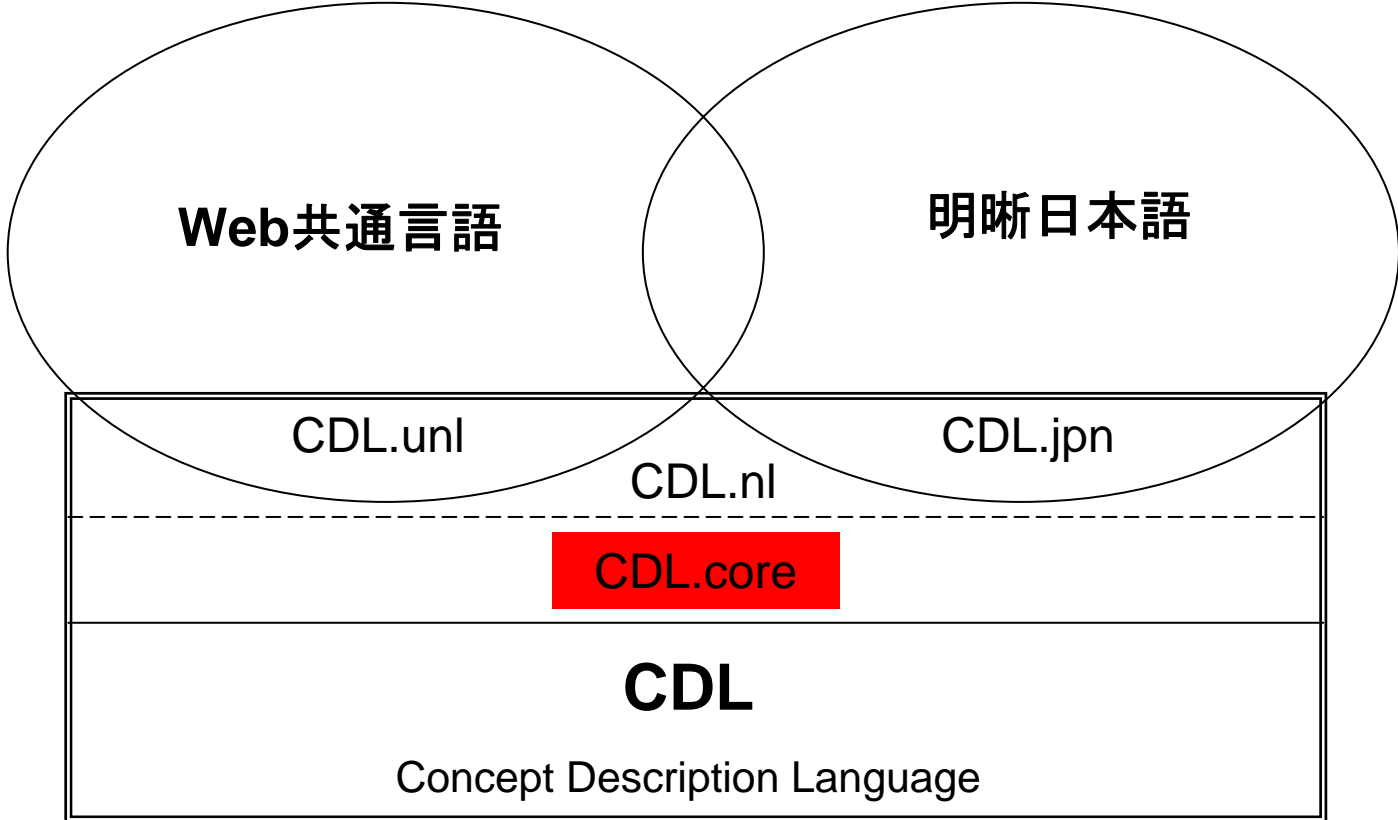
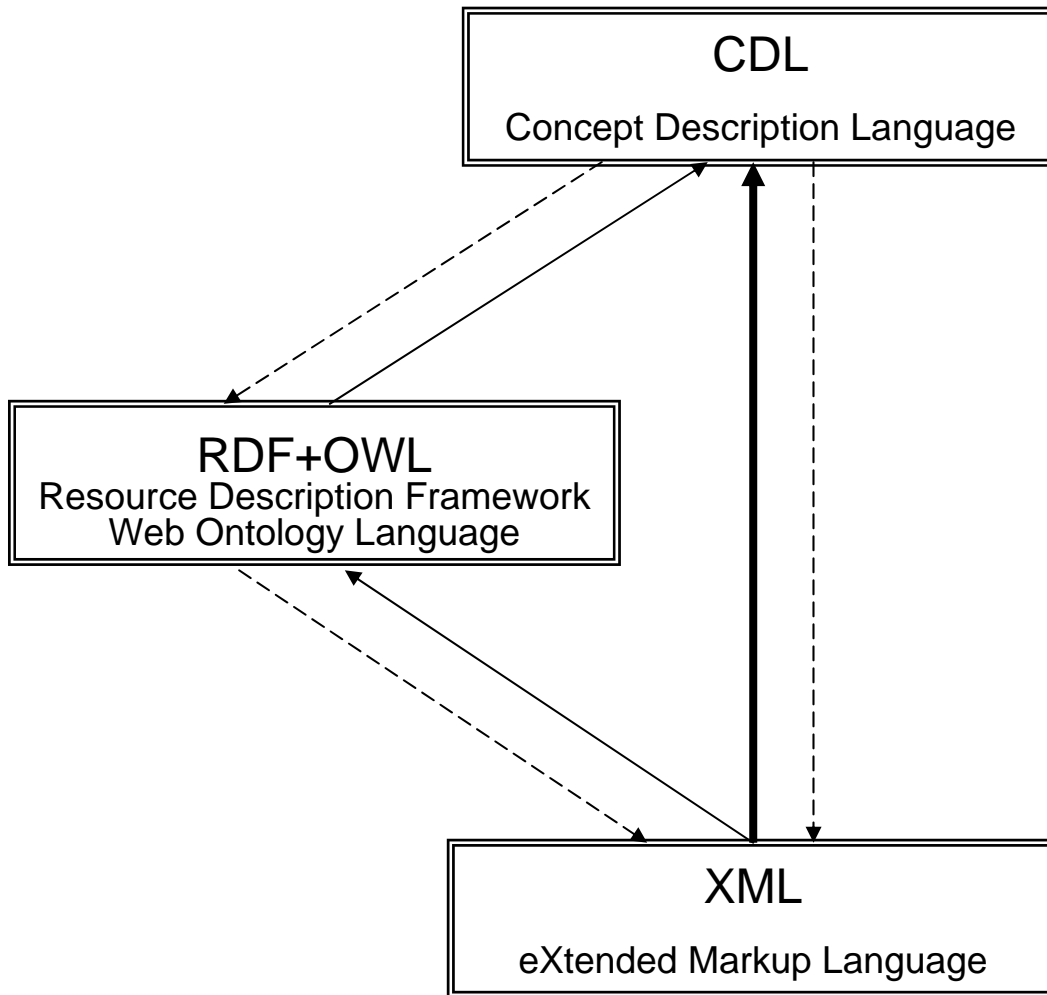


# CDL.core 言語仕様

2006年6月3日

横井俊夫  
ISeC, 東京工科大学





# CDL.coreとXML

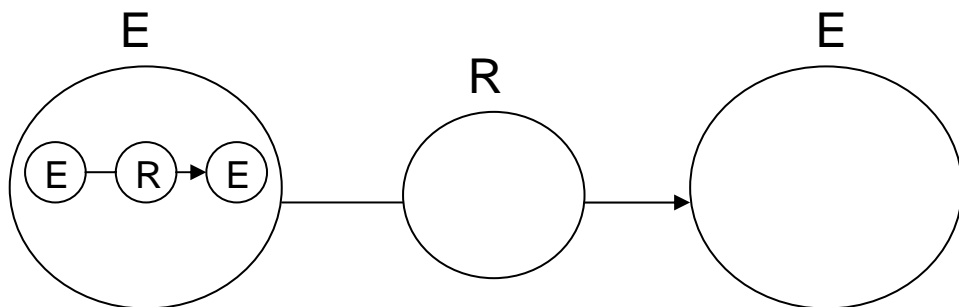
## – CDL:

- 記述 (description)
- コンテンツの意味構造の概念化の記述
- 意味構造のデータモデルはハイパーなネットワーク構造
- 概念の定義は対象から独立した概念定義辞書にまとめられる

## – XML:

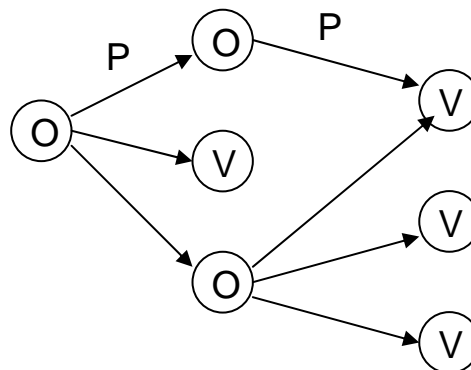
- 注記 (annotation)
- コンテンツの文書構造 (構文構造) の注記
- 文書構造のデータモデルは入れ子となるツリー構造
- タグの定義は対象に付随し、名前空間機能で相互参照する

# CDL.coreとRDF+OWL



CDLでは、EntityとRelation(その縮退形のAttributeとValue)でネットワーク構造を表現、それぞれにComplexタイプとElementタイプを設ける。Complexタイプがハイパーなノードである。

RDFでは、もともとのRDFの記述形式を引き継ぎ、Object、Property、Valueの3つ組でフレーム形式のネットワーク構造を表現する。



# CDL.core

## – Model&Syntax

- 意味の概念化モデルをハイパーなネットワーク構造として表現
- 表現のための基本データ構造のグラフ表記、テキスト表記、ビジュアル化表記

## – Schema

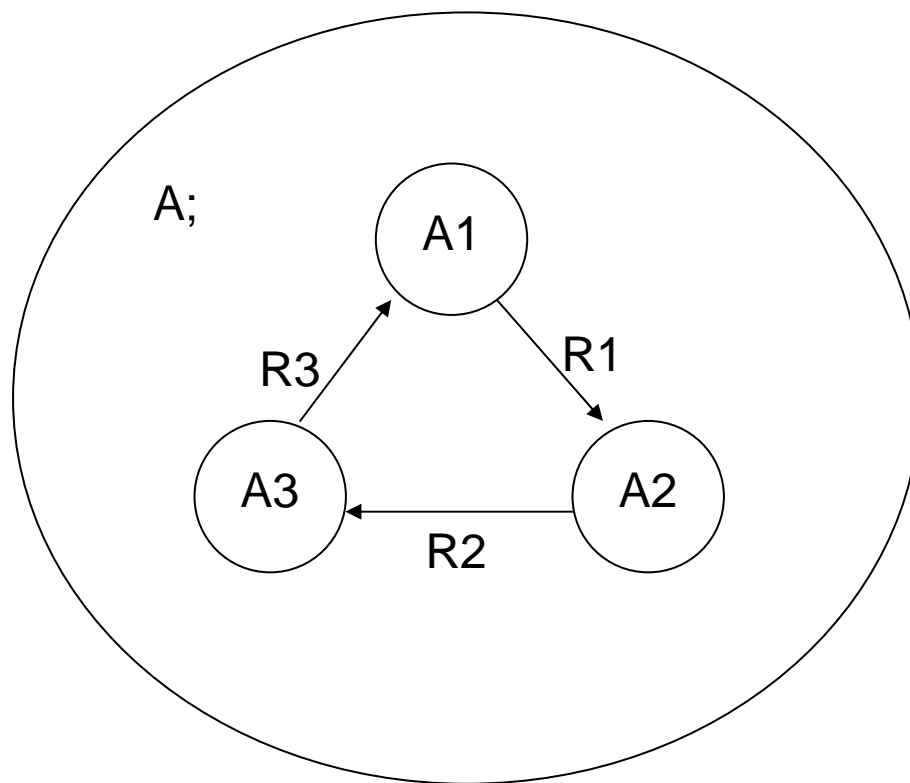
- 概念を定義するためのメタな概念記述
- 記述のためのメタな概念はCDL.coreの概念定義辞書にア prioriに定義

## – Concept Definition Dictionary (CDD)

- 実体 (Entity)、関係 (Relation)、属性 (Attribute) を最上位概念
- 論理的にはひとつの複合実体概念

# Model&Syntax

## グラフ表記



## テキスト表記

{#a A;

{#x1 A1;} {#x2 A2;} {#x3 A3;}

{#y1 R1;} {#y2 R2;} {#y3 R3;}

[#x1 #y1 #x2]

[#x2 #y2 #x3]

[#x3 #y3 #x1]}

## ビジュアル化表記

{#a A;

[{#a1 A1;}—{#r1 R1;}→{#a2 A2;}]

[#a2—{#r2 R2;}→{#a3 A3;}]

[#a3—{#r3 R3;}→#a1]}

## さらに

{#a A; [{#a1 A1;}—R1→A2—R2→A3—R3→#a1]}

# 複合概念による概念のモデル

- 複合概念全体という概念

<昨日コンピュータを購入したとの報告を受取った>

↓

{#A 事象;

  {#B 事象;

    <#b1;昨日>

    <#b2;コンピュータ>

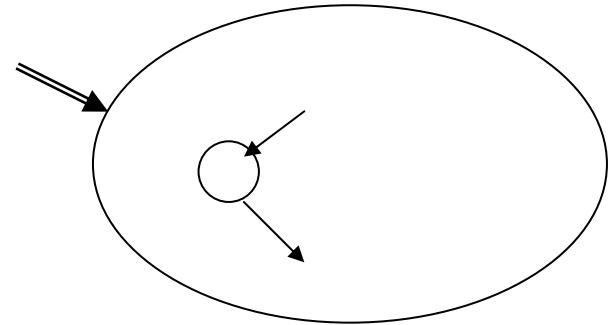
    <#b3;購入した>

    [#b3-obj→#b2, #b3-tim→#b1]}

  <#2;報告>

  <#3;受取った>

  [#2-cnt→#B, #3-obj→#2]}



- 複合概念によって修飾される概念

<昨日購入したコンピュータを設置した>

↓

{#A 事象;

{#B 事象;

<#b1;昨日>

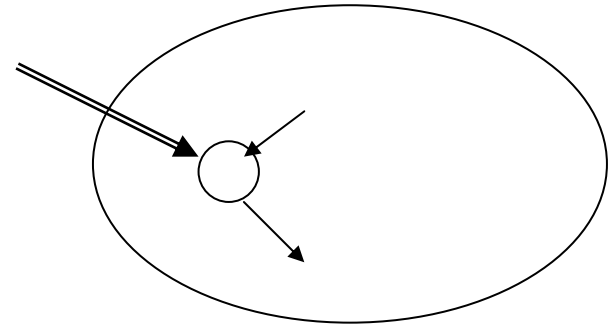
<#b2;コンピュータ>

<#b3;購入した>

[#b3-obj→#b2, #b3-tim→#b1]}

<#2;設置した>

[#2-obj→#b2]}



①コンピュータ#b2は事象#B、事象#Bは事象#A  
で修飾

```
{#A 事象;  
  {#B 事象;  
    <#b1;昨日>  
    <#b2;コンピュータ>  
    <#b3;購入した>  
    [#b3-obj→#b2, #b3-tim→#b1]}  
  <#2;設置した>  
  [#2-obj→#b2]}
```

③コンピュータ#b2は事象#A、事象#Aと#Bは  
状況#Xで修飾

```
{#X 状況;  
{#A 事象;  
  <#b2;コンピュータ>  
  <#2;設置した>  
  [#2-obj→#b2]}  
{#B 事象;  
  <#b1;昨日>  
  <#b3;購入した>  
  [#b3-obj→#b2, #b3-tim→#b1]}  
[#A-seq→#B]}
```

②コンピュータ#b2と事象#Bは共に事象#Aで修飾

```
{#A 事象;  
  {#B 事象;  
    <#b1;昨日>  
    <#b3;購入した>  
    [#b3-obj→#b2, #b3-tim→#b1]}  
  <#b2;コンピュータ>  
  <#2;設置した>  
  [#2-obj→#b2]}
```

④コンピュータ#b、事象#A、事象#Bは共に状況  
#Xで修飾

```
{#X 状況;  
<#b2;コンピュータ>  
{#A 事象;  
  <#2;設置した>  
  [#2-obj→#b2]}  
{#B 事象;  
  <#b1;昨日>  
  <#b3;購入した>  
  [#b3-obj→#b2, #b3-tim→#b1]}  
[#A-seq→#B]}
```

# 概念記述の例

<昨日購入したコンピュータを設置した>

↓

{#A 事象;

{#B 事象;

<#b1;昨日>

<#b2;コンピュータ>

<#b3;購入した>

<#b4;>

<#b5;>

[#b3 #b5 #b2] [#b3 #b4 #b1]}

<#2;設置した>

<#3;を>

[#2 #3 #b2]}

↓

↓  
{#A 事象;  
  {#B 事象;  
    {#b1 明鏡:さくじつ[昨日] ;}  
    {#b2 明鏡:コンピューター[computer];}  
    {#b3 明鏡:こうにゆう[購入]する tmp=past;}  
    {#tim tim;}  
    {#obj obj;}  
    [#b3 #obj #b2] [#b3 #tim #b1]}  
  {#2 明鏡:せっち[設置]する① tmp=past;}  
  [#2 #obj #b2]}

↓  
{#A 事象tmp=past;  
  {#B 事象tmp=past;  
    {#b1 明鏡:さくじつ[昨日] ;}  
    {#b2 明鏡:コンピューター[computer];}  
    {#b3 明鏡:こうにゆう[購入]する;}  
    [#b3 -obj→#b2, -tim→#b1]}  
  {#2 明鏡:せっち[設置]する①;}  
  [#2 -obj→#b2]  
  [#A -seq→#B]}

↓

```
{#A 事象tmp=past;  
  {#B 事象tmp=past;  
    明鏡:さくじつ[昨日]#b1  
    明鏡:コンピューター[computer]#b2  
    明鏡:こうにゆう[購入]する#b3  
    [#b3-obj→#b2, -tim→#b1]}  
    明鏡:せっち[設置]する①#2  
    [#2-obj→#b2]  
  [#A-seq→#B]}
```

