそのコンテキスト下で定義されるロール概念との関係を表す `hozo:hasRole` のサブプロパティにおいて、値域を「前輪となる車輪」および「前輪 role」に制限したものである。「前輪となる車輪」は、前輪 role との間に `playRole` 関係(`hozo:playRole` プロパティ)を持つ車輪クラスとして定義される。なお、前輪は車輪クラスのインスタンスが前輪 role を担った状態として定義されるロールホルダーであるので、車輪クラスと前輪 role の両方の定義内容を合わせたものとなる。しかし、`rdfs:subClassOf` を用いて車輪クラスと前輪 role を多重継承したものとして前輪を記述することは、`subClassOf(is-a 関係)` のセマンティクスの違いから不適切であり[古崎 02a]、このようなロール概念の性質を扱うには適切なインスタンスモデルの管理機構を用意する必要がある。

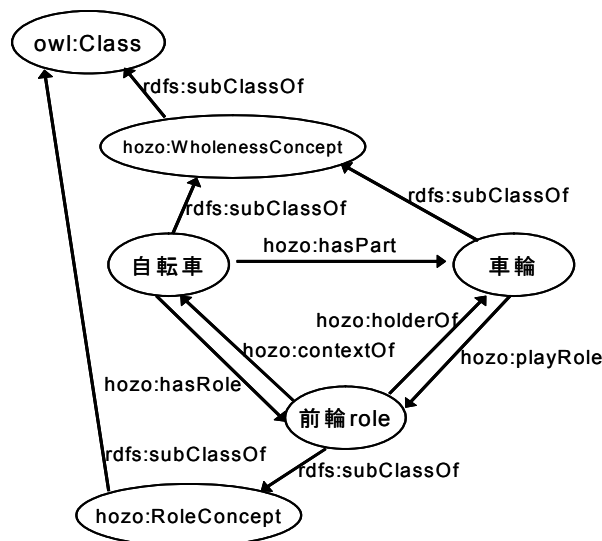


図5 詳細表現によるロール概念のOWL表記

```

...
<owl:Class rdf:ID="hozo:RoleConcept">
  <rdfs:subClassOf>
    <owl:Restriction owl:cardinality="1" >
      <owl:onProperty rdf:resource="hozo:contextOf"/>
    </owl:Restriction >
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction owl:cardinality="1" >
      <owl:onProperty rdf:resource="hozo:holderOf"/>
    </owl:Restriction >
  </rdfs:subClassOf>
</owl:Class>
...
<owl:Class rdf:ID="前輪role">
  <rdfs:subClassOf rdf:resource="hozo:RoleConcept" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hozo:contextOf"/>
      <owl:allValuesFrom rdf:resource="#自転車" />
    </owl:Restriction >
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hozo:holderOf"/>
      <owl:allValuesFrom rdf:resource="#車輪" />
    </owl:Restriction >
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="自転車">
  <rdfs:subClassOf rdf:resource="hozo:WholnessConcept"/>
  <rdfs:subClassOf>
    <owl:Restriction owl:cardinality="1" >
      <owl:onProperty rdf:resource="#hasPartAs前輪"/>
    </owl:Restriction >
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction owl:cardinality="1" >
      <owl:onProperty rdf:resource="#has前輪role"/>
    </owl:Restriction >
  </rdfs:subClassOf>
...
</owl:Class>
...
<owl:ObjectProperty rdf:ID="hozo:contextOf">
  <rdfs:domain rdf:resource="hozo:RoleConcept"/>
  <rdfs:range rdf:resource="hozo:WholnessConcept"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hozo:holderOf">
  <rdfs:domain rdf:resource="hozo:RoleConcept"/>
  <rdfs:range rdf:resource="hozo:WholnessConcept"/>
</owl:ObjectProperty>
...

```

図6 詳細表現によるOWL出力のXML表記(一部)

(3) ロール概念の階層の扱い

法造において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(図2)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。例えば、自転車をコンテキストとした「前輪 role」とオートバイをコンテキストとした「前輪 role」は、自転車とオートバイの上位概念である二輪車をコンテキストとした「前輪 role」の下位のロール概念になると考えられる。このようなロール概念の階層の取り扱いに関しては、法造の枠組み拡張に関する研究を進めている[砂川 05]。それに伴い、OWL形式でエクスポートし

た際のロール概念の階層化についても、検討を進めていく予定である。簡易表現においては rdfs:subPropertyOf, 詳細表現においては rdfs:subClassOf を用いて階層化が可能と考えられるが、domain やrangeを適切に設定するための制約のかけ方や、定義の継承などについても検討が必要と思われる。

3.3 実装状況

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.4.06)では、RDF(S)形式によるエクスポートおよび、簡易表現によるOWLエクスポートのプロトタイプが実装されている。この「法造」のエクスポート機能によるオントロジーの出力結果は Protégé[Protégé], Oiled[Oiled], OntoEdit[OntoEdit]といった、他の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が確認した。また、代表的な Semantic Web に向けたアノテーションツールである OntoMat[OntoMat]での利用も可能であること確認を確認している(ただし日本語によるオントロジーは未対応)。

今後、詳細表現によるOWLエクスポートの実装を進めると共に、ロール概念の階層の取り扱いについても検討を進めていく。

4. まとめ

本論文では、我々が開発を進めてきたオントロジー構築・利用環境「法造」の Semantic Web アプリケーション開発に向けた拡張として、OWL形式でのエクスポートの仕様を中心に述べた。簡易表現と詳細表現の2通りのエクスポート方法を比較すると、ロール概念の特徴を正しく反映したアプリケーション開発には、複雑なOWL表現が必要であることが分かる。このことから、ロール概念の特徴を十分に活かすには、HozoAPIを利用した開発がより用意であると考えられる。一方、簡易表現によるOWLエクスポートの結果は、他のOWL対応ツールで利用することが可能であることから、「法造」を簡単なOWLエディタとして利用することも可能であることが確認できた。今後、これらの知見を活かして「法造」の拡張を進めると共に、実際の Semantic Web アプリケーション開発への適用を進めていきたい。

参考文献

- [石川 02] 石川, 久保, 古崎, 来村, 溝口: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的设计と開発—石油精製プラントを例として—, 人工知能学会論文誌, Vol. 17, No. 5, pp. 585-597, 2002.
- [古崎 02a] 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.
- [古崎 02b] 古崎, 来村, 佐野, 本松, 石川, 溝口:オントロジー構築・利用環境「法造」の開発と利用—実規模プラントオントロジーを例として—, 人工知能学会論文誌, Vol. 17, No. 4, pp. 407-419, 2002.
- [溝口 99] 溝口, 池田, 来村:オントロジー工学基礎論, 人工知能学会誌, Vol.14, No.6, pp.1019-1032, 1999.
- [Oiled] <http://oiled.man.ac.uk/index.shtml>
- [OntoEdit] <http://www.ontoprise.de/>
- [OntoMat]<http://annotation.semanticweb.org/ontomat/index.html>
- [OWL] <http://www.w3.org/TR/webont-req/>
- [Protégé] <http://protege.stanford.edu/index.html>
- [砂川 05] 砂川, 古崎, 来村, 溝口:オントロジーにおけるコンテキストに依存する概念の取り扱い, 第19回人工知能学会全国大会論文集, 2D1-04,2005.
- [W3C-SW] W3C Semantic Web:<http://www.w3.org/2001/sw/>