

## INTAP 程度表現オントロジ Degree Expression Ontology (DEX)

INTAP 次世代 Web 委員会 Report Draft 0.7 13 Mar. 2007

Editors:

Hiroyuki Sato (NTT) (主に 1～3 章および付録 2 の編集)

Itaru Hosomi (NEC) (主に 4～5 章および付録 1 の編集)

Copyright © 2006-2007 INTAP 次世代 Web 委員会, All Rights Reserved.

---

### Abstract

「程度表現オントロジ (Degree Expression Ontology、略称 DEX)」は物事の特徴評価を表す値 (程度値) の表現の方法を規定したものです。さまざまなリソースに対する程度値を、ユーザが単純かつ簡易に表現することを可能にし、さらに記述された異なる程度表現間でリソースやその特徴毎に柔軟に程度値の比較や集計、相互運用性のあるマッピングを自動化することを目的として策定されています。本ドキュメントは、W3C で策定された RDF のデータモデルを用いて程度表現を記述する方法や、オントロジ記述言語 OWL を基盤とした程度値間の相対的な関係の意味的な定義など、「程度表現オントロジ」の仕様を記述したものです。

### Status of this document

2006 年 4 月から (予備検討はそれ以前から) 2007 年 3 月までの INTAP 次世代 Web 委員会での仕様検討および実装経験を経て一般に公開される最初のバージョンです。本仕様に規定されたボキャブラリを利用することで、相互に程度値のマッピングが可能な 4 種類の程度表現により、URI で示されるリソースの評価などを記述することが可能です。今後、パブリックコメントなどの反映により一部の仕様を変更する可能性があります。

---

### 1 Introduction

本ドキュメントは INTAP 次世代 Web 委員会で策定している「程度表現オントロジ (Degree Expression Ontology、略称 DEX)」の仕様について決定した事項をまとめたものです。

本ドキュメントでは程度表現を「物事の特徴評価を表す値 (程度値) の表現方法」と定義しています。

図1に具体的な程度表現の例を示します。商品「A」および「B」という物事の「価格」、「デザイン」、「信頼性」、「使い易さ」といった特性・観点に対して、ユーザの主観などによって、または何らかの客観的な基準に基づいて「△」、「◎」、「ふつう」、「良い」、「2」、「1」、「★★★」、「★」といった値（程度値）が付与されています。

商品	価格	デザイン	信頼性	使い易さ
A	△	ふつう	2	★★★
B	◎	良い	1	★

程度表現の例

図1: 程度表現の例

程度表現オントロジは、さまざまなリソースに対する程度値を、ユーザが単純かつ簡易に表現することを可能にし、さらに記述された異なる程度表現間でリソースやその特性毎に柔軟に程度値の比較や集計、相互運用性のあるマッピングを自動化することを目的として策定されています。

図2では「使い易さ」という特性に対して異なる方法で付与された程度値を比較するため、また1つの表として統合するために意味的マッピングを行った例を示しています。

商品	価格	使い易さ
A	△	★★★
B	◎	★

商品	価格	使い易さ
C	5,980	2
D	3,150	4

商品	価格(評価)	価格(金額)	使い易さ
A	△	—	3
B	◎	—	5
C	—	5,980	2
D	—	3,150	4

★★★ ←  
★ ←  
意味的マッピングが必要  
(数値が小さいほど高評価である場合の例)

図2: 異なる程度値間を比較したり統合したりする際の意味的マッピングの例

上記を実現するために、本仕様ではW3C (World Wide Web Consortium)で策定されたRDF (Resource Description Framework)を用いて程度表現を記述できるように語彙を定義し、程度値間の意味的マッピングを実現するために、段階や階層を持った程度値間の関係を同じくW3Cで策定されたOWL (Web Ontology Language)を用いて定義しています。本仕様書においてRDF/XMLおよびOWL/XMLの記述例を示す際に、本仕様で定義した語彙は以下のIRIの名前空間にバインドされた接頭辞dex:を用いて表現しています。

接頭辞(Prefix)	名前空間(IRI)
dex:	http://www.net.intap.or.jp/INTAP/s-web/2007/03/dex#

また、以降で使用する XML 文での記述例においては、接頭辞 dex: とその他の必要な名前空間に対する接頭辞の実体参照が DTD によって予め定義されているものとします。実体参照の定義は以下のようなものです。

```

<?xml version="1.0"?>
<!DOCTYPE dex [
<!ENTITY dex "http://www.net.intap.org/INTAP/s-web/2007/03/dex#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
]>
(以下 RDF 本文)

```

} DTD による定義

なお、以下に本仕様の検討を行ったメンバをサブ Working Group (WG) 毎に示します。

#### A. 程度値オントロジ策定 WG

程度値の記述方法と語彙を程度値オントロジとして定義。

細見 格 (日本電気株式会社)  
清水 昇 (慶應義塾大学)  
萩野 達也 (慶應義塾大学)  
岡部 雅夫 (東京電力株式会社)  
竹内 勝 (株式会社日立製作所)  
福重 貴雄 (松下電器産業株式会社)

#### B. 程度表現オントロジ策定 WG

程度表現の記述を行う際の基本モデルである程度文脈オントロジの策定。

佐藤 宏之 (日本電信電話株式会社)  
松井くにお (株式会社富士通研究所)  
武田 英明 (国立情報学研究所)  
飯島 正 (慶應義塾大学)

## C. ユースケース策定 WG [USECASES]

情報家電や Web コンテンツの信頼性などの評価に関するユースケースを検討し、程度表現オントロジの有効性の評価や課題を抽出。

森田 幸伯 (沖電気工業株式会社)

川村 隆浩 (株式会社東芝)

## D. 関連技術調査・相互運用性評価 WG [RELATEDWORKS]

程度表現の RDF/XML 以外での表現方法の検討や関連仕様の調査、程度表現オントロジの横断検索アプリケーションへの適用による実装評価。

渡邊 圭輔 (三菱電機株式会社)

乙守 信行 (株式会社ジャストシステム)

内藤 求 (株式会社ナレッジ・シナジー)

長尾 美保 (株式会社ジャストシステム)

## 2 Requirements (要求仕様)

程度表現の表現方法や表現に利用するボキャブラリを説明する前に、本章ではそれらがどのような要求仕様に基づいてデザインされたものであるかを示します。これらの要求仕様は、情報家電における機器接続性などの特性や、Web 上の商品レビューサイトにおける商品の特性を程度表現オントロジによって表現するなどのユースケース[USECASES]の検討に基づいて策定されています。

以下に程度表現オントロジの要求仕様を列挙します。

### 要求仕様 a.

程度表現オントロジは、URI で表現可能なリソースに対して特性とその程度値を記述することを可能にする。

### 要求仕様 b.

程度表現の記述は任意のユーザが行うことができる。リソースの著作者に限定せず、第三者も記述することを可能とする。

### 要求仕様 c.

程度表現を用いて記述された異なる程度値同士の意味的マッピング（意味の相互変換）を可能とする。

### 要求仕様 d.

ユーザは程度値や特性を記述するための標準仕様にあるボキャブラリを用いるだけで、

簡単に程度表現を（マークアップ言語などにより）記述することができる。

#### 要求仕様 e.

ユーザは標準仕様にあるボキャブラリを継承して、新しい程度値や特性を定義することができる。

#### 要求仕様 f.

程度表現は OWL DL の範囲で記述できるようにする。

**Note** 従来のオントロジや **microformats** では、ある評価対象の種類や主題、著者といった属性を表す語彙を定義しているものの、評価結果としての値を表すための語彙は記述者に任されたり単純な 5 段階評価のみに限定されたりしています。

実際の Web 上では、評価結果を表す“程度”の表現方法は多岐にわたります。図 2 に示したように、特定のマーク（星型など）の数で肯定的意見の程度の大きさを表したり、アルファベットとプラス／マイナス記号の組合せで格付けなどを表したりする場合もあります。一方で、ブログなどの自由記述文では評価の良し悪しを表すような形容詞や点数で程度の大小が表現されています。

**RDF** で評価内容が記述され、エージェントが評価対象や評価者を識別できるとしても、評価結果が何段階評価の何番目であるのかといった“程度”の記述内容が理解できなければ、異なるサイトの評価内容を比較／統合することは困難です。これは従来、こうした評価結果などを表す標準的な語彙が定義されていなかったためですが、異なる評価内容間の関係をオントロジで定義することにより、様々な評価やその他のレイティング情報を意味的に比較／統合することが可能になります。

程度を表す語彙の定義方法としては、オントロジで意味的に定義するほかに、必要な語彙のセットを全て用意しておくという方法もあります。しかし、同じ 5 段階評価でも、1 を基準として 5 を最高点とした評価結果の表し方もあれば、0 を基準として肯定と否定が各 2 段階ずつという場合も考えられます。また、評価結果には 3 段階や 4 段階の評価もあります。数種類の表現方法で大半のケースはカバーできると思われませんが、例外的な表現方法を受け容れられないのは既存の情報を対応させる上で問題です。オントロジで必要最小限の語彙を部品レベルで定義することにより、必要に応じて容易に拡張できることが、程度表現の語彙定義としては望ましいと考えます。

**Note** 時間や役割に応じて変化する評価値を扱うことは本仕様の範囲外です。例えば、教材などのリソースに対して程度値を付与する場合、最初に評価したときと、習熟などを経て一定時間経過した後では教材の「難しさ」の評価が変わる可能性があります。また、教師の役割からみれば易しいが、生徒の役割からみると難しいという場合もあ

ります。これらについては程度表現を与えた日時や評価者名などを程度表現中に記述することによって、集計時に程度表現を付与した時間や評価者の役割を判別した集計を行うことで問題を回避できる可能性があります。

### 3 Model (程度文脈オントロジ DEX Context Ontology)

本章では、程度表現オントロジを用いて程度表現の記述を行う際の基本モデルを示します。

#### 3.1 リソースに対する総合評価

図3の程度表現の記述例を用いて、基本的なモデルとボキャブラリを説明します。図中央の空白ノードは評価を表し、評価クラス (`dex:Rate`) のインスタンスとして表現されています。この空白ノードをサブジェクト (主語) としてプロパティ (述語) `dex:hasTheme` がオブジェクト (目的語) であるリソースにつながっています。これは評価の対象を示しています。また、`dex:hasDegree` のオブジェクトが実際の程度値を表現するノードを示しています。これは程度クラスのサブクラス `dex:Balance` のインスタンスであり、かつ程度値クラスのサブクラス `dex:StrongPositive` のインスタンスであることから、5段階評価の真ん中に基準値をおいた評価において、最高の評価であることを示しています。この `dex:hasDegree` の値域に関しては4章に詳細を記述しています。

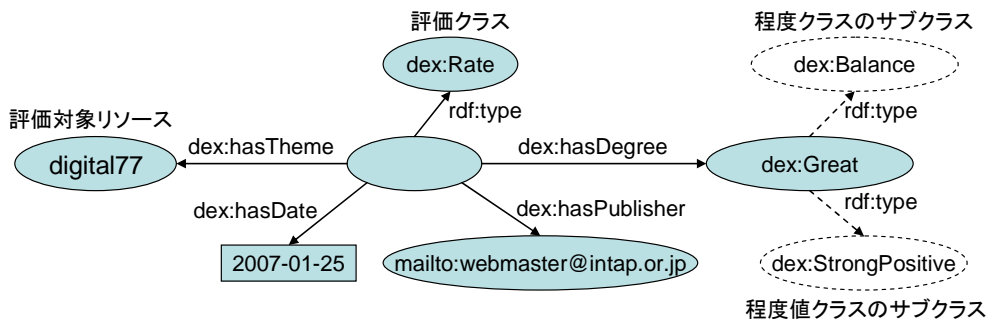


図3：程度表現の記述例

図3の RDF/XML での表現は以下です。評価対象リソース(デジタルカメラ”digital77”)の機種を示す URI を指定し、デジタルカメラの評価をメタデータとして記述する例を示しています。

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dex="http://www.net.intap.org/INTAP/s-web/2007/03/dex#">

  <dex:Rate>
    <dex:hasTheme rdf:resource="http://intap.or.jp/product/camera#digital77"/>
    <dex:hasDegree rdf:resource="&dex;Great"/>
    <dex:hasDate rdf:datatype="&xsd;date">2007-01-25</dex:hasDate>
    <dex:hasPublisher rdf:resource="mailto:webmaster@intap.or.jp"/>
  </dex:Rate>
</rdf:RDF>
```

程度表現の記述に用いるクラス

クラス		ノート
dex:Rate	評価クラス	この評価クラスのサブクラスとして、特定の観点・特性を評価するクラスを独自に定義可能。
dex:Degree	程度クラス	具体的な程度の種類の記述には通常このクラスのサブクラスのインスタンスを用いる。程度クラスおよびそのサブクラス (dex:Balance など) の詳細は 4 章に記述。
dex:Value	程度値クラス	具体的な程度値の記述には通常このクラスのサブクラスのインスタンスを用いる。程度値クラスおよびそのサブクラス (dex:StrongPositive など) の詳細は 4 章に記述。

程度表現の記述に用いるプロパティ

プロパティ (type)	domain	range	ノート
dex:hasTheme (owl:ObjectProperty)	dex:Rate	owl:Thing	
dex:hasRate (owl:ObjectProperty)	owl:Thing	dex:Rate	dex:hasRate と dex:hasTheme は owl:inverseOf の関係。
dex:hasDegree (owl:ObjectProperty)	dex:Rate	owl:intersectionOf (dex:Degree dex:Value)	このプロパティの値となるインスタンスのクラスは dex:Degree かつ dex:Value
dex:hasDate (owl:DatatypeProperty)	dex:Rate	xsd:date	
dex:hasDateTime (owl:DatatypeProperty)	dex:Rate	xsd:dateTime	程度表現を記述した時刻の記述が必要な場合は、dex:hasDate の代わりにこのプロパティを用いて記述可能。
dex:hasPublisher (owl:ObjectProperty)	dex:Rate	owl:Thing	このプロパティの値としては、評価が記載されていた Web サイトや Web ページの URL、評価を行ったユーザのメールアドレスなど、評価がどこでもしくは誰によって行われたかを示すユニークな ID となるものを推奨。



**Note** `dex:hasRate` と `dex:hasTheme` は `owl:inverseOf` の関係になっています。評価対象と評価を結びつけるためにどちらの述語を用いても構いませんが、本仕様書では両方の記述が可能な場合は、図3の基本モデルに示した `dex:hasTheme` を用いた記述方法を推奨します。理由は `dex:hasTheme` のオブジェクトとして評価対象を示す方が評価対象と評価を独立して記述し易く、その結果、複数のデータに対する評価を一度に記述することができたりするためです。

**Note** 評価クラス (`dex:Rate`) のインスタンスを `subject` とするプロパティとその値の追加については特に制限はありません。例えば、必要があれば自分がつけた評価に自信があることなどを表現しても構いません。またセルフレーティングなのかサードパーティによるものであるかなどを示すことも可能です。評価の属性 (プロパティ) に関しては、今後必要に応じて `dex:hasPublisher` や `dex:hasDate` に加えてボキャブラリを拡張する可能性があります。

### 3.2 リソースに対する特性・観点についての評価

本節ではリソースの特性・観点について評価を行う方法について説明します。「使い易さ」など評価の特性や観点は評価クラスのサブクラスとして定義します。例えば、`ex:使い易さ` を `dex:Rate` のサブクラスとして定義した場合、そのインスタンスとして評価を表現することにより、リソースに対して「使い易さ」の評価をすることが可能になります。

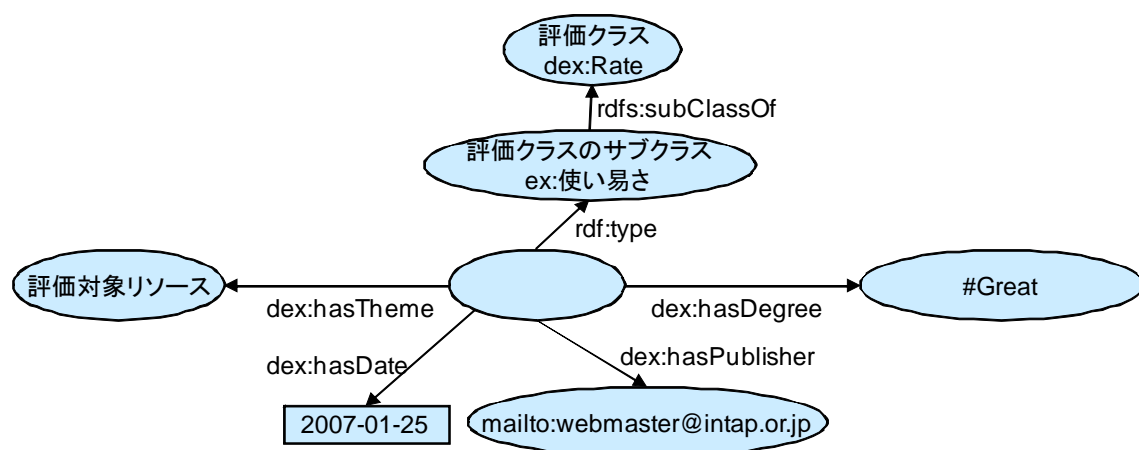


図5：特性や観点に関する評価の記述例

ex: 使い易さ は次のように定義できます。

```
<owl:Class rdf:ID="使い易さ">
  <rdfs:subClassOf rdf:resource="&dex;Rate"/>
</owl:Class>
```

**Note** この方法で特性や観点を指定することにより、それらのマッピングが容易になります。程度表現オントロジとは別に「ex:使い易さ」と「ex:操作性」は **equivalent** とすることを記述し、評価の集約などに利用することが可能になります。また評価クラスのサブクラス階層を定義することができます。例えば車を評価するために ex:燃費という評価クラスのサブクラスを定義し、さらに ex:10 モード燃費と ex:10・15 モード燃費を ex:燃費のサブクラスとして定義したとします。この際に ex:10 モード燃費と ex:10・15 モード燃費のインスタンスとしての評価があった場合、それらを集約して ex:燃費の評価に相当するもの決定することが考えられます。これらはいずれも程度表現オントロジの仕様の範囲外ですが、4 章で説明する評価値のマッピングとあわせて、機械的に特性や観点毎の集計を行うために考慮しておく必要があります。

参考のため本仕様の付録 2 に主に商品の評価などに用いることが可能な評価項目として、評価クラスのサブクラスのボキャブラリを **dex:**とは別のユニークな名前空間と共に定義したものを示します。

**Note** リソースの特性や観点について評価する場合に、リソースのある特定の項目について評価をしたい場合があります。例えばある携帯電話の「メール」の「使い易さ」を評価する場合です。この場合に、図 6 のように記述することも考えられます。このような記述方法を用いると、評価対象の機能を特定することにより、評価クラスのサブクラスの定義を増やさずに、同じ評価対象リソースの複数の機能の評価が可能となります。一方、評価項目（評価クラスのサブクラス）として直接「ex:メールの使い易さ」を定義して図 5 に示した記述方法で評価対象リソースの「ex:メールの使い易さ」を評価することも可能です。この記述方法は特定の機能の使い易さのみが評価項目として重要である場合などは有効であると考えられます。評価クラスのサブクラスのデザインにはこれらを留意して行う必要があります。

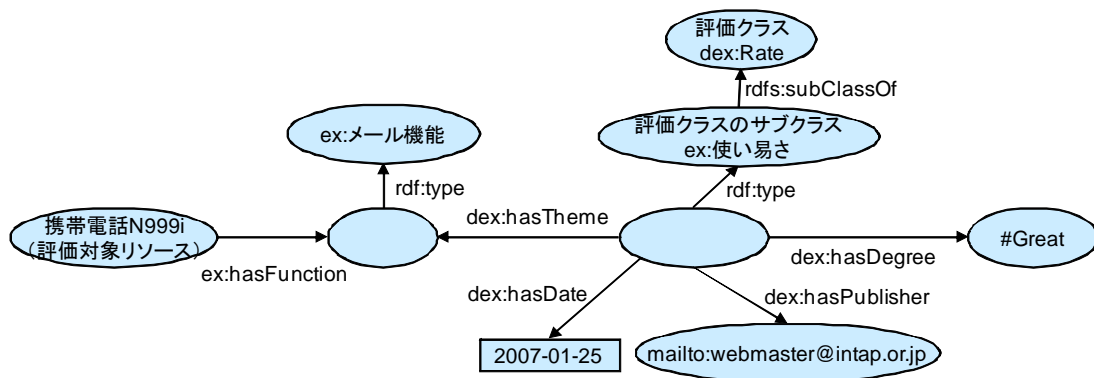


図 6：評価対象リソースの特定の機能を評価する記述の一例

## 4 Vocabulary (程度値オントロジ DEX Value Ontology)

本章では、程度の値を表す語彙を定義する程度値オントロジを4つの基本的な程度(Degree)クラス(Score, Balance, Vote, Flag)とそれらの中で個々の値を表す 12 個の程度値(Value)クラス(Positive, Neutral, Negative, etc.)から構成する案について述べます。設計方針としては、一般的な用途の大半をカバーしつつ可能な限り覚え易く使い易い小さなオントロジを定義し、また独自のカスタマイズ(語彙の追加定義)がし易いものを目指しています。

### 4.1 程度値のクラスとプロパティ

#### 4.1.1 程度クラスと離散的程度クラス

程度を表すクラスの最上位クラス `dex:Degree` は `owl:Class` のサブクラスであり、程度値が離散値の場合は `dex:Degree` のサブクラスとして離散的程度クラス `dex:DiscreteDegree` を使うこととします。

#### 4.1.2 程度値クラス

個々の程度の値を表す程度値クラス(`dex:Value` とそのサブクラス群)を程度クラス(`dex:Degree` とそのサブクラス群)とは区別して定義します。程度クラスは、必要な種類の程度値クラスを用いて定義します。

#### 4.1.3 程度値記述のためのプロパティ

程度値間の順序制約を表す `isSubsequentTo` プロパティを定義します。`isSubsequentTo` プロパティは、ドメイン、レンジともに `dex:Value` です。

## 4.2 語彙のクラス定義

以下では、程度値を表す具体的なクラスとそれらの間の関係を定義します。

### 4.2.1 4つの程度クラスとそれらの相互関係

程度値オントロジでは、以下の4つの程度クラスを標準で用意しています。

程度クラス	上	←程度値→					下	定義済みインスタンス
Score	4	3	2	1	0		{Five, Four, Three, Two, One}	
Balance		2		1	0	-1	-2	{Great, Good, Fair, Bad, Terrible}
Vote			1		0	-1		{Yes, Void, No}
Flag			1		0			{Valid, Invalid}

← 積極的肯定                      積極的否定 →

図 7 4つの程度クラスと各程度値間の関係

各程度クラスの値(程度値)は、図 7 では数字を用いていますが、それぞれは左右方向に上位/下位の順序関係を持つ程度値クラスのサブクラス(後述)として定義し、程度値間の算術演算は定義していません。また、図 7 における程度値のマイナス(-)は積極的否定を意味し、マイナスが付かない値は積極的肯定を表します。

### 4.2.2 程度値オントロジの OWL/XML による定義案

程度値の最上位クラスである Value クラスに対して、図 8 のような 12 種類のサブクラスを定義します。

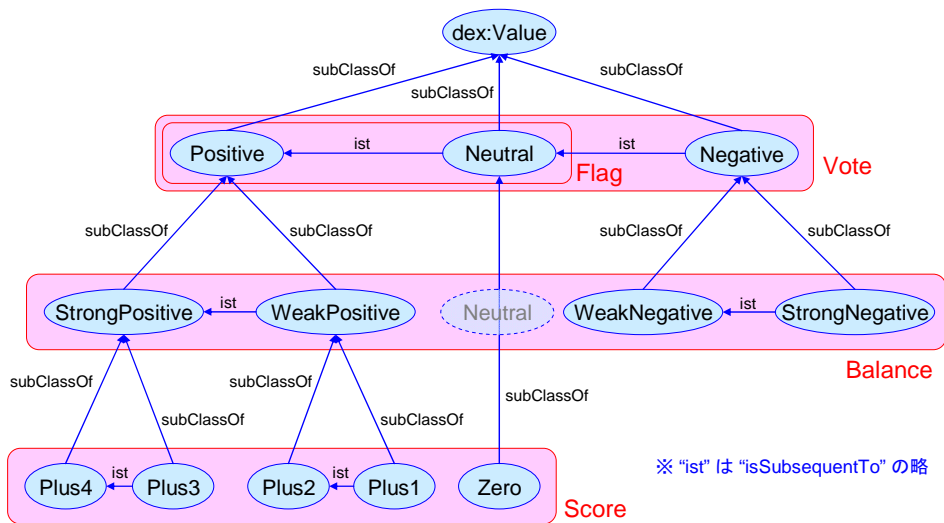


図 8 程度値クラスとそのサブクラスの間関係

`dex:Zero` および `dex:Plus1`～`dex:Plus4` の程度値クラスは、他の程度値クラスとは異なる暫定的な名称となっています。

また、これらの程度値を使ってユーザが独自の程度クラスを定義することもできます。例えば、{`Positive`, `Negative`} の 2 つの程度値クラスのみを使って下記のような正負を表す極性クラス (`dex:Polarity`) を定義することができます。

```

<!-- 極性 (Polarity) クラス -->
<owl:Class rdf:ID="Polarity" >
  <rdfs:subClassOf rdf:resource=" #DiscreteDegree" />
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection" >
        <owl:Class rdf:about=" #Positive" />
        <owl:Class rdf:about=" #Negative" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype=" &xsd:string" >Polarity</rdfs:label>
  <rdfs:label rdf:datatype=" &xsd:string" xml:lang=" ja" >極性</rdfs:label>
</owl:Class>

```

上記のように `dex:Polarity` を定義するだけで、他の `dex:Vote` や `dex:Balance` などとの程度値の対応関係が分かるので、語彙の拡張は容易です。

### 4.3 程度の記述例

下記は、程度に関するメタデータの RDF/XML による記述例です。

```
<dpr:Usability rdf:ID="Rate_IntapDigital77">
  <dex:hasTheme rdf:resource="http://intap.or.jp/product/camera.html#digital77"/>
  <dex:hasDegree
rdf:resource="http://www.net.intap.or.jp/INTAP/s-web/2007/03/dex#Great"/>
</dpr:Usability>
```

`dex:hasDegree` プロパティの行は、予め上記のメタデータを記述するドキュメントに

```
xmlns:dex="http://www.net.intap.or.jp/INTAP/s-web/2007/03/dex#"
```

のように定義してあれば、

```
<dex:hasDegree rdf:resource="&dex;Great"/>
```

と書くだけで済みます。

程度値のインスタンス `dex:Great` は程度値オントロジで定義済みの語彙ですが、もし独自の語彙を使いたければ、程度値クラスと評価クラスを用いて例えば次のように「最高」を定義できます。

```
<dex:StrongPositive rdf:ID="最高">
  <rdf:type rdf:resource="&dex;Balance"/>
</dex:StrongPositive>
```

また、上記の「最高」は `dex:Great` と同じ意味のインスタンスになりますので、

```
<dex:StrongPositive rdf:ID="最高">
  <owl:sameAs rdf:resource="&dex;Great"/>
</dex:StrongPositive>
```

という定義の仕方もできます。

## 5 関連仕様

本章では、程度表現オントロジを用いたメタデータを RDFa [RDFa]の記法に基づいて XHTML のコンテンツに埋め込む例を示します。以下が、RDFa を使った程度表現の記述例です。

```
<html xmlns:dex="http://www.net.intap.or.jp/INTAP/s-web/2007/03/dex#">
<head><title>A sample of RDFa description with DEX</title></head>
<body>
<div class="dex:Rate">
  <a rel="dex:hasTheme"
href="http://intap.or.jp/product/camera.html#digital77">
  
  New Digital Camera: INTAP digital77</a>
<div><p>
  "2007年待望の最新デジタルカメラ INTAP digital77 デビュー!"
</p></div>
評価: <span rel="dex:hasDegree"
href="http://www.net.intap.or.jp/INTAP/s-web/2007/03/dex#Five">*****</span>
<p>Review by
  <a rel="dex:hasPublisher" href="http://intapreview.com/2007/03/">INTAP レビュー
2007年3月号</a>,
  <span property="dex:hasDate" content="20070310">2007年3月10日</span>
</p>
</div>
</body>
</html>
```

このように RDFa を使うことで、microformats [MICROFORMATS]と同程度の簡単さで評価とその程度の大きさを HTML 文書の中にメタデータとして書くことができます。なお、RDFa を含む関連仕様の詳細については[RELATEDWORKS]に記載されています。

### [USECASES]

平成 18 年度セマンティック Web 技術の調査報告書 第 2 章 2.3 ユースケースを参照。

### [RELATEDWORKS]

平成 18 年度セマンティック Web 技術の調査報告書 第 2 章 2.4 関連技術・相互運用性評価を参照。

### **[RDFa]**

平成 18 年度セマンティック Web 技術の調査報告書 第 2 章 2.4 関連技術・相互運用性評価を参照。RDFa の最新の仕様は <http://www.w3.org/2006/07/SWD/RDFa/> を参照。

### **[MICROFORMATS]**

平成 18 年度セマンティック Web 技術の調査報告書 第 2 章 2.4 関連技術・相互運用性評価を参照。microformats の仕様等については <http://microformats.org/> を参照。



## 付録 1 : 程度値オントロジの OWL/XML 定義

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.net.intap.or.jp/INTAP/s-web/2007/03/dex#"
  xml:base="http://www.net.intap.or.jp/INTAP/s-web/2007/03/dex">

  <owl:Ontology rdf:about=""/>

  <!-- 基本語彙 -->

  <owl:Class rdf:ID="Degree">
    <rdfs:label rdf:datatype="xsd:string">Degree</rdfs:label>
    <rdfs:label rdf:datatype="xsd:string" xml:lang="ja">程度</rdfs:label>
  </owl:Class>

  <owl:Class rdf:ID="DiscreteDegree">
    <rdfs:subClassOf rdf:resource="#Degree"/>
    <rdfs:label rdf:datatype="xsd:string">Discrete Degree</rdfs:label>
    <rdfs:label rdf:datatype="xsd:string" xml:lang="ja">離散的程度</rdfs:label>
  </owl:Class>

  <owl:Class rdf:ID="Value">
    <rdfs:label rdf:datatype="xsd:string">Value</rdfs:label>
    <rdfs:label rdf:datatype="xsd:string" xml:lang="ja">程度値</rdfs:label>
  </owl:Class>

  <owl:ObjectProperty rdf:ID="isSubsequentTo">
    <rdf:type rdf:resource="owl:TransitiveProperty"/>
    <rdfs:domain rdf:resource="#Value"/>
    <rdfs:range rdf:resource="#Value"/>
```

```
</owl:ObjectProperty>
```

```
<!-- 程度値クラス -->
```

```
<owl:Class rdf:ID="Positive">  
  <rdfs:subClassOf rdf:resource="#Value"/>  
</owl:Class>
```

```
<owl:Class rdf:ID="Neutral">  
  <rdfs:subClassOf rdf:resource="#Value"/>  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty>  
        <owl:ObjectProperty rdf:about="#isSubsequentTo"/>  
      </owl:onProperty>  
      <owl:allValuesFrom rdf:resource="#Positive"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

```
<owl:Class rdf:ID="Negative">  
  <rdfs:subClassOf rdf:resource="#Value"/>  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty>  
        <owl:ObjectProperty rdf:about="#isSubsequentTo"/>  
      </owl:onProperty>  
      <owl:allValuesFrom rdf:resource="#Neutral"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

```
<owl:Class rdf:ID="StrongPositive">  
  <rdfs:subClassOf rdf:resource="#Positive"/>  
</owl:Class>
```

```
<owl:Class rdf:ID="WeakPositive">
  <rdfs:subClassOf rdf:resource="#Positive"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#isSubsequentTo"/>
      </owl:onProperty>
      <owl:allValuesFrom rdf:resource="#StrongPositive"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:ID="WeakNegative">
  <rdfs:subClassOf rdf:resource="#Negative"/>
</owl:Class>
```

```
<owl:Class rdf:ID="StrongNegative">
  <rdfs:subClassOf rdf:resource="#Negative"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#isSubsequentTo"/>
      </owl:onProperty>
      <owl:allValuesFrom rdf:resource="#WeakNegative"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:ID="Plus4">
  <rdfs:subClassOf rdf:resource="#StrongPositive"/>
</owl:Class>
```

```
<owl:Class rdf:ID="Plus3">
  <rdfs:subClassOf rdf:resource="#StrongPositive"/>
  <rdfs:subClassOf>
    <owl:Restriction>
```

```
    <owl:onProperty>
      <owl:ObjectProperty rdf:about="#isSubsequentTo"/>
    </owl:onProperty>
    <owl:allValuesFrom rdf:resource="#Plus4"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:ID="Plus2">
  <rdfs:subClassOf rdf:resource="#WeakPositive"/>
</owl:Class>
```

```
<owl:Class rdf:ID="Plus1">
  <rdfs:subClassOf rdf:resource="#WeakPositive"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#isSubsequentTo"/>
      </owl:onProperty>
      <owl:allValuesFrom rdf:resource="#Plus2"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:ID="Zero">
  <rdfs:subClassOf rdf:resource="#Neutral"/>
</owl:Class>
```

<!-- 程度クラス -->

<!-- 程度クラス／フラグ (Flag) クラス -->

```
<owl:Class rdf:ID="Flag">
  <rdfs:subClassOf rdf:resource="#DiscreteDegree"/>
  <rdfs:subClassOf>
    <owl:Class>
```

```
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Positive"/>
      <owl:Class rdf:about="#Neutral"/>
    </owl:unionOf>
  </owl:Class>
</rdfs:subClassOf>
<rdfs:label rdf:datatype="&xsd:string">Flag</rdfs:label>
<rdfs:label rdf:datatype="&xsd:string" xml:lang="ja">フラグ</rdfs:label>
</owl:Class>
```

<!-- 程度クラス／投票 (Vote) クラス -->

```
<owl:Class rdf:ID="Vote">
  <rdfs:subClassOf rdf:resource="#DiscreteDegree"/>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Positive"/>
        <owl:Class rdf:about="#Neutral"/>
        <owl:Class rdf:about="#Negative"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="&xsd:string">Vote</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string" xml:lang="ja">投票</rdfs:label>
</owl:Class>
```

<!-- 程度クラス／バランス (Balance) クラス -->

```
<owl:Class rdf:ID="Balance">
  <rdfs:subClassOf rdf:resource="#DiscreteDegree"/>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#StrongPositive"/>
        <owl:Class rdf:about="#WeakPositive"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="&xsd:string">Balance</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string" xml:lang="ja">バランス</rdfs:label>
</owl:Class>
```

```
        <owl:Class rdf:about="#WeakNegative"/>
        <owl:Class rdf:about="#StrongNegative"/>
    </owl:unionOf>
</owl:Class>
</rdfs:subClassOf>
<rdfs:label rdf:datatype="&xsd:string">Balance</rdfs:label>
<rdfs:label rdf:datatype="&xsd:string" xml:lang="ja">バランス</rdfs:label>
</owl:Class>
```

<!-- 程度クラス/スコア (Score) クラス -->

```
<owl:Class rdf:ID="Score">
  <rdfs:subClassOf rdf:resource="#DiscreteDegree"/>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Plus4"/>
        <owl:Class rdf:about="#Plus3"/>
        <owl:Class rdf:about="#Plus2"/>
        <owl:Class rdf:about="#Plus1"/>
        <owl:Class rdf:about="#Zero"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
  <rdfs:label rdf:datatype="&xsd:string">Score</rdfs:label>
  <rdfs:label rdf:datatype="&xsd:string" xml:lang="ja">スコア</rdfs:label>
</owl:Class>
```

<!-- 評価クラス -->

```
<owl:Class rdf:ID="Rate">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasTheme"/>
```

```

    </owl:onProperty>
    <owl:someValuesFrom rdf:resource="&owl;Thing"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasDegree"/>
    </owl:onProperty>
    <owl:allValuesFrom rdf:resource="#Value"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasPublisher"/>
    </owl:onProperty>
    <owl:allValuesFrom rdf:resource="&owl;Thing"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:ID="hasDate"/>
    </owl:onProperty>
    <owl:allValuesFrom rdf:resource="&xsd;date"/>
  </owl:Restriction>
</rdfs:subClassOf>
  <rdfs:label rdf:datatype="&xsd:string" xml:lang="ja">評価</rdfs:label>
</owl:Class>

```

<!-- 評価記述用プロパティ -->

```

<owl:ObjectProperty rdf:about="#hasTheme">
  <rdfs:domain rdf:resource="#Rate"/>
</owl:ObjectProperty>

```

```
<owl:ObjectProperty rdf:about="#hasRate">
  <rdfs:range rdf:resource="#Rate"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="#hasDegree">
  <rdfs:domain rdf:resource="#Rate"/>
  <rdfs:range>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Degree"/>
        <owl:Class rdf:about="#Value"/>
      </owl:intersectionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="#hasPublisher">
  <rdfs:domain rdf:resource="#Rate"/>
</owl:ObjectProperty>
```

```
<owl:DatatypeProperty rdf:about="#hasDate">
  <rdfs:domain rdf:resource="#Rate"/>
  <rdfs:range rdf:resource="&xsd:date"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#hasDateTime">
  <rdfs:domain rdf:resource="#Rate"/>
  <rdfs:range rdf:resource="&xsd:dateTime"/>
</owl:DatatypeProperty>
```

<!-- 評価値のインスタンス定義 -->

```
<Zero rdf:ID="One">
  <rdf:type rdf:resource="#Score"/>
</Zero>
```



```
<Plus1 rdf:ID="Two">
  <rdf:type rdf:resource="#Score"/>
</Plus1>

<Plus2 rdf:ID="Three">
  <rdf:type rdf:resource="#Score"/>
</Plus2>

<Plus3 rdf:ID="Four">
  <rdf:type rdf:resource="#Score"/>
</Plus3>

<Plus4 rdf:ID="Five">
  <rdf:type rdf:resource="#Score"/>
</Plus4>

<StrongPositive rdf:ID="Great">
  <rdf:type rdf:resource="#Balance"/>
</StrongPositive>

<WeakPositive rdf:ID="Good">
  <rdf:type rdf:resource="#Balance"/>
</WeakPositive>

<Neutral rdf:ID="Fair">
  <rdf:type rdf:resource="#Balance"/>
</Neutral>

<WeakNegative rdf:ID="Bad">
  <rdf:type rdf:resource="#Balance"/>
</WeakNegative>

<StrongNegative rdf:ID="Terrible">
  <rdf:type rdf:resource="#Balance"/>
</StrongNegative>
```

```
<Positive rdf:ID="Yes">
  <rdf:type rdf:resource="#Vote"/>
</Positive>
```

```
<Neutral rdf:ID="Void">
  <rdf:type rdf:resource="#Vote"/>
</Neutral>
```

```
<Negative rdf:ID="No">
  <rdf:type rdf:resource="#Vote"/>
</Negative>
```

```
<Positive rdf:ID="Valid">
  <rdf:type rdf:resource="#Flag"/>
</Positive>
```

```
<Neutral rdf:ID="Invalid">
  <rdf:type rdf:resource="#Flag"/>
</Neutral>
```

```
</rdf:RDF>
```

## 付録 2 : DEX Products Rating Vocabularies

程度表現を記述する際に指定可能な評価項目(特性)のボキャブラリの1つを定義しています。評価項目のボキャブラリは、評価対象リソースが所属するドメインに応じてさまざまなものを自由に定義可能ですが、ここでは主に製品などの評価対象リソースに対して、ユーザが主観的な評価を付与する際に利用できるボキャブラリを示します。これらのボキャブラリは全て評価クラス `dex:Rate` のサブクラスです。

定義した語彙は以下の IRI の名前空間にバインドされた接頭辞 `dpr:` を用いて表現しています。

接頭辞(Prefix)	名前空間(IRI)
<code>dpr:</code>	<a href="http://www.net.intap.or.jp/INTAP/s-web/2007/03/dpr#">http://www.net.intap.or.jp/INTAP/s-web/2007/03/dpr#</a>

クラス	ラベル	ノート
<code>dpr:Usability</code>	使い易さ	簡単に使えるか、操作がし易いかといった使い易さに関する満足度。
<code>dpr:Functionality</code>	機能性	意図した通りに役だったかなど機能に関する満足度。
<code>dpr:Performance</code>	性能	性能に関する満足度。
<code>dpr:Extensibility</code>	拡張性	他のプロダクトとの組み合わせなど発展的に利用することに関する満足度。
<code>dpr:CostPerformance</code>	コストパフォーマンス	価格に関してコストパフォーマンスなどに関する満足度。
<code>dpr:Size</code>	サイズ	大きさや小ささなどサイズに関する満足度。
<code>dpr:Reliability</code>	信頼性	故障が少ない、安定して使える、安全に使えるといった信頼性に関する満足度。
<code>dpr:Design</code>	デザイン性	デザインに関する満足度。
<code>dpr:Maintainability</code>	保守性	アフターサービスやサポート体制などに関する継続利用する上での満足度。
<code>dpr:EnvironmentConscious</code>	環境配慮	環境への負荷低減のための配慮などに関する満足度

上記で定義されたボキャブラリ(dpr:Usability)を用いた程度表現の記述例は以下です。4.3節および付録1に示されている評価値のインスタンス(本例では dex:Five)を用いて記述しています。

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dex="http://www.net.intap.org/INTAP/s-web/2007/03/dex#"
  xmlns:dpr="http://www.net.intap.org/INTAP/s-web/2007/03/dpr#">

  <dpr:Usability>
    <dex:hasTheme rdf:resource="http://intap.or.jp/product/camera#digital77"/>
    <dex:hasDegree rdf:resource="&dex;Five"/>
    <dex:hasDate rdf:datatype="&xsd;date">2007-01-25</dex:hasDate>
    <dex:hasPublisher rdf:resource="mailto:webmaster@intap.or.jp"/>
  </dpr:Usability>

</rdf:RDF>
```