人工知能におけるオントロジーとその応用

武田英明
国立情報学研究所
takeda@nii.ac.jp
http://www-kasm.nii.ac.jp/~takeda/

目次

オントロジーとは

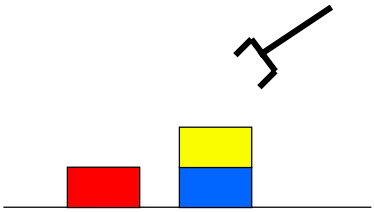
- オントロジー
  - 概念体系の提供
    - 計算機可読性
    - 人間理解可能性

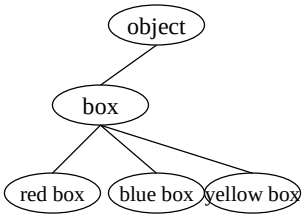
  人間と計算機をつなぐもの

オントロジーの定義

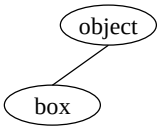
- Gruber
  - 概念化の明示的な仕様
- FIPA98
  - 特定のドメインの項像の明示的な仕様。
  - 対象領域を参照するための語彙（論理定数と述語記号）と領域に存在する制約の表現と語彙の解釈を制限する論理的言明。
  - あるトピックに関する知識の表現と通信のための語彙とその語彙で示される実在物(entity)の関係と属性の集合である。
- もっとわかり易くいえば
  - 共通の概念の体系（“語彙”とその定義とそれら間の関係）
  - *語彙以上OOのオブジェクト未満*

概念化



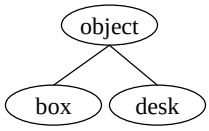


on_desk(A)
on(A, B)
put(A,B)



on_desk(A)
on(A, B)
put(A,B)

box
color:{red, blue, yellow}



on(A/box, B/object)
put(A/box,B/object)

box
color:{red, blue, yellow}

汎用性と目的依存性のトレードオフ

なぜ知識表現ではいけなかったのか

- 知識表現
  - 人工知能で人間の知識の表現手段として考案されたもの
  - 例：フレーム型，ルール型，ネットワーク型，論理型
- エキスパートシステム
  - 上記の方法で専門家の知識を記述して，専門家の行う作業を行うシステム
- エキスパートシステムの問題点
  - システムの拡張性
    - 新しい機能をつけくわえよう
  - メンテナンス性
    - 環境が変化したので変更しよう
  - 知識の問題
    - 知識の共有や再利用が困難

エキスパートシステムでの知識表現の問題点

- 普遍性への無配慮
  - 汎用性と効率性のトレードオフ
  - 別に普遍的な知識がほしい訳ではない
  - しかしそこへの配慮がないのでは困る
- ４つの無配慮
  - 全体性への無配慮
  - 網羅性への無配慮
  - 体系性への無配慮
  - 変化可能性への無配慮

普遍性への無配慮

- 全体性への無配慮
  - 目的外の視点は考慮されない
  - とある「リンゴ選別機」の知識
    - サイズ，色，硬さ，重量は知っている（センサがあるから）
    - ざらつき，採取者，採取からの日数（センサがなかった）
- 網羅性への無配慮
  - 対象外の事物は記述されない
  - とある「リンゴ選別機」の知識
    - 国産のりんごの種類は記述している
    - 外国産は知らない
- 体系性への無配慮
  - 背後にある概念との関係に関して無配慮
  - とある「リンゴ選別機」の知識
    - りんごが果物の一種であることは知らない

オントロジーの役割

- オントロジー：*普遍性のある*知識表現
  - 形式性，推論可能性
    - 知識表現の性質
  - 全体性（対象の包括的記述）
  - 網羅性（対象領域のカバー）
  - 体系性
  - 合意可能性
    - 目的の明示化による合意の可能性を示すこと
      - まったく目的のない表現はありえない
- オントロジーは表現や記述（だけ）の問題ではなく，その利用の仕方（作り方，使い方）において意味がある

オントロジーの形式的考察

- 何をもってオントロジーと呼んでいるのか
  1. Ontology as a philosophical discipline
  2. Ontology as a an informal conceptual system ➡ semantics（非形式的）
  3. Ontology as a formal semantic account ➡ semantics（形式的）
  4. Ontology as a specification of a conceptualization" ➡ semanticsとsyntaxの中間
  5. Ontology as a representation of a conceptual system via a logical theory ➡ syntax
     - 5.1 characterized by specific formal properties
     - 5.2 characterized only by its specific purposes
  6. Ontology as the vocabulary used by a logical theory
  7. Ontology as a (meta-level) specification of a logical theory
- とくに2-5

オントロジーの形式的考察

- **オントロジー**
  - **オントロジー制約** ontological commitment
  - **オントロジー理論** ontological theory

オントロジーの形式的考察

- **オントロジー理論** ontological theory
  - **特定の状況でなく成り立つ理論**(theory)
  - Syntax
  - **例**

T1:

$\forall x.\mathrm{apple}(x) \supset \mathrm{fruit}(x)$

$\forall x.\mathrm{pear}(x) \supset \mathrm{fruit}(x)$

$\mathrm{apple}(a1)$

$\mathrm{red}(a1)$

T2:

$\forall x.\mathrm{apple}(x) \supset \mathrm{fruit}(x)$

$\forall x.\mathrm{pear}(x) \supset \mathrm{fruit}(x)$

オントロジーの形式的考察

- **オントロジー制約** ontological commitment
  - **その概念がどのような性質（永続的なのかとか）を定める**
  - **一階述語論理からみると**semantics

T1:

$\forall x.\mathrm{apple}(x) \supset \mathrm{fruit}(x)$

$\forall x.\mathrm{pear}(x) \supset \mathrm{fruit}(x)$

$\mathrm{apple}(a1)$

$\mathrm{red}(a1)$

T3:

$\Box(\forall x.\mathrm{apple}(x) \supset \mathrm{fruit}(x))$

$\Box(\forall x.\mathrm{pear}(x) \supset \mathrm{fruit}(x))$

$\Box(\forall x.\mathrm{apple}(x) \supset \Box\ \mathrm{apple}(x))$

$\Box(\forall x.\mathrm{pear}(x) \supset \Box\ \mathrm{pear}(x))$

$\Box(\forall x.\mathrm{fruit}(x) \supset \Box\ \mathrm{fruit}(x))$

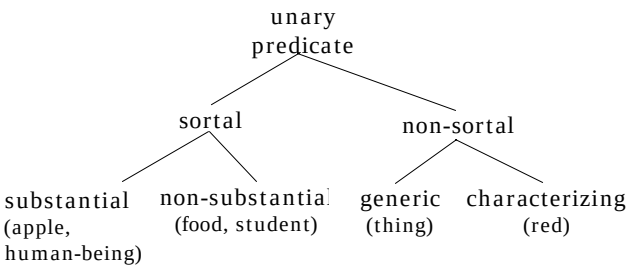
$\neg\Box(\forall x.\mathrm{red}(x) \supset \Box\ \mathrm{red}(x))$

Appleというものは存在すれば存在しつづけるものである

redというものは存在すれば存在しつづけるものではない

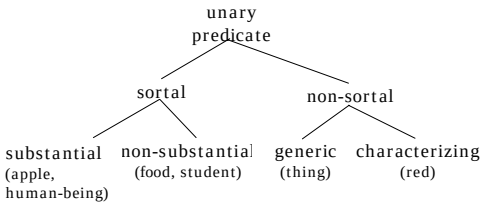
Universalsのオントロジー

- Sortal/non-sortal: sortal**とはその個体を峻別する原理を持つもの**
- substantial/non-substantial: sortal**の区分**
- generic predicate/characterizing predicate: non-sortal**の区分**



Universalsのオントロジー

- 自然であるnatural:
  - C|=∃x◇Px
- オントロジー的に固定されている (ontologically rigid)
  - C|=∀x(Px⊃□Px)
- 分割可能(divisive)
  - C|=∃x◇(Px∧∃y.y<x) ∧∀x□(Px⊃(∃y.(y<x⊃Py)))



- 一般的述語:rigidかつdivisive
- 実質的にsortal: rigidかつ非divisive
- 非実質的にsortal: 非rigidかつnaturalな述語で、ある実質的にsortalに包含される。
  - C|=∀x□(Px⊃Sx)
- 特徴づけ述語:非rigidかつnaturalな述語で、ある実質的にsortalに包含されない

オントロジーの実現

- **オントロジーのレベル**
- **オントロジー構築のガイドライン**
- **オントロジーの種類**
- **オントロジーの構築例**
- **オントロジー記述言語**

オントロジーの記述の３つのレベル（溝口）

- レベル　１： 概念の切り出しと階層的記述
  - 例：Yahooオントロジー，Metadataタグ
- レベル　２： 意味定義の明示化・形式化
  - 豊富な関係定義 公理の利用
  - 推論可能性 （自分自身）
  - 例：多くのオントロジー
- レベル　３： 構築されたモデル実行に関わる計算論
  - モデルの推論可能性
  - 例：タスクオントロジー

オントロジー開発のガイドライン

- オントロジー開発におけるいくつかの原則
  - 明瞭性と客観性
    - オントロジーは客観的な定義と言語による正確な意味をもった語彙を提供すべき
  - 完全性：定義はなるべく必要十分条件がよい
  - 一貫性：定義に無矛盾な推論ができること
  - 単調的拡張：既定義の用語を変更せずに新たな一般語や特殊語が導入できること
  - 最小のオントロジー・コミットメント：モデル対象をできるだけ少ない公理で表現すること

オントロジーのcriteria
(知識ベースとの比較）by溝口理一郎

(a)それはその分野の人々（エージェント）の合意した知識を表現したものか？

(b)人々はそれを正確に定義された用語として参照しているか？

(c)その表現に用いられている言語は人々が言いたいことを表現するのに十分な表現力を持っているか？

(d)それは複数の異なった問題解決に再利用可能か？

(e)それは安定しているか？

(f)それは，新しい知識ベース，データベーススキーマなどの複数のアプリケーションプログラムを開発するための出発点として利用可能か？

オントロジーの構成

- 対象オントロジー
  もの、ことを表現するオントロジー
  - Top-level オントロジー
  - 領域オントロジー
    - 例）機械のオントロジー、旅行のオントロジー
  - アプリケーション・オントロジー
- タスク・オントロジー
  行為やプロセスを表現するオントロジー

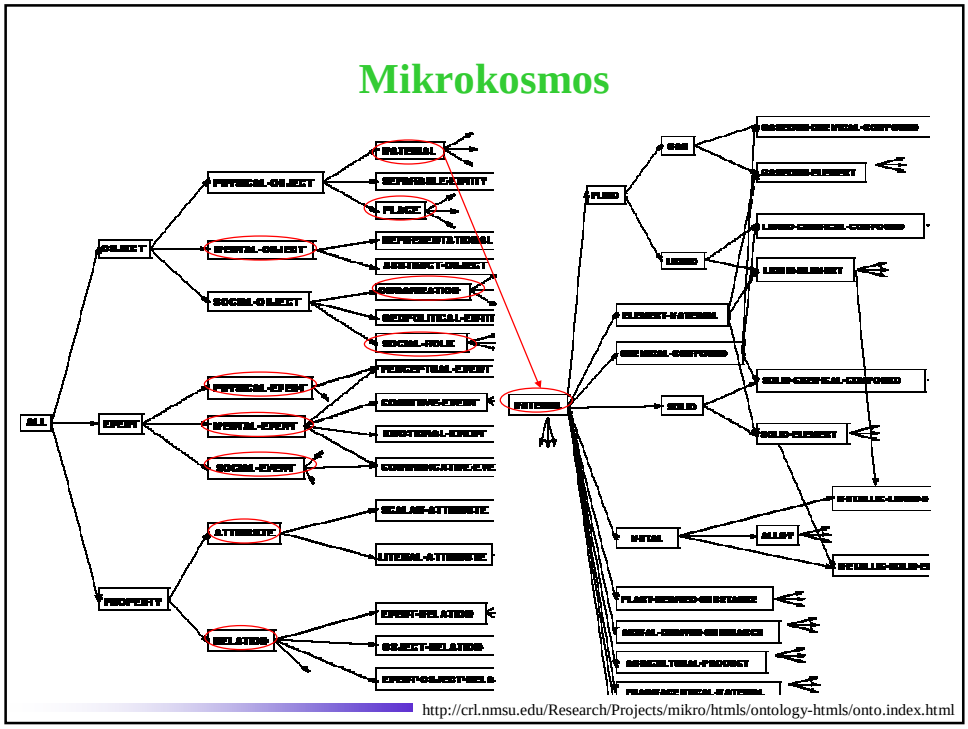
オントロジーの構成的定義

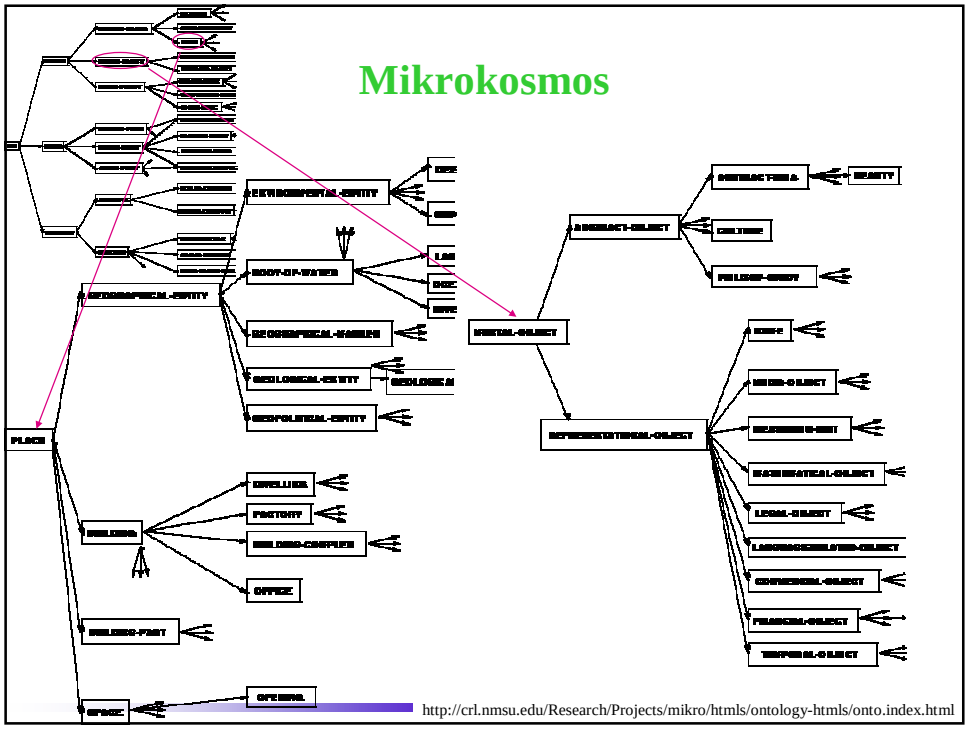
- 概念の切り出し
- タキソノミーの構築（階層構造の構築）
- 概念間の関係記述
- 形式化（公理化）

オントロジーの構築例

- Cyc
  - Top-level ontology**のみ公開．約**3000
- KSL ontology server
  - **誰でもオントロジーを作っておいておける**
- SENSUS
- ON9
- Mikrokosmos

- WordNet


Mikrokosmos
http://crl.nmsu.edu/Research/Projects/mikro/htmls/ontology-htmls/onto.index.html


Mikrokosmos
http://crl.nmsu.edu/Research/Projects/mikro/htmls/ontology-htmls/onto.index.html

オントロジー記述言語Ontolingua

「（文献の）著者」

```
(define-class AUTHOR (?author)
   "An author is a person who writes things.
    An author must have created at least one document.
    In this ontology, an author is known by his or her real name."
   :def (and (person ?author)
             (= (value-cardinality ?author author.name) 1)
             (value-type ?author author.name biblio-name)
             (>= (value-cardinality ?author author.documents) 1)
             (<=> (author.name ?author ?name)
                  (person.name ?author ?name))))
```

?authorが著者であるとは、?authorがpersonであり、関係author.nameで規定される、ちょうど１つの対象が存在し、それはクラスbiblio-nameのインスタンスでなければならず、author.documentsで関係づけられる少なくとも１つの対象が存在し、author.nameという関係とperson.nameという関係が同値であることである。

KIF (Knowledge Interchange Format)

- 1階述語論理を拡張して、項の定義、メタ知識(知識に関する知識)、集合、非単調理論などを記述できるようにした知識交換言語
- １階述語論理の表現をLisp風の記法で表現
- 拡張部分
  - 集合・リストの表現setof,やlistof
  - ifやcondによる条件記述子
  - defobject, deffunction, defrelationを使ってオブジェクト、関数、関係を定義

KIFによる表現の例

述語論理 / 数学の表記	KIF
(1) 項	
$x + y$	(+ ?x ?y)
{a, b, c, d}	(setof a b c d)
{x | wheel(x)∧ made-in-japan(x)} ;「日本製の車輪の集まり」	(setofall ?x (and (wheel ?x) (made-in-japan x)))
ι x [wheel(x) ∧ made-in-japan(x)] ;「その日本製の車輪」	(the ?x (and (wheel ?x) (made-in-japan x)))
κ x [wheel(x) ∧ made-in-japan(x)] ;「…は日本製の車輪である」	(kappa ?x (and (wheel ?x) (made-in-japan x)))
λ x [founded-year(vendor(x))] ;「x の製造者の設立年」	(lambda ?x (founded-year (maker ?x)))
if number(x) ∧ $x \neq 0$ then $1/x$;「x が 0 でない数のとき、$1/x$（を与える関数）」	(if (and (number ?x) (not (= ?x 0))) (/ 1 x))
(2) 命題	
p (a, b)	(p a b)
∀x∃y[product(x)→vendor(y)∧makes(y,x)]	(forall ?x (exists ?y (=> (product ?x) (and (vendor ?y) (makes ?y ?x)))))

OIL

- OIL =

  modelling primitives from frames *(OKBC-lite)*

+ semantics and inference from Description Logic

+ syntax from RDF(S) & XML(S)

OILのシンタックス

- Ontology Container: **オントロジー自身の情報**
  - title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, rights
- Ontology definitions
  - class-def
    - type: primitive(**必要条件**) or defined(**必要十分条件**)
    - name, documentation: string
    - subclass-of: class
    - slot-constraint
      - name, has-value, value-type, max-cardinality, min-cardinality, cardinality
  - slot-def
    - name, documentation: string
    - subslot-of: slot
    - domain, range: class
      - pair(x, y) -> domain x, range y
    - inverse slot
    - properties: transitive, symmetric
- import
- rule-base

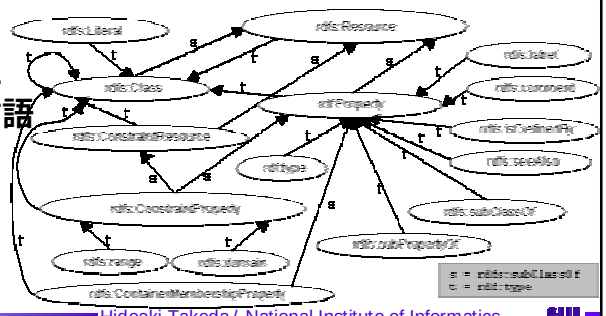
OILの例

```
ontology-container
        title "African animals"
        creator "Ian Horrocks"
        subject "animal, food, vegetarians"
        description "A didactic example ontology describing African animals"
        description.release "1.01"
        publisher "I. Horrocks"
        type "ontology"
        format "pseudo-xml"
        format "pdf"
        identifier "http://www.cs.vu.nl/~dieter/oil/TR/oil.pdf"
        source "http://www.africa.com/nature/animals.html"
        language "OIL"
        language "en-uk"
        relation.hasPart "http://www.ontosRus.com/animals/jungle.onto"
```

```
ontology-definitions
        slot-def eats
                inverse is-eaten-by
        slot-def has-part
                inverse is-part-of
                properties transitive
        class-def animal
        class-def plant
                subclass-of NOT animal
        class-def tree
                subclass-of plant
        class-def branch
                slot-constraint is-part-of
                        has-value tree
        class-def leaf
                slot-constraint is-part-of
                        has-value branch
        class-def defined carnivore
                subclass-of animal
                slot-constraint eats
                        value-type animal
        class-def defined herbivore
                subclass-of animal
                slot-constraint eats
                        value-type plant OR
                                (slot-constraint is-part-of
                                has-value plant)
        class-def herbivore
                subclass-of NOT carnivore
        class-def giraffe
                subclass-of animal
                slot-constraint eats
                        value-type leaf
        class-def lion
                subclass-of animal
                slot-constraint eats
                        value-type herbivore
        class-def tasty-plant
                subclass-of plant
                slot-constraint eaten-by
                        has-value herbivore carnivore
```

RDF and RDF Schema

- RDF(Resource Description Framework)
  - WWWのメタデータの記述
  - 属性／値の組
  - ラベル付きグラフ
  - XML serialization
    表記のため
- RDF Schema
  - 語彙の提供：簡単なオントロジー言語



OILとRDFS

- 語彙の対応と拡張

OIL primitive	RDFS syntax	type
class-def	rdfs:Class	class
subclass-of	rdfs:subClassOf	property
class-expression	*oil:ClassExpression* (placeholder only)	class
AND	oil:AND (subclass of ClassExpression)	class
OR	oil:OR (subclass of ClassExpression)	class
NOT	oil:NOT (subclass of ClassExpression)	class
slot-constraint	*oil:SlotConstraint* (placeholder only) oil:hasSlotConstraint (rdf:type of rdfs:ConstraintProperty) *oil:NumberRestriction* (placeholder only) (subclass of oil:SlotConstraint)	class property class

OIL primitive	RDFS syntax	type
has-value	oil:HasValue (subclass of oil:SlotConstraint)	Class
value-type	oil:ValueType (subclass of oil:SlotConstrain)t	class
max-cardinality	oil:MaxCardinality (subclass of oil:NumberRestriction)	class
min-cardinality	oil:MinCardinality (subclass of oil:NumberRestriction)	class
cardinality	oil:Cardinality (subclass of oil:NumberRestriction)	class

OIL primitive	RDFS syntax	type
slot-def	rdf:Property	class
subslot-of	rdfs:subPropertyOf	property
domain	rdfs:domain	property
range	rdfs:range	property
inverse	oil:inverseRelationOf	property
transitive	oil:TransitiveRelation	class
symmetric	oil:SymmetricRelation	class

OIL by RDFS

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-
19990303#"
xmlns:oil="http://www.ontoknolwedge.org/oil/rdfschema"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcq="http://purl.org/dc/qualifiers/1.1/"
xmlns:ont2="http://www.semanticweb.org/oil/ontology-
example">
<rdf:Description about = "">
<dc:Title>"African Animals"</dc:Title>
<dc:Creator>"Ian Horrocks"</dc:Creator>
<dc:subject >"animal, food, vegetarians"</dc:subject >
<dc:Description>"A didactic example ontology describing
African animals"</dc:Description>
<dcq:description.release> "1.01"</dcq:description.release>
<dc:publisher> "I. Horrocks"</dc:publisher>
<dc:type> "ontology"</dc:type>
<dc:format> "oil"</dc:format>
<dc:format> "pdf"</dc:format>
<dc:identifier>
"http://www.cs.vu.nl/~dieter/oil/TR/oil.pdf"</dc:identifier>
<dc:source>
"http://www.africa.com/nature/animals.html"</dc:source>
<dc:language> "OIL"</dc:language>
<dc:language> "en-uk"</dc:language>
</rdf:Description>
<rdf:Description
xmlns:syllogism="http://old.greece/syllogism/">
<rdf:type
resource="http://www.ontoknowledge.org/oil//rdfschema#Rul
eBase"/>
<syllogism:premise>if it rains, you get
wet</syllogism:premise>
<syllogism:fact>it rains</syllogism:fact>
<syllogism:conclusion>you get wet</syllogism:conclusion>
</rdf:Description>
```

```
<rdf:Property rdf:ID="eats">
  <oil:inverseRelationOf rdf :resource="#is -eaten-by"/>
</rdf:Property>
<rdf:Property rdf:ID="is-eaten-by"/>
  <rdf:Property rdf:ID="has-part">
    <oil:inverseRelationOf rdf :resource="#is -part-of"/>
  </rdf:Property>
  <rdf:Property rdf:ID="is-part-of"/>
<rdfs:Class rdf:ID="animal"/>
<rdfs:Class rdf:ID="plant">
  <rdfs:subClassOf >
    <oil:NOT>
      <oil:hasOperand rdf :resource="#animal"/>
    </oil:NOT>
  </rdfs:subClassOf >
</rdfs:Class>
<rdfs:Class rdf:ID="tree">
  <rdfs:subClassOf rdf :resource="#plant"/>
</rdfs:Class>
<rdfs:Class rdf:ID="branch">
  <oil:hasSlotConstraint >
    <oil:has-value>
      <oil:hasProperty rdf :resource="#is -part-of"/>
      <oil:hasClass rdf :resource="#tree"/>
    </oil:has-value>
  </oil:hasSlotConstraint >
</rdfs:Class>
<rdfs:Class rdf:ID="leaf">
  <oil:hasSlotConstraint >
    <oil:has-value>
      <oil:hasProperty rdf :resource="#is -part-of"/>
      <oil:hasClass rdf :resource="#branch"/>
    </oil:has-value>
  </oil:hasSlotConstraint >
</rdfs:Class>
<rdfs:Class rdf:ID="carnivore">
  <rdfs:subClassOf rdf :resource="#animal"/>
  <oil:hasSlotConstraint >
    <oil:valueType >
      <oil:hasProperty rdf :resource="#eats"/>
      <oil:hasClass rdf :resource="#animal"/>
    </oil:valueType >
  </oil:hasSlotConstraint >
</rdfs:Class>
```

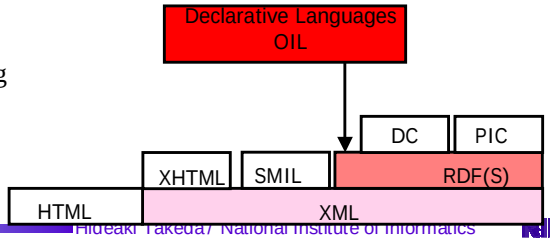


```
ontology-definitions
    slot-def eats
        inverse is-eaten-by
    slot-def has-part
        inverse is-part-of
        properties transitive
    class-def animal
    class-def plant
        subclass-of NOT animal
    class-def tree
        subclass-of plant
    class-def branch
        slot-constraint is-part-of
            has-value tree
    class-def leaf
        slot-constraint is-part-of
            has-value branch
    class-def defined carnivore
        subclass-of animal
        slot-constraint eats
            value-type animal
    class-def defined herbivore
        subclass-of animal
        slot-constraint eats
            value-type plant OR
                (slot-constraint is-part-of
                has-value plant)
    class-def herbivore
        subclass-of NOT carnivore
    class-def giraffe
        subclass-of animal
        slot-constraint eats
            value-type leaf
    class-def lion
        subclass-of animal
        slot-constraint eats
            value-type herbivore
    class-def tasty-plant
        subclass-of plant
        slot-constraint eaten-by
            has-value herbivore, carnivore
```

```
<rdf:Property rdf:ID="eats">
  <oil:inverseRelationOf rdf :resource="#is -eaten-by"/>
</rdf:Property>
<rdf:Property rdf:ID="is-eaten-by"/>
  <rdf:Property rdf:ID="has-part">
    <oil:inverseRelationOf rdf :resource="#is -part-of"/>
  </rdf:Property>
  <rdf:Property rdf:ID="is-part-of"/>
<rdfs:Class rdf:ID="animal"/>
<rdfs:Class rdf:ID="plant">
  <rdfs:subClassOf >
    <oil:NOT>
      <oil:hasOperand rdf :resource="#animal"/>
    </oil:NOT>
  </rdfs:subClassOf >
</rdfs:Class>
<rdfs:Class rdf:ID="tree">
  <rdfs:subClassOf rdf :resource="#plant"/>
</rdfs:Class>
<rdfs:Class rdf:ID="branch">
  <oil:hasSlotConstraint >
    <oil:has-value>
      <oil:hasProperty rdf :resource="#is -part-of"/>
      <oil:hasClass rdf :resource="#tree"/>
    </oil:has-value>
  </oil:hasSlotConstraint >
</rdfs:Class>
<rdfs:Class rdf:ID="leaf">
  <oil:hasSlotConstraint >
    <oil:has-value>
      <oil:hasProperty rdf :resource="#is -part-of"/>
      <oil:hasClass rdf :resource="#branch"/>
    </oil:has-value>
  </oil:hasSlotConstraint >
</rdfs:Class>
<rdfs:Class rdf:ID="carnivore">
  <rdfs:subClassOf rdf :resource="#animal"/>
  <oil:hasSlotConstraint >
    <oil:valueType >
      <oil:hasProperty rdf :resource="#eats"/>
      <oil:hasClass rdf :resource="#animal"/>
    </oil:valueType >
  </oil:hasSlotConstraint >
</rdfs:Class>
```

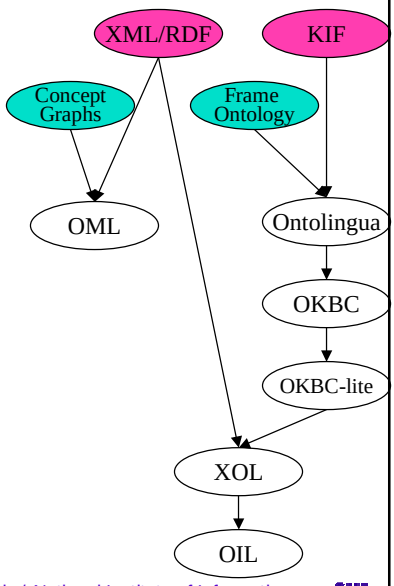
OIL:

- Currently available tools
  - Definition of language
    - semantics
    - XML encoding
    - RDF encoding
  - Tools:
    - Translators (XSL based)
    - reasoner (FaCT, DL-based)
    - OntoEdit
  - case-studies
    - GIS ontology mapping
    - (KA)2 ontology
    - CIA world fact book



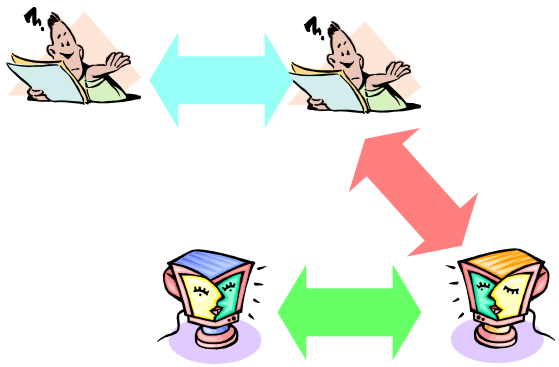
オントロジーの記法

- KIF(Knowledge Interchange Format)
  - **一階論理ベース**
- Ontolingua
  - KIF**ベース**
- OKBC(Open Knowledge Base Connectivity)
  - Ontolingua+frame ontology
- XOL(XML Ontology Exchange Language)
  - OKBC-lite**のセマンティックス**
  - XML**のシンタックス**
- OIL(Ontology Interchange Language)
  - XOL**ベース**
- OML(Ontology Markup Language)
  - SHOE
  - Conceptual Graphs**のセマンティックス**

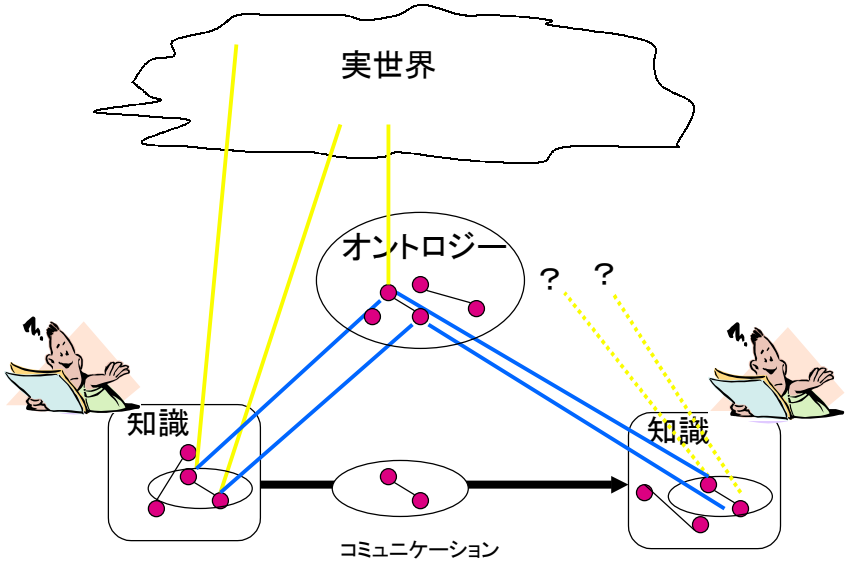


オントロジーの利用

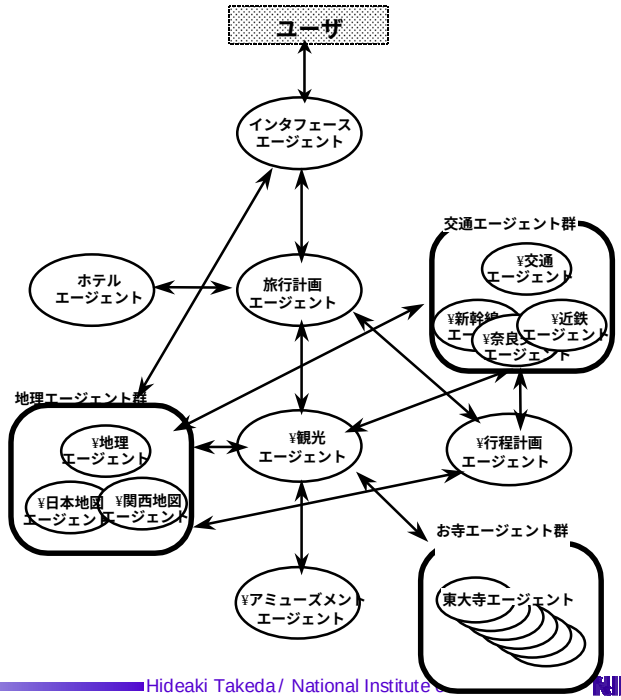
- エージェント間コミュニケーションのためのオントロジー
- 人間とシステムの間のコミュニケーションのためのオントロジー
- 人間と人間の間のコミュニケーションのためのオントロジー



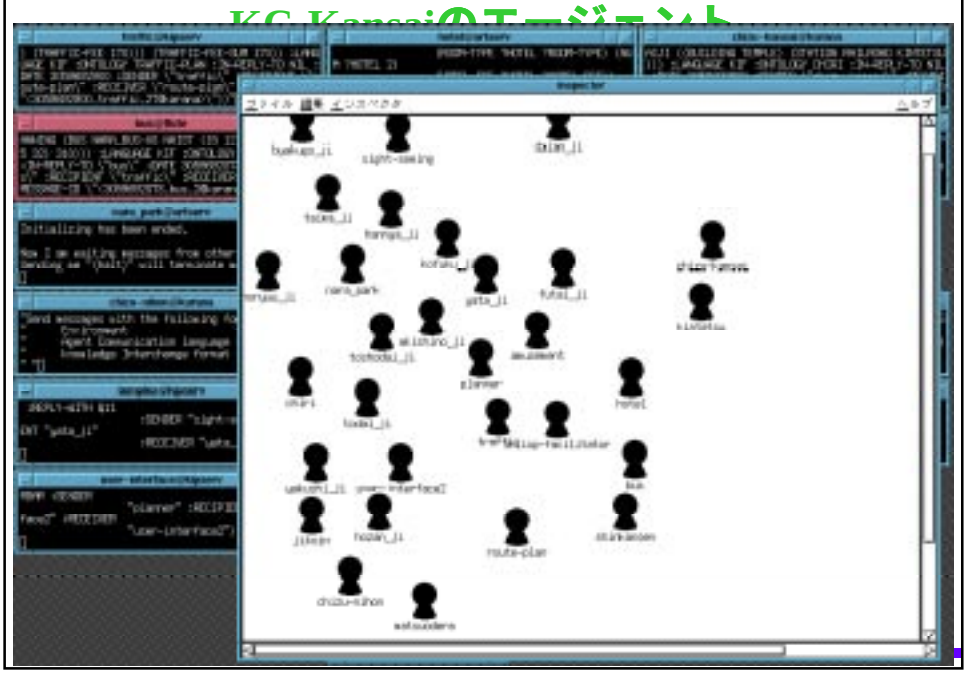
エージェントor人間間でのオントロジーの役割

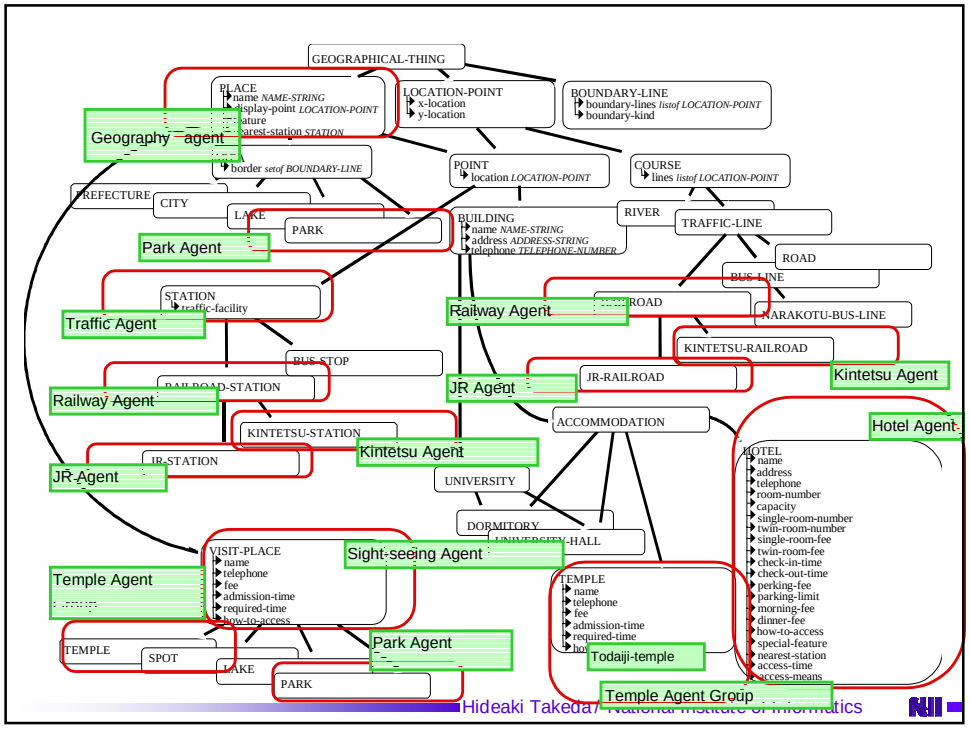


KC-Kansai



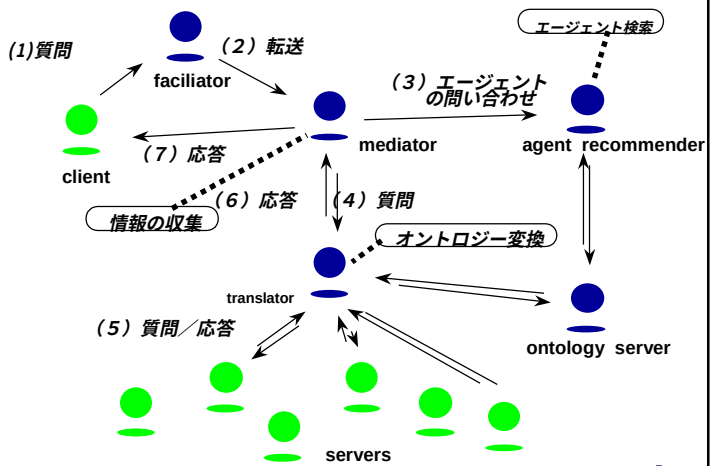
KC-Kansaiのエージェント





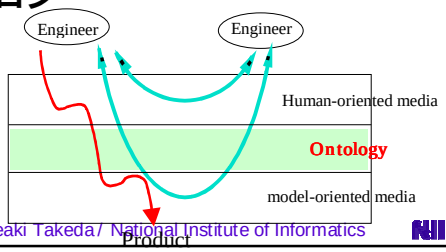
KC０での利用

- メッセージのアスペクト間変換をTranslator agentとして実現
- Mediatorは必要時にtranslatorを呼ぶ出すことでメッセージの伝達を行なう。



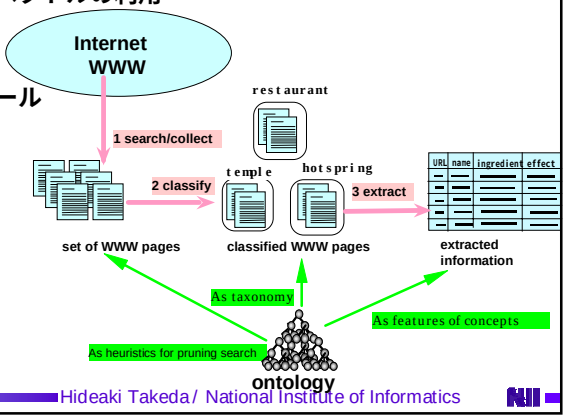
中間メディアとしてのオントロジー

- 計算機向きメディアと人間向きメディアのギャップ
  - 計算機向きメディア
    - CAD/CAM/simulation/図面
    - 明瞭な構文と意味
  - 人間向きメディア
    - 資源言語文、図、表, …
    - 曖昧な構文と意味
- 中間メディアとしてのオントロジー
  - 人間に理解可能
  - 機械に容易に処理可能

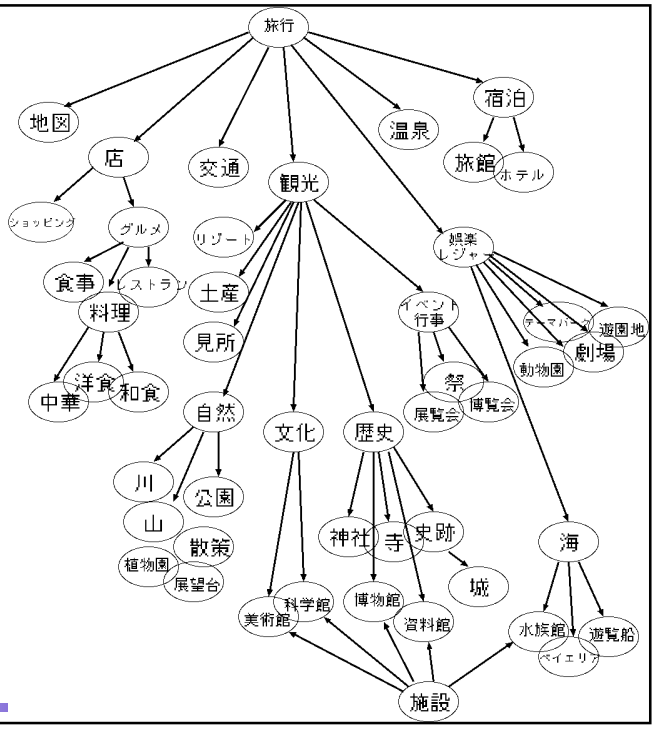


IICA

- 情報収集
  - 関連する概念を含むページだけを収集
  - オントロジーの概念でフィルタリング
- 情報分類
  - 概念ごとにクラスタリング
  - 概念クラスを表現する特徴ベクトルの利用
- 情報抽出
  - 状態遷移図
  - 出現パターンを記述するルール



オントロジー



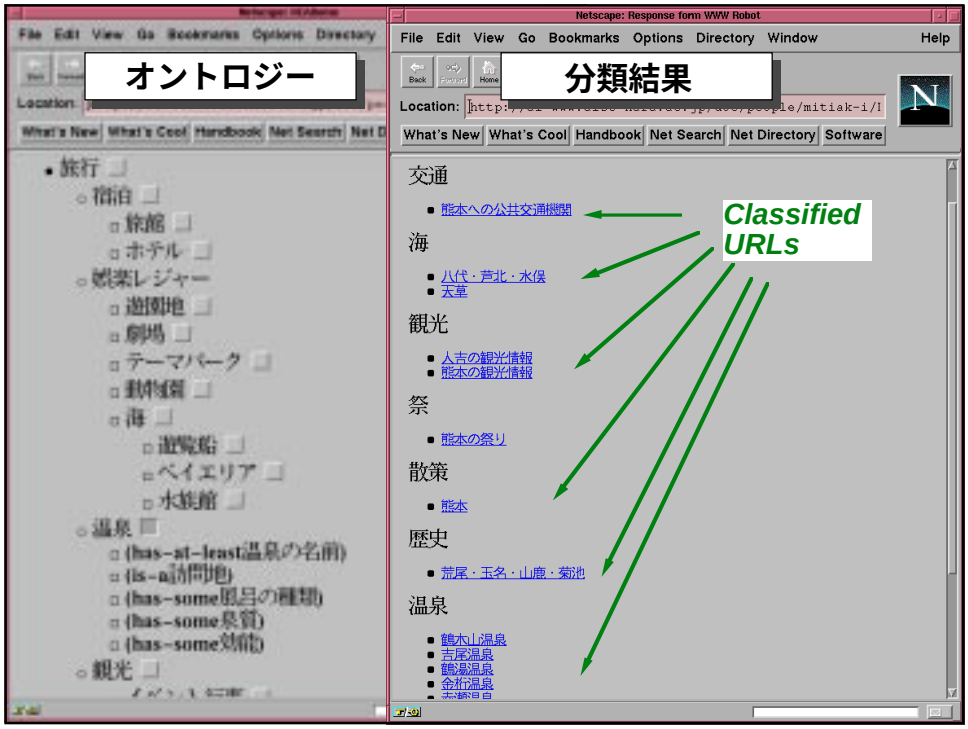
IICAにおけるパターンルール

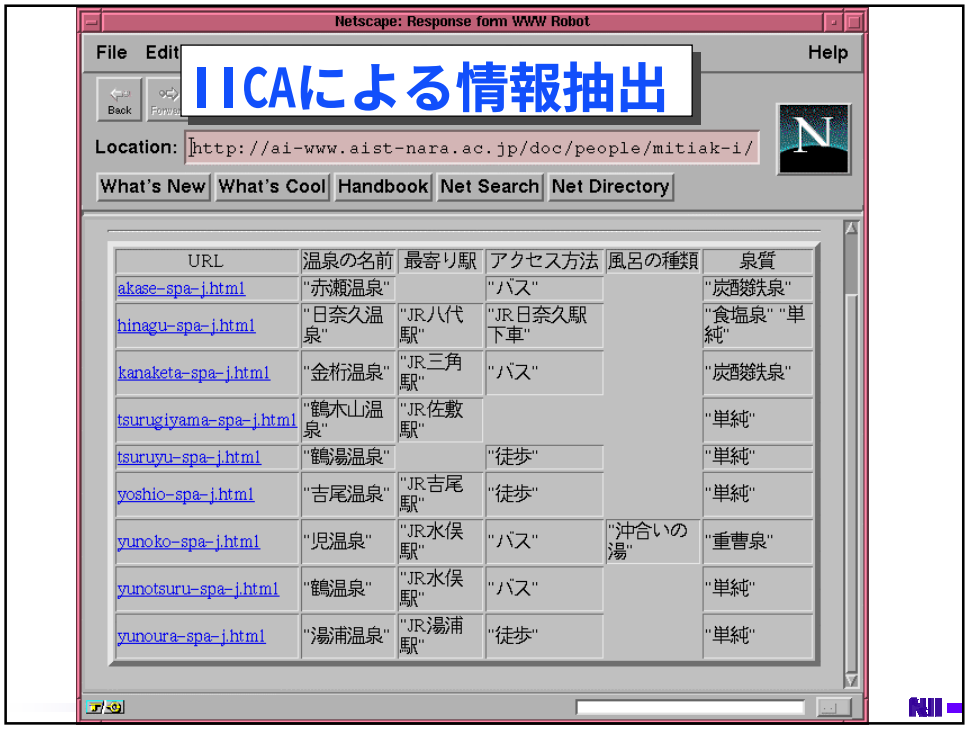
- **クラス概念定義**

(**define-pclass**
(hot-spring
((**has-at-least** name)
(**has-some** type-of-bath)
(**has-some** ingredient)
(**has-some** effect))))

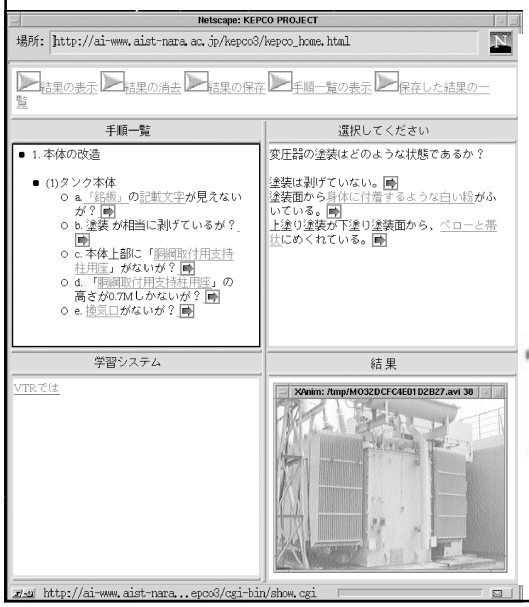
- **属性概念定義**

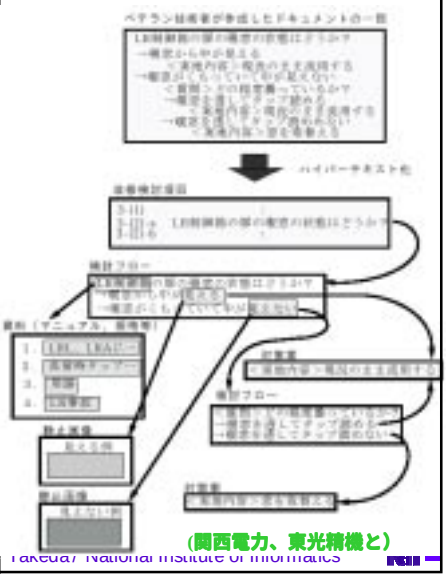
(**define-concept**
(effect
(**is** disease **with** (or “効能>”“効果”“効く”))))
(**define-concept**
(disease (is (**or** “+病>” “+傷” “+痛”))))


オントロジー
分類結果
Netscape: Response form WWW Robot
File Edit View Go Bookmarks Options Directory Window Help
What's New What's Cool Handbook Net Search Net Directory Software
Location:
旅行
宿泊
旅館
ホテル
娯楽レジャー
遊園地
劇場
テーマパーク
動物園
海
遊覧船
ベイエリア
水族館
温泉
(has-at-least温泉の名前)
(is-a訪問地)
(has-some風呂の種類)
(has-some泉質)
(has-some効能)
観光
交通
熊本への公共交通機関
海
八代・芦北・水俣
天草
観光
人吉の観光情報
熊本の観光情報
祭
熊本の祭り
散策
熊本
歴史
荒尾・玉名・山鹿・菊池
温泉
鶴木山温泉
吉尾温泉
鶴湯温泉
金桁温泉
Classified URLs


Netscape: Response form WWW Robot
File Edit Help
IICAによる情報抽出
Back
Location: http://ai-www.aist-nara.ac.jp/doc/people/mitiak-i/
What's New What's Cool Handbook Net Search Net Directory
URL 温泉の名前 最寄り駅 アクセス方法 風呂の種類 泉質
akase-spa-j.html "赤瀬温泉" "バス" "炭酸鉄泉"
hinagu-spa-j.html "日奈久温泉" "JR八代駅" "JR日奈久駅下車" "食塩泉" "単純"
kanaketa-spa-j.html "金桁温泉" "JR三角駅" "バス" "炭酸鉄泉"
tsurugiyama-spa-j.html "鶴木山温泉" "JR佐敷駅" "単純"
tsuruyu-spa-j.html "鶴湯温泉" "徒歩" "単純"
yoshio-spa-j.html "吉尾温泉" "JR吉尾駅" "徒歩" "単純"
yunoko-spa-j.html "児温泉" "JR水俣駅" "バス" "沖合いの湯" "重曹泉"
yunotsuru-spa-j.html "鶴温泉" "JR水俣駅" "バス" "単純"
yunoura-spa-j.html "湯浦温泉" "JR湯浦駅" "徒歩" "単純"

現場技術情報共有の知的支援





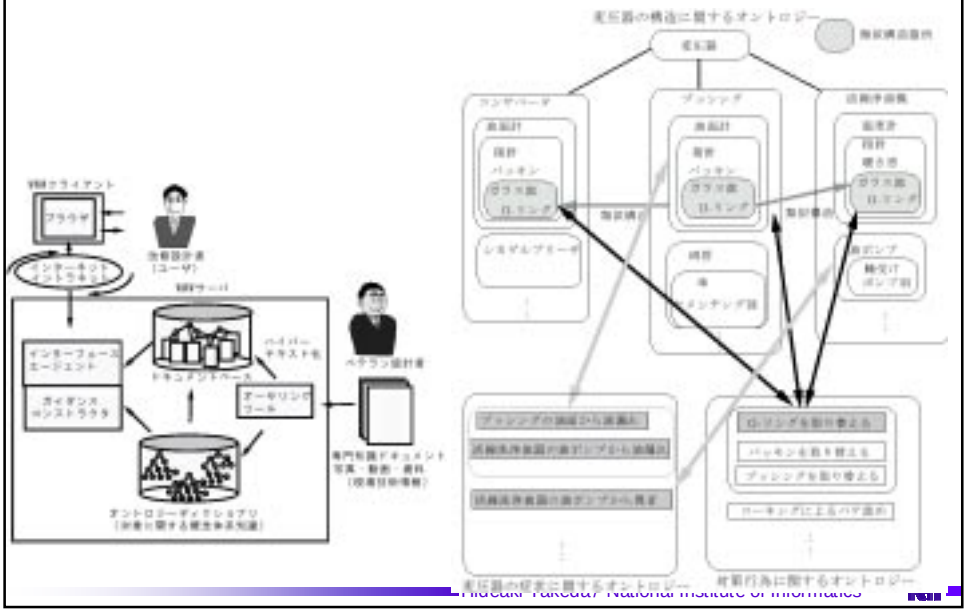
(関西電力、東光精機と)

オントロジー

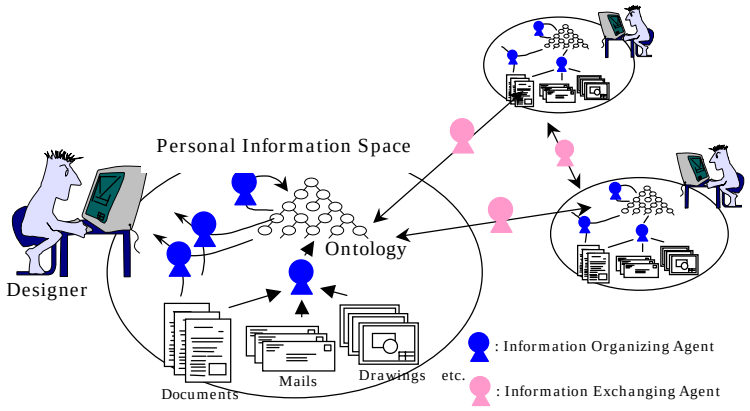
点検診断行為に関する知識の種類	専門家が用いる語彙知識
1. 点検行為に関するメタな知識	「計画を見直す」，「～を使いきる」，「～を変更しない」，「現状のままとする」，「～をやめる」，「～を完了する」，「～を手配する」，「～を充当する」
2. 対象の位置・構造の変化に関わる操作	「～を検知する」，「～を設置する」，「～を設置しない」，「～を取付ける」，「～を取り外す」，「～を調整で繰ぎ足す」，「～を引きおろす」，「一構造にする」，「～を取替える」，「～を取替えない」，「～を交換する」，「～を変更する」，「～を変える」，「新たな環境にする」
3. 対象の位置・構造の変化に関わる操作	「～を(補修)塗装する」，「～を塗る」，「～を塗す」，「グリス・アップする」，「を洗う」，「～に注水する」，「～を取り除く」，「～を拭き落とす」，「～を脱る」，「～を揺らす」，「～を回転中する」

症状に関する知識の種類	専門家が用いる語彙知識
1. 対象の可視性	「～が見えない」，「～が見にくい」，「～を目視確認ができない」
2. 対象の有無	「～がある」，「～が無い」，「～が設置されていない」，「～が要る」
3. 対象の変化・変質	「～が変色している」，「～が生じている」「～が硬質化している」，「～から油漏れしている」，「～から油がにじんでいる」「～が欠損している」，「～は(を)どうする」，「～が(で)汚れているが」，「～になっている」，「～が剥げている」，「～の汚れがある」，「～してある」
4. 対象の位置・距離	「～の方向を向いている」，「～の直前になる」「～の高さにない」，「～の絶縁距離(は良いか)」，「～～通れない」，「～から外れている」「～がずれている」
5. 対象の形式・型番	「～型になっている」，「～式であるが」「～に適合しない」，「～と違う」，「～が使用されている」，「～でない」
6. 対象の量	「～が高い」「～が低い」，「～を調整しない」「～が不足している」「～が足りない」「～が多い」，「～の値が悪い」，「～に近い」
7. 対象からの音	「～から異音がする」

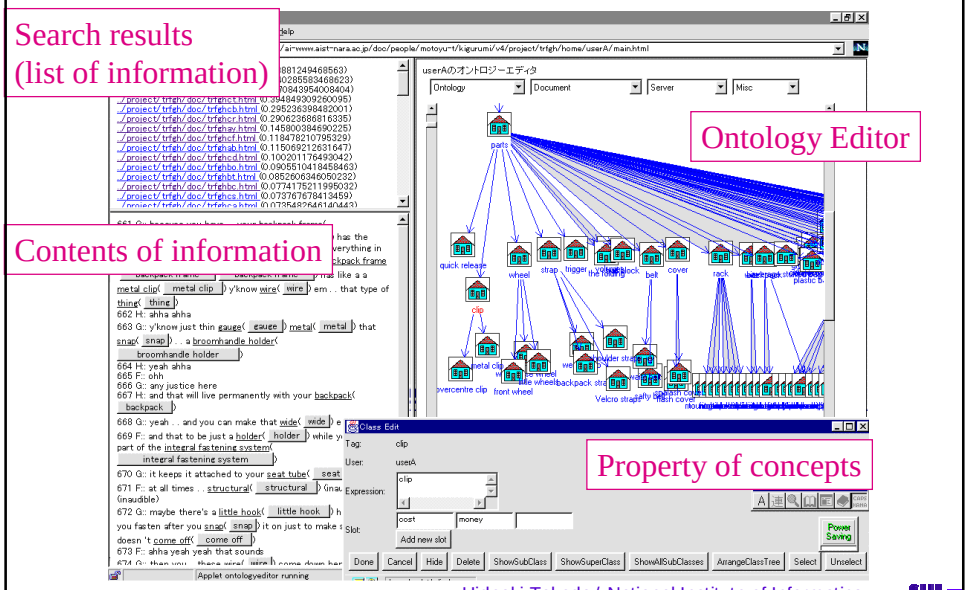
オントロジーを利用した検索



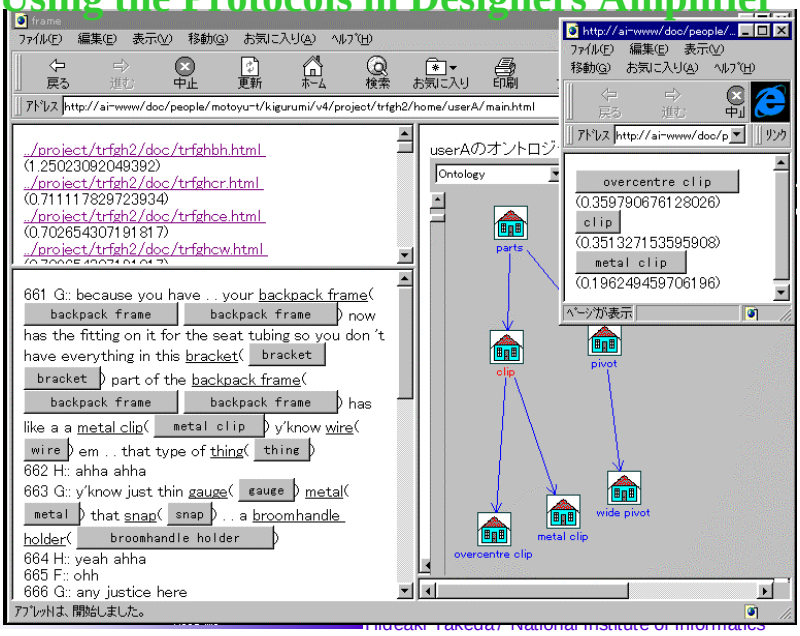
Architecture of Designers Amplifier



Designers Amplifier

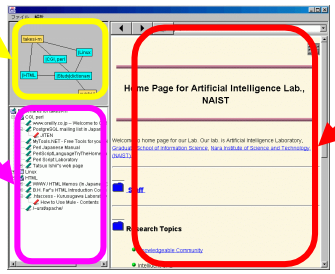


Using the Protocols in Designers Amplifier



共有情報発見によるコミュニティ形成支援システム：kMedia

- インターネット上の有益な情報を共有
- 情報整理の支援・促進
- コミュニティ形成の支援



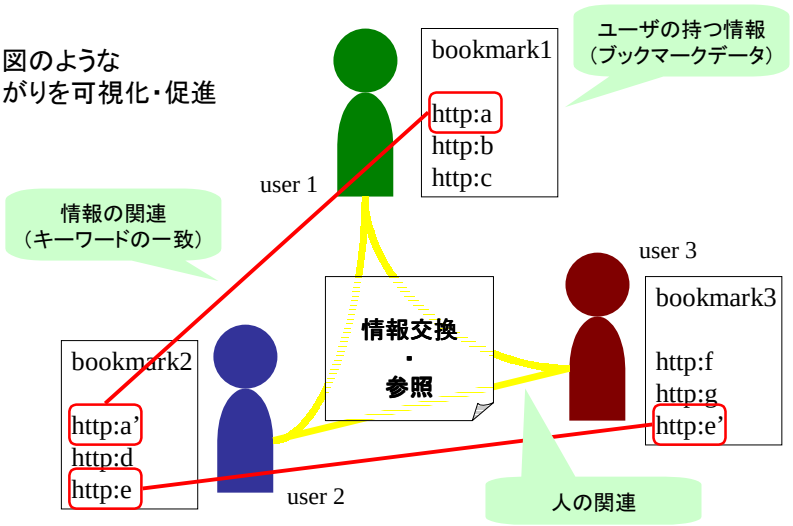
ブラウザのブックマークデータを各人が持つ知識とし、このデータの推薦・整理・関連づけを行うことによって目的を達成

ブックマークデータを用いることの利点

- ◆データが既存であり新規作成も容易
- ◆扱いが容易（ユーザ・システムの両方）
- ◆カテゴリ構造が一種の知識構造になっている（いろいろな用途に流用可能）

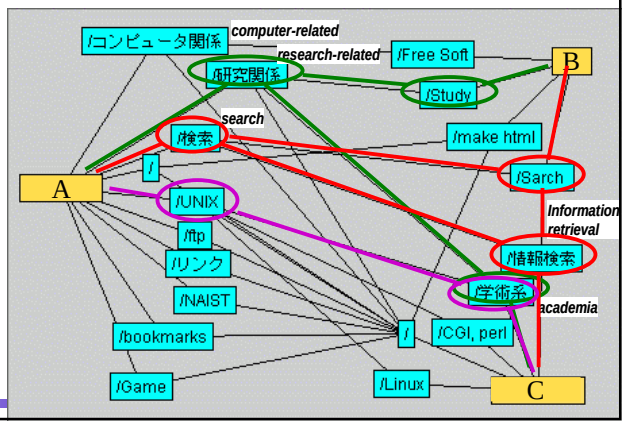
システム概念図

右図のような
繋がりを可視化・促進



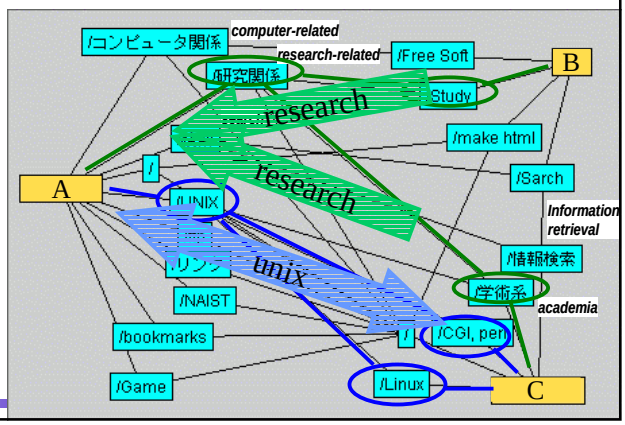
トピック関係の発見

- 共通性のある関係の発見
  - (search, IR), (academia, research-related)
  - 表現の違いを吸収
- コミュニティ特有の関係の発見
  - (Unix, academia)
  - そのコミュニティの特徴を表現



人間同士の関係の発見

- 他のメンバーとどんな共通の話題をもっているか？
- 誰がどんなことが得意か？



まとめ

- オントロジー
  - 概念体系の提供
    - 計算機可読性
    - 人間理解可能性

  人間と計算機をつなぐもの
- オントロジーに関する研究
  - まだ未発展（永遠に未発展？）
  - 実践が理論化を進展させる
    - ~~オントロジーの基礎理論~~
    - ~~オントロジー記述言語~~
    - ~~オントロジー構築~~
    - ~~オントロジーの利用~~

    - オントロジーの利用
    - オントロジー構築
    - オントロジー記述言語
    - オントロジーの基礎理論