

# 自律ソフトウェアとは？

- 自律性：(ソフトウェア)エージェントの重要な特徴の1つ
  - [Franklin and Graesser '96]の定義
    - 環境とのインタラクション(環境への影響を含めて)
    - 自分自身の予定(agenda, 意図, 目的)を追求
  - つまり, 自分の目的達成のために, 環境とインタラクションを行うシステム(あるいはソフトウェア)

# エージェントとは？

- 研究者により多様な定義・説明
- アラン・ケイ ([Kay '84]) : 計算機の世界の中に住んで仕事を行う「ソフトロボット」
- [Crowston and Malone '88] : メッセージの集合の全体, または一部を自動的に処理することにより, 人間の協調プロセスを支援
- [Maes '94] : ユーザと同じ作業環境で協調するパーソナルアシスタント

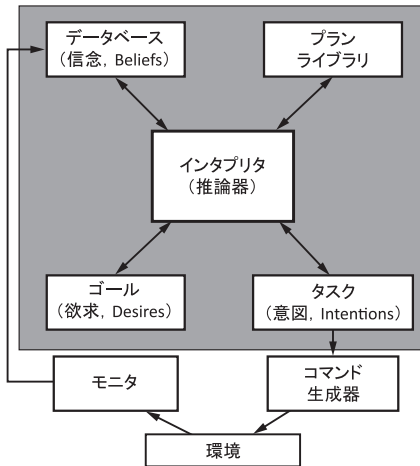
# 背景

- 多数の自律ソフトウェアの開発・提案  
→問題点の認識や知見の蓄積が進行
  - 問題点：構造や振舞いが複雑
    - オブジェクト指向+人工知能技術
    - 環境とのインタラクション→環境は予期せぬ変化を起こしうる
  - 知見：多くの共通概念・理論・技術
    - エージェントアーキテクチャ：BDIなど
    - エージェント間インタラクション：言語行為論, KQMLなど
    - エージェントの特性：リアクティブ・プロアクティブ, 知性, 移動性, 学習, 協調, 適応, 創発など
- エージェント技術による自律ソフトウェアの枠組みとして  
体系化

# BDI アーキテクチャ

- 個々のエージェントのアーキテクチャ
- M. Bratman, P.R. Cohen, H.J. Levesque, M.P. Georgeff らが提唱
- BDI: Belief, Desire, Intention(信念, 欲求, 意図)
  - Belief : (各時点で)エージェントが信じている(環境や自分自身の)状態
  - Desire : エージェントが達成したいと思っている状態
  - Intention : Desireを達成するために実際に実行する行動

# BDI アーキテクチャ



# エージェント間インタラクション

- 理論的アプローチ: 言語行為論 (Speech Act Theory)
  - 哲学の分野で, J.L. Austin, J.R. Searle らが提唱
  - 古くからエージェント間インタラクションに適用 ([Cohen and Levesque '85]など)
  - 言葉の意味に含まれる, 話者から聞き手に対して意図される行為に関する理論
  - 言語行為はパフォーマティブ (performative) と呼ばれる種類に分けられる
    - パフォーマティブ (の大分類) の例: 問合せ, 応答, 情報伝達

# エージェント間インタラクション

- 言語行為論の適用例：KQML
  - Knowledge Query Manipulation Language
  - エージェント間で交換するメッセージの記述言語
  - T. Finin らが提唱
  - 構文：( パフォーマティブ[:キーワード式]\*)
    - キーワードと式の対のリストは引数
  - ファシリテータエージェントのためのパフォーマティブ

# KQML 記述例

パフォーマンス  
(回答を1回だけ要求)

```
1: (ask-one
2:   :content (PRICE IBM ?price)
3:   :receiver stock-server
4:   :language LPROLOG
5:   :ontology NYSE-TICKS)
```

変数

→ IBM 社の株価を1回だけ要求



# 背景

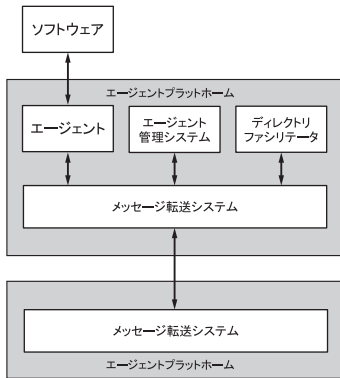
- 1990年代中頃までに、多数のエージェントシステムが登場
  - しかし、大規模なものは少数
  - 一方、大規模な(非エージェント)システムの開発においては、開発方法論の必要性が広く知られていた
  - 開発方法論：開発手法・技術の知識やプラクティスを体系化したもの
    - 例：構造化手法, オブジェクト指向
- エージェント指向開発方法論が登場

# ソフトウェア開発方法論の概要

- ソフトウェア開発工程に対応した部分
  - 5工程：要求, 設計, 実装, テスト・デバッグ, 運用・保守
  - 要求と設計を合わせて上流工程と呼ぶ
    - 上流工程に焦点を当てた方法論が多数
- 開発工程全体にわたる部分
  - 開発管理：開発体制や進捗状況などの管理
    - SWEBOK では構成管理を別項目としている
  - プロセス：開発手順の構成と評価
  - モデルと手法
  - 品質

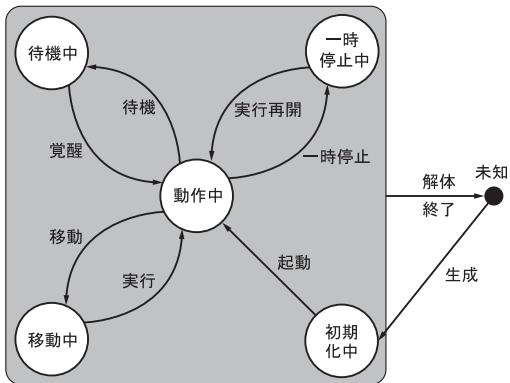
# エージェント管理仕様

- 参照モデル：エージェント管理に関するシステム構成



# エージェント管理仕様

- エージェントライフサイクル



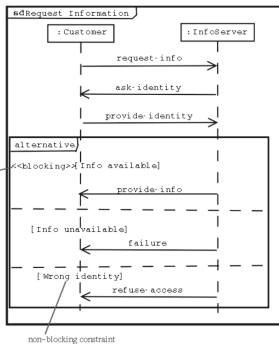
# エージェント間通信言語 FIPA ACL

- 言語行為論に基づくメッセージ記述言語
- おもな構成要素(パラメータ)
  - パフォーマティブ
  - 送受信者
  - メッセージ内容と関連情報(記述言語, オントロジーなど)
  - プロトコル
  - メッセージ ID 情報
- おもなパフォーマティブ
  - request, request-when, request-whenever : 動作の依頼
  - query-if, query-ref : 問合せ
  - inform, inform-if, inform-ref : 通知

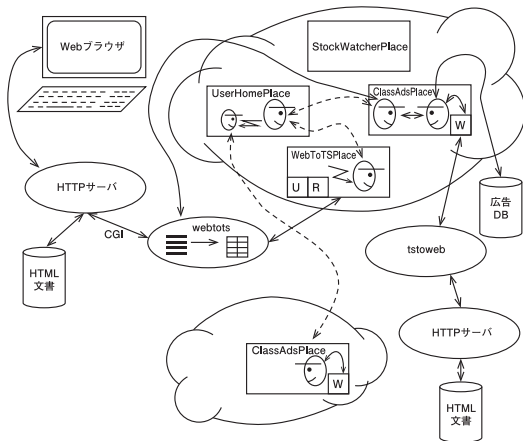
# Agent UML (AUML)

- エージェントシステムのための UML の拡張
- おもな特徴：シーケンス図の拡張

- シーケンス図：  
オブジェクト間の  
メッセージ交換手順を  
時系列で示す図
- 拡張部分：  
メッセージの種類、  
制約など

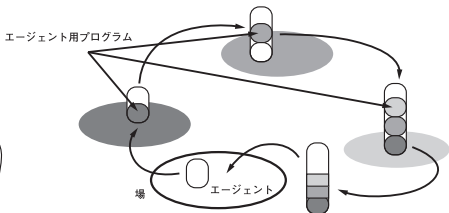
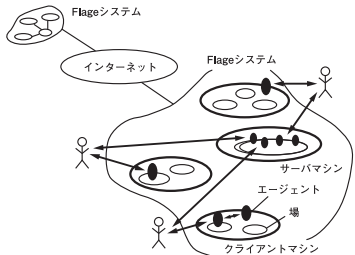


# Telescript



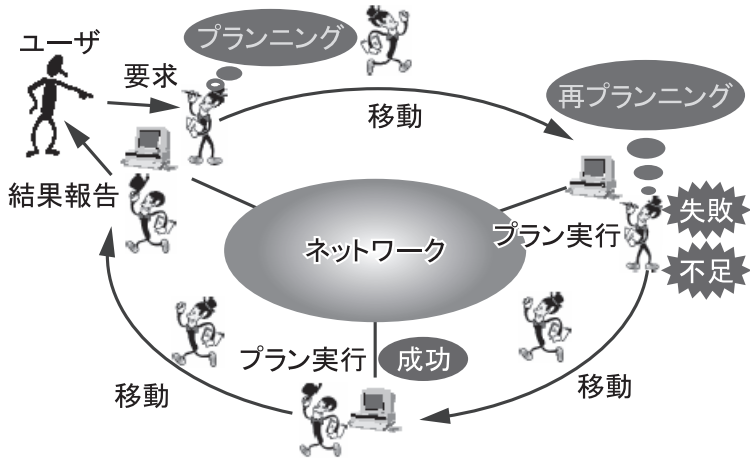
# 開発支援ソフトウェア事例

- Flage: 環境変化に対し動的に適応するエージェントのための言語
  - エージェントは場 (field) から場へと移動し, 新たな場に入ると場が提供する適応のための振舞いを動的に獲得

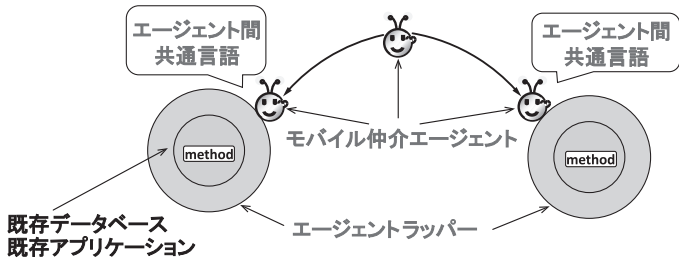




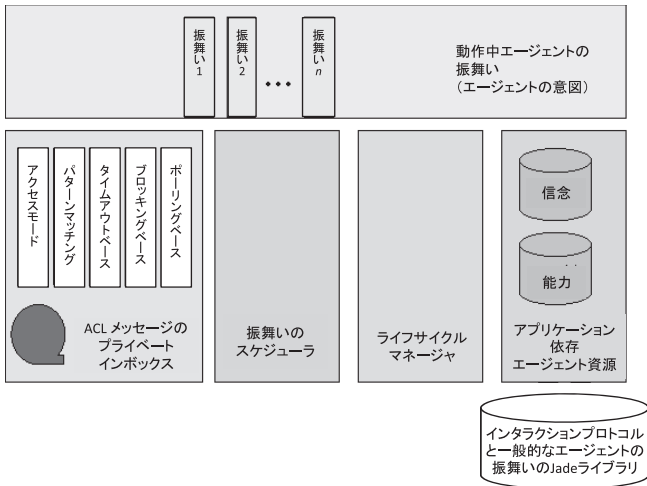
# Plangentの動作



# Bee-gent の動作



# 一般的な JADE エージェントのインタラクションアーキテクチャ



# 数理論理学の概要

- 記号を用いて論理的推論の正しさを数学的に定式化した理論体系
  - 記号論理学ともいう
- 構成要素
  - 命題や論理式の構文規則
    - 命題：正しいか否か(真偽)が明確に定まる主張
    - 論理式：変数を含んでいる可能性のある命題
  - 公理系：真であるとあらかじめ定義された論理式の集合
  - 推論規則：真な論理式から新たに真な論理式を導出する規則

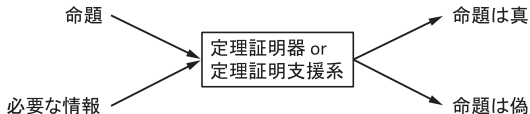
# 数理論理学の概要

- 構成要素(続き)

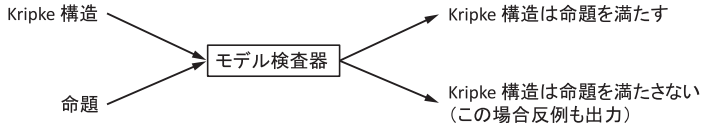
- 演繹：複数の推論規則の適用により, 論理式の集合から論理式を導出する手続き
  - 論理式の集合を前提, 導出される論理式を結論と呼ぶ
- 証明：公理系からの演繹
- 定理：証明の結論となる命題
- 意味論：論理式の構成要素を意味領域と呼ばれる数学的対象にマッピングする写像
  - 特に命題は真理値(真と偽を表す値)にマッピングされる

# 推論技術

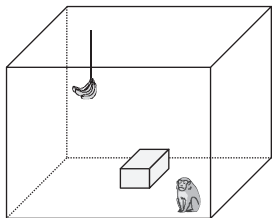
## • 定理証明



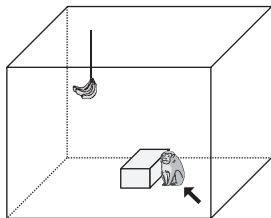
## • モデル検査



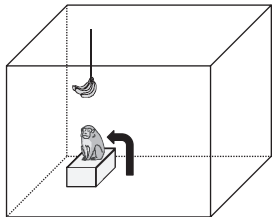
# プランニングとは？



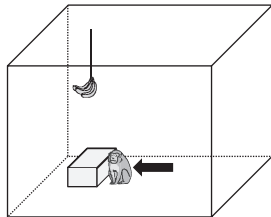
サルはバナナをとりたい



箱の横に移動

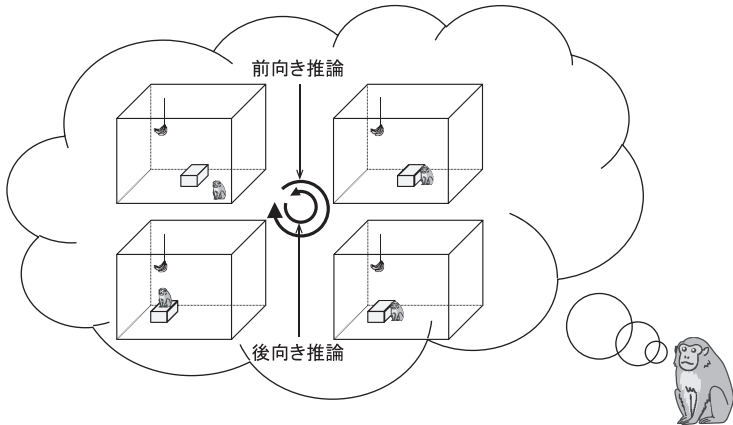


箱に乗ってバナナをとる



箱をバナナの下まで押す

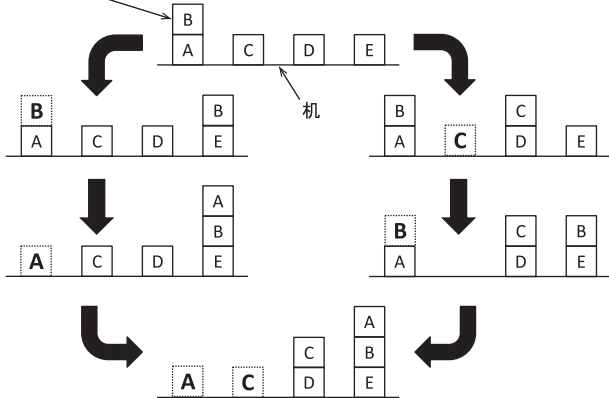
# プランニングの種類 - 線形プランニング





# プランニングの種類 - 非線形プランニング

積み木



# ソフトウェア工学 (Software Engineering) とは？

- IEEE 標準 (IEEE 610.12-1990) による定義
  - ソフトウェアの開発, 運用, および保守に対する, 体系的で洗練され, かつ定量化可能な手法 (すなわち工学) の応用
- ソフトウェア工学の詳細 – SWEBOK
  - Software Engineering Body of Knowledge (ソフトウェア工学知識体系)
  - IEEE により策定

# ソフトウェア工学知識体系 SWEBOK

- 15の知識分野から構成
  - 3つに分類：工程別の知識，工程全体にわたる知識，そのほか

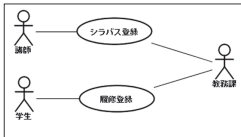
- 工程別



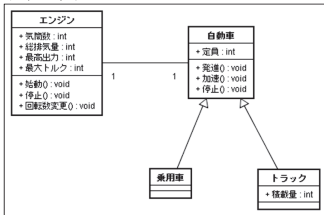
- 工程全体：構成管理，開発管理，プロセス，モデルと手法，品質
- そのほか：実務，経済学，計算技術基礎，数学的基礎，工学的基礎

# オブジェクト指向開発方法論 - UML図表記の例

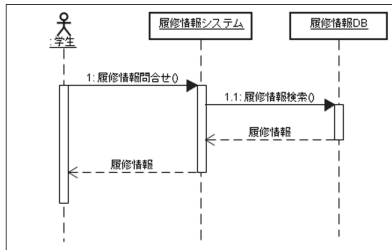
ユースケース図



クラス図

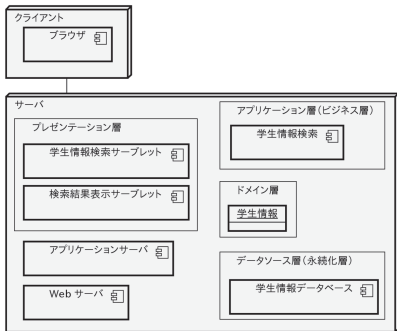


シーケンス図

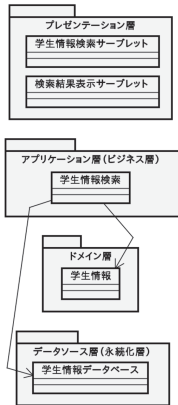


# ソフトウェアアーキテクチャ

- 学務情報システムの例



物理ビュー



開発ビュー(クラス図)

# 形式手法 – モデル検査による検証事例

検証したい性質

入力

モデル検査ツール  
(SPIN)

入力

自動出力

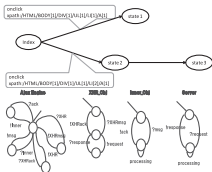
不具合実行トレース

Ajax アプリケーション

状態遷移モデル



作成



# 形式手法 - 定理証明による検証事例

The screenshot shows the CoqIDE interface with a file named `assocAppend.v`. The editor contains a proof script for the theorem `assocAppend`, which states that the `append` function is associative. The script uses inductive types for lists, a fixpoint definition for `append`, and various proof tactics like `intros`, `induction`, `replace`, `reflexivity`, and `Qed`.

```
Inductive list (A : Type) : Type :=
| nil : list A
| cons : A -> list A -> list A.

Fixpoint append (A : Type) (l1 l2 : list A) : list A :=
match l1 with
| nil => l2
| cons x l3 => cons A x (append A l3 l2)
end.

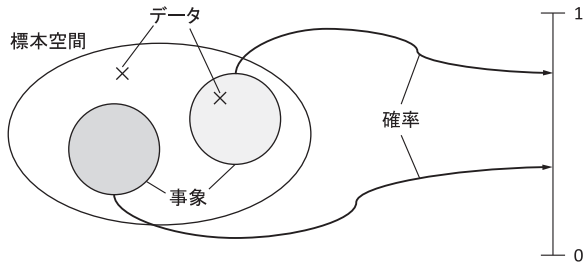
Theorem assocAppend : forall (A : Type) (l1 l2 l3 : list A),
  append A l1 (append A l2 l3) = append A (append A l1 l2) l3.
intros.
induction l1.
replace (append A (nil A) (append A l2 l3))
  with (append A l2 l3).
reflexivity.
reflexivity.
replace (append A (cons A a l1) (append A l2 l3))
  with (cons A a (append A l1 (append A l2 l3))).
replace (append A l1 (append A l2 l3))
  with (append A (append A l1 l2) l3).
reflexivity.
reflexivity.
Qed.
```

The right-hand pane shows the current goal state:

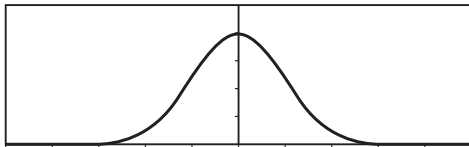
```
2 subgoal
A : Type
a : A
l1 : list A
l2 : list A
l3 : list A
IHl1 : append A l1 (append A l2 l3) = append A (append A l1 l2) l3
----- (1/2)
cons A a (append A l1 (append A l2 l3)) =
append A (append A (cons A a l1) l2) l3
----- (2/2)
cons A a (append A l1 (append A l2 l3)) =
append A (cons A a l1) (append A l2 l3)
```

At the bottom, the status bar indicates "Ready, proving assocAppend" and "Line: 24 Char: 13 CoqIde started".

# 確率論の基本概念

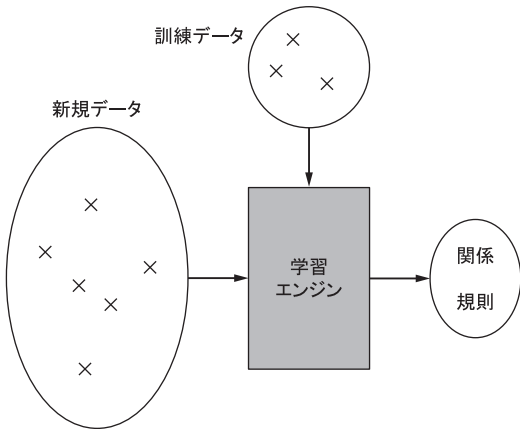


確率分布の例：正規分布

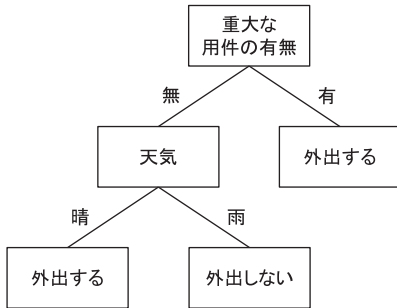




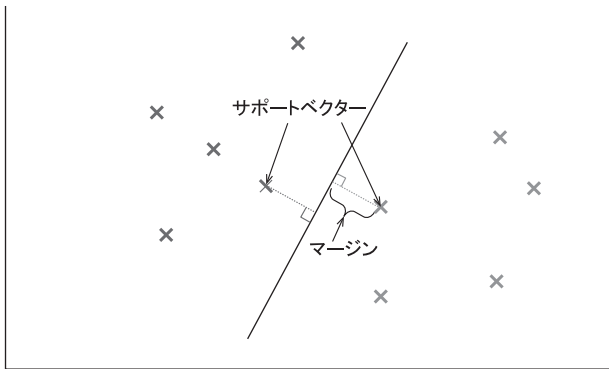
# 機械学習(教師あり学習)



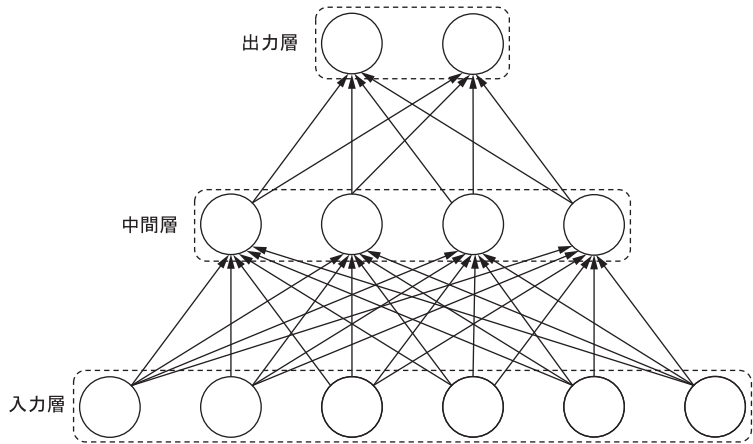
# 決定木



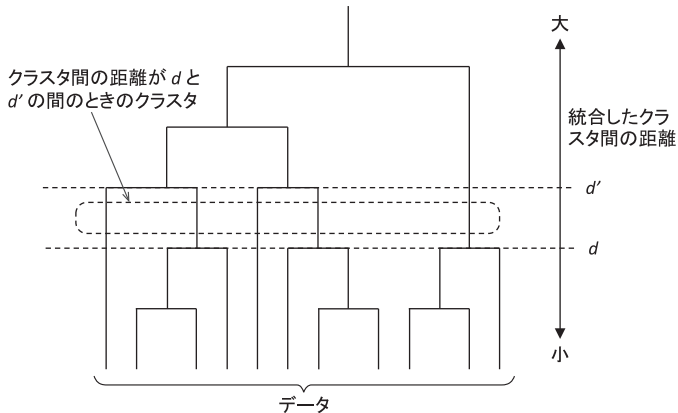
# サポートベクターマシン



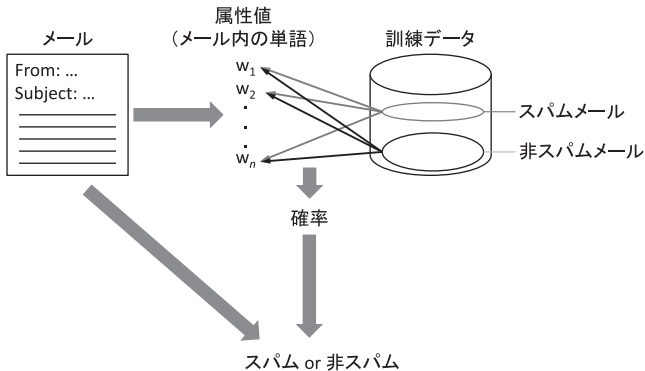
# 多層前向き結合型 ニューラルネットワーク



# 階層型クラスタリング (デンドログラム(樹形図)表現)



# ナイーブベイズ手法

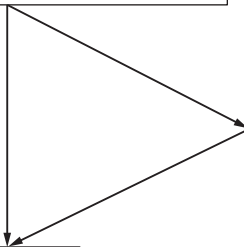


# ベイジアンネットワーク

天気  
晴れ : 0.5   くもり : 0.3   雨 : 0.2

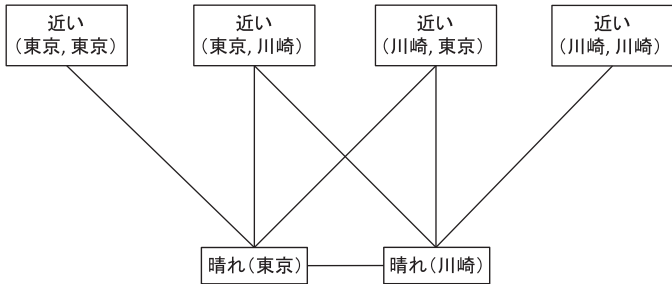
天気 \ 気温	晴れ	くもり	雨
30°C以上	0.8	0.4	0.2
30°C未満	0.2	0.6	0.8

遊園地の混雑  
大? 小?



# マルコフ論理ネットワーク

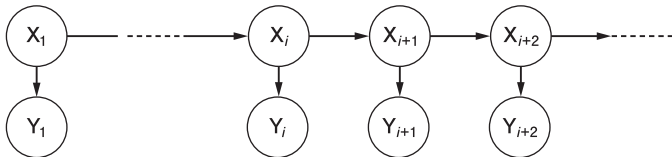
論理式「 $(\text{晴れ}(x) \wedge \text{近い}(x, y)) \Rightarrow \text{晴れ}(y)$ 」 $\equiv$ 「 $\neg \text{晴れ}(x) \vee \neg \text{近い}(x, y) \vee \text{晴れ}(y)$ 」と  
定数「東京」「川崎」に対するマルコフ論理ネットワーク



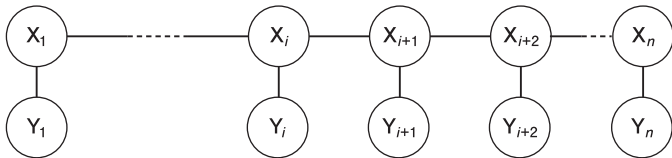


# 条件付き確率場

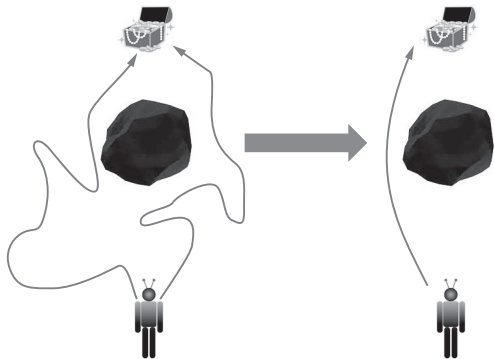
隠れマルコフモデル



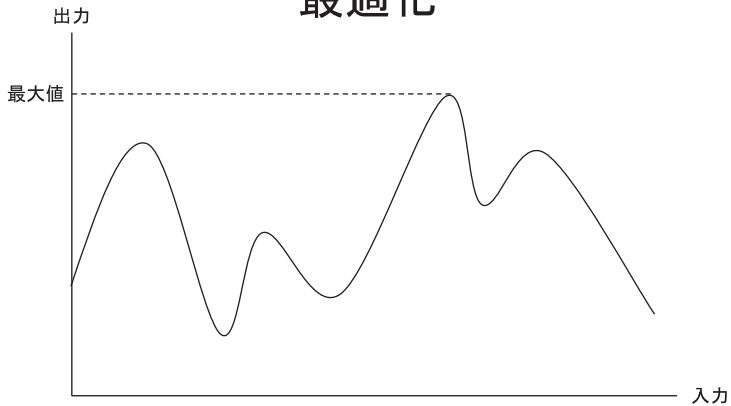
CRF



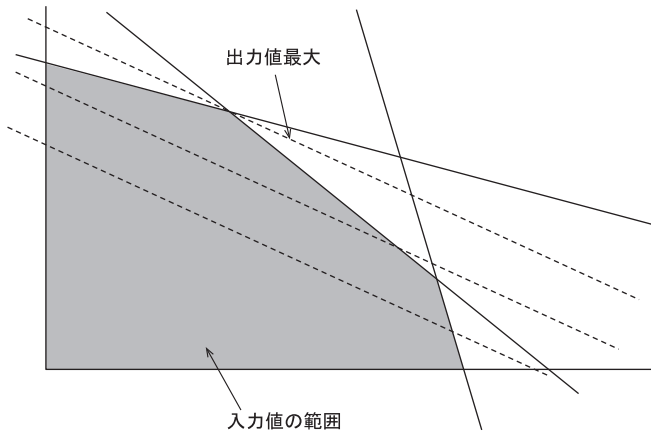
# 強化学習の例：Q学習



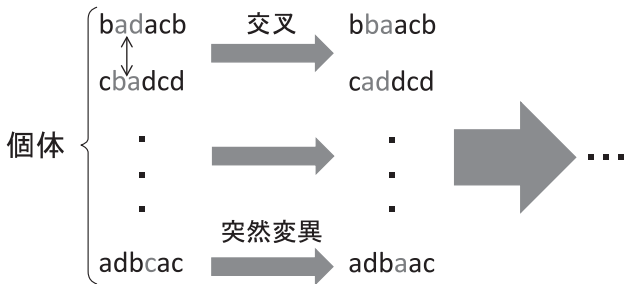
# 最適化



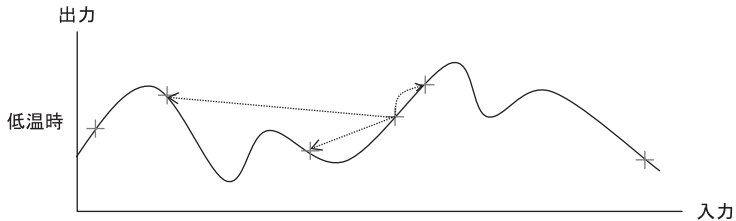
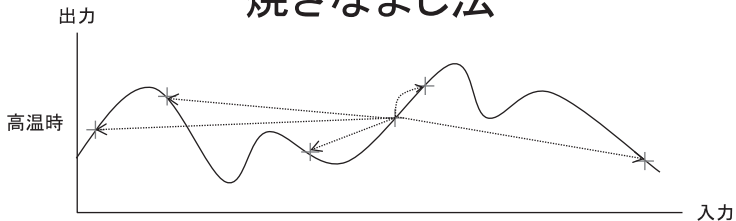
# 線形計画法



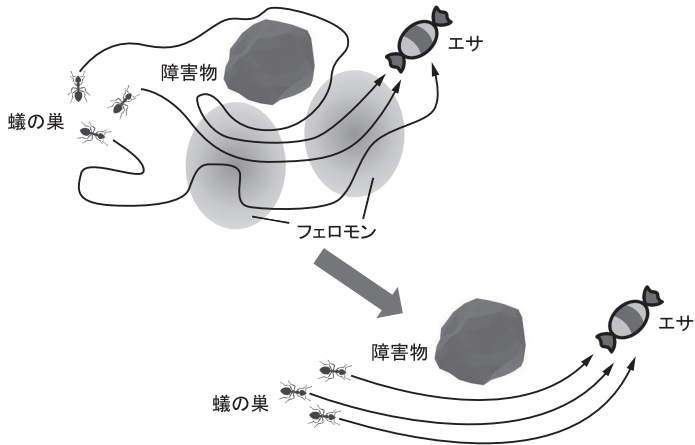
# 遺伝アルゴリズム



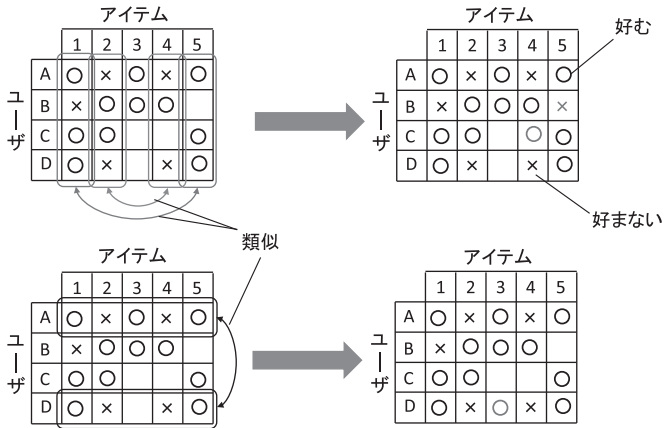
# 焼きなまし法



# 蟻コロニー最適化



# 協調フィルタリング





# 内容ベースフィルタリング

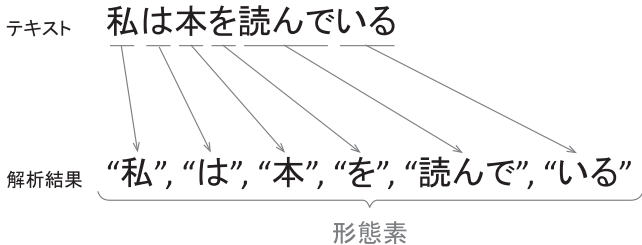
アイテム	1	2	3
タイトル	Java 入門	ハワイ旅行	戦争と平和(全4巻)
作者	○田△夫	○藤△美	トルストイ
ジャンル	コンピュータ	観光ガイド	小説
ページ数	200	400	2800
発刊年	2010	2015	2006
出版社	○○社	△△出版	○△社
価格	3 000円	2 000円	3 600円



購入



# 形態素解析



# 係り受け解析

テキスト

私は友人が手を振っているのを見つけた

文節

解析結果

私は → 振っているのを → 見つけた  
友人が → 手を →

# 特徴語抽出

テキスト

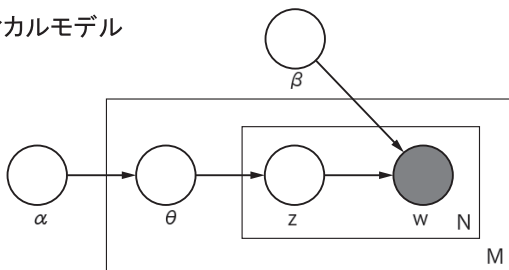
一方でテキストを構成する単語の中には、そのテキストを特徴付けるような単語が含まれることが多い

TF-IDF による解析結果

語	単語	テキスト	一方	構成	特徴	多い	含む	こと	中
TF-IDF値	3.961	3.218	1.377	1.316	1.170	1.018	1.002	0.429	0.000

# トピック分析

LDA グラフィカルモデル



$\alpha$  :  $\theta$ の分布のハイパーパラメータ  
 $\beta$  :  $w$ の分布のハイパーパラメータ  
 $\theta$  :  $z$ の分布のパラメータ  
Dir( $\alpha$ )に従う  
 $z$  : トピック  
多項分布に従う

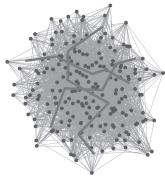
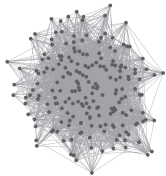
$w$  : トピック $z$ に関する単語  
多項分布に従う  
 $M$  : 文書数  
 $N$  : 文書中の単語数

# 感情分析

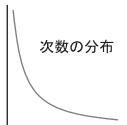
テキスト	P/N 判定	より詳細な感情
明日から旅行なのでうれしい。	positive	喜
このカメラの操作は簡単だ。	positive	楽
あなたの家の大きさにはびっくりした。	neutral	驚
このような失敗をすることは情けない。	negative	悲
列に割り込まれて腹が立った。	negative	怒

# 複雑ネットワーク

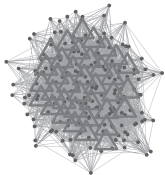
スモールワールド性：複雑なグラフでも2ノード間の距離は小さい



スケールフリー性

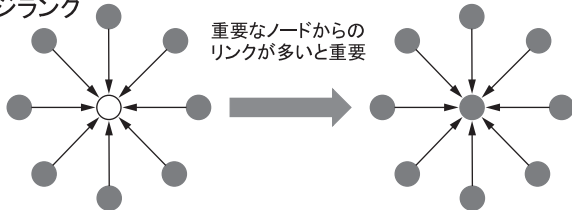


クラスタ性

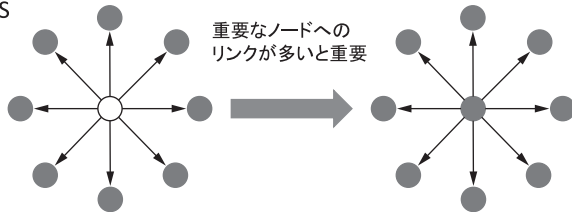


# ノードの重要性

ページランク



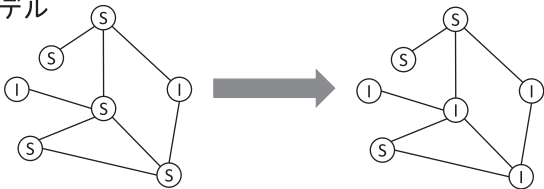
HITS



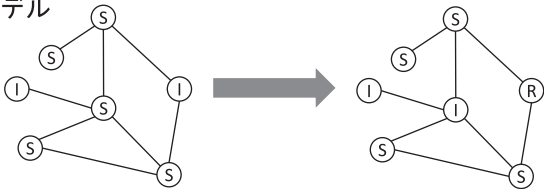


# 拡散モデル

SIS モデル



SIR モデル

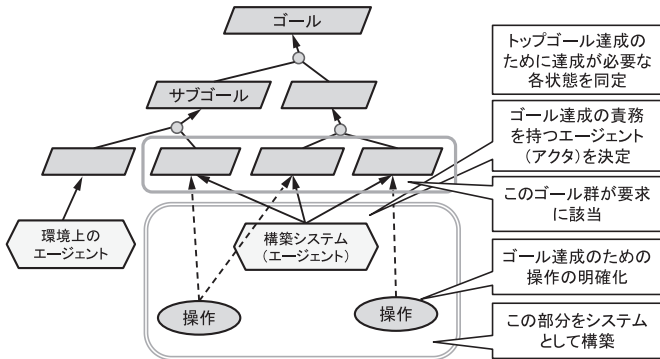


# KAOSの分析手順

Why

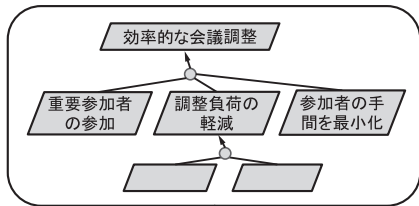
Who

What

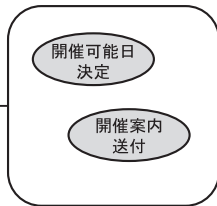


# KAOSで用いるモデル

## ゴールモデル



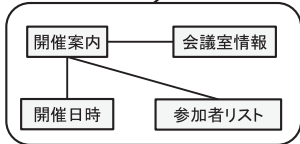
## 操作モデル



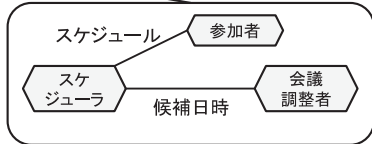
操作化

関心事

責務割り当て

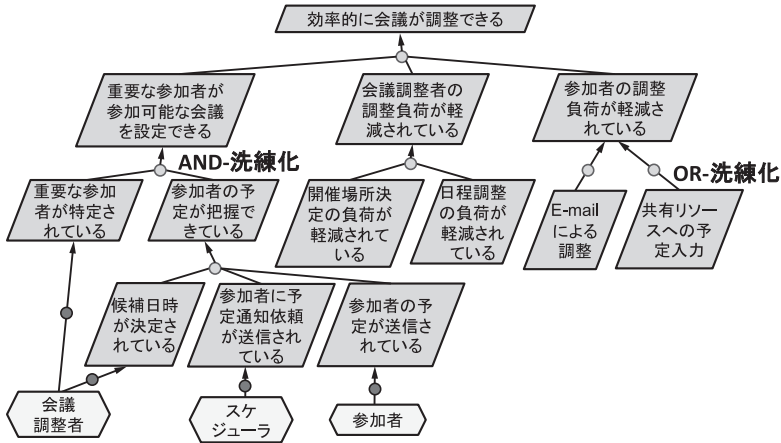


## オブジェクトモデル

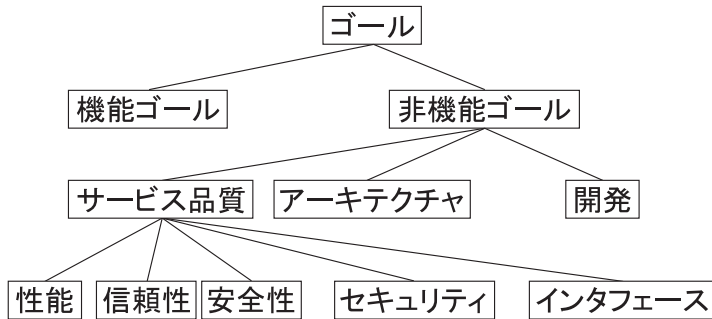


## エージェントモデル(責務モデル)

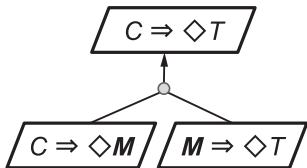
# ゴールモデルの例



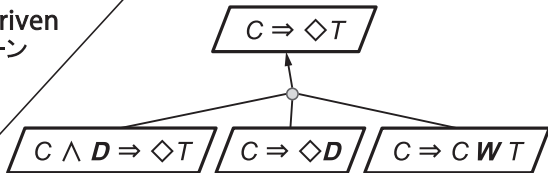
# カテゴリ概観



# ゴールパターン

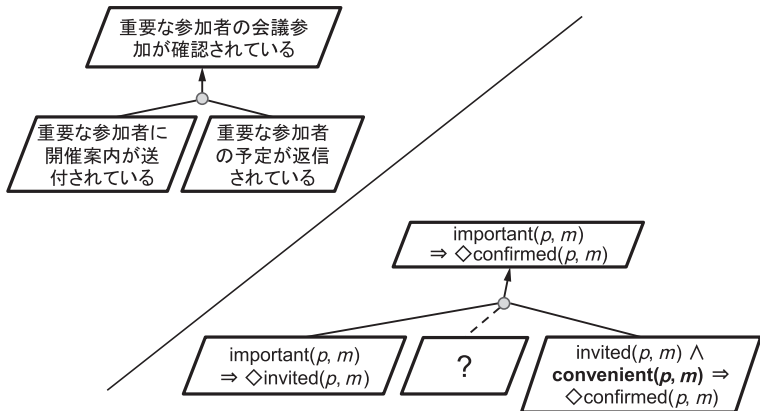


Milestone-driven  
洗練パターン

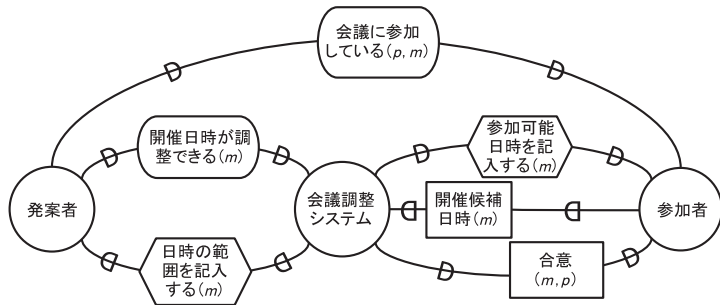


Guard-introduction  
洗練パターン

# ゴールパターンの活用例



# i\* : SD (Strategic Dependency) モデル



凡例



アクタ



タスク



依存関係

(左のアクタは右のアクタに要素( )に関して依存)



ゴール



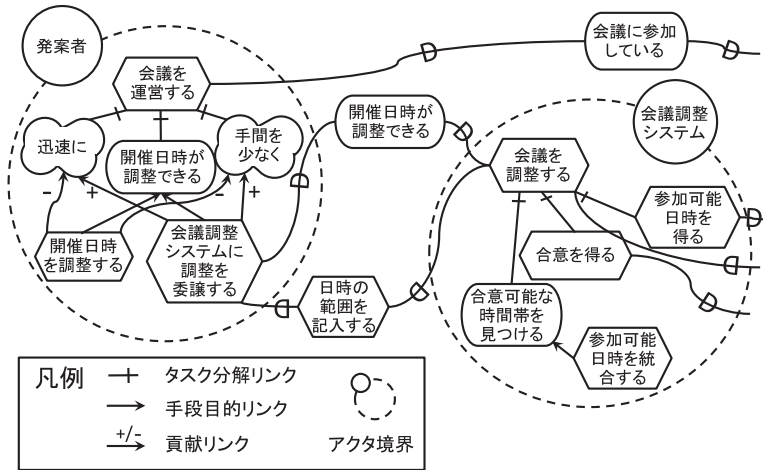
リソース



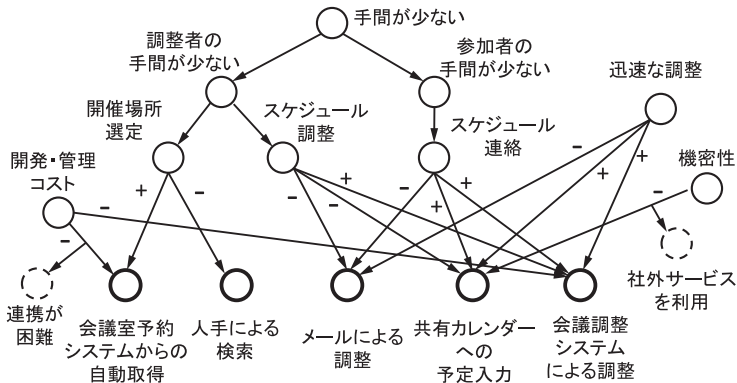
ソフトゴール



# i\* : SR (Strategic Rationale) モデル

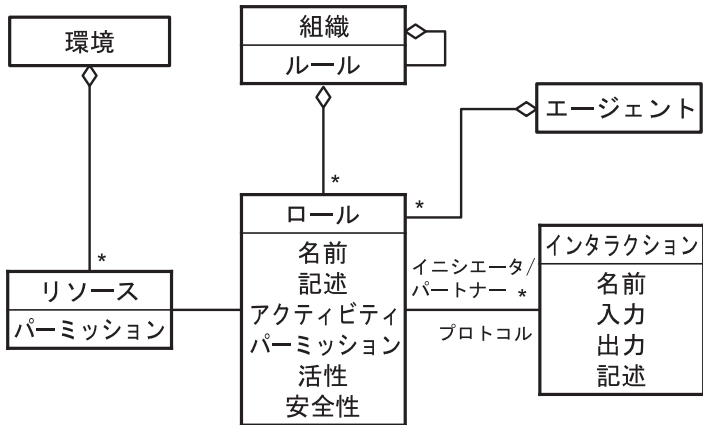


# NFRフレームワークによる分析例

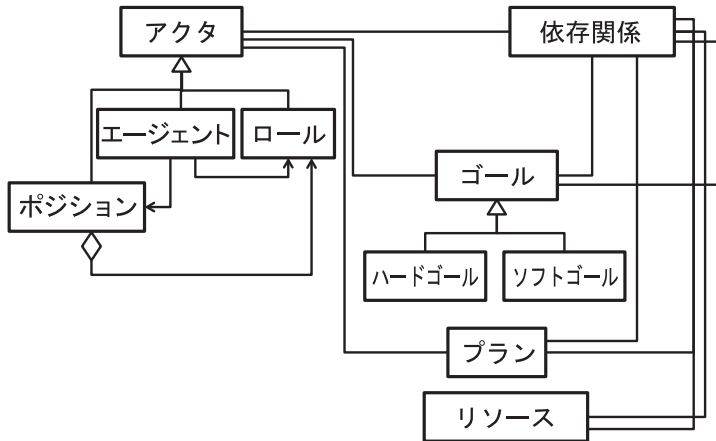


凡例 ○ 非機能ゴール    ● 満足化ゴール    ⊖ 論証ゴール

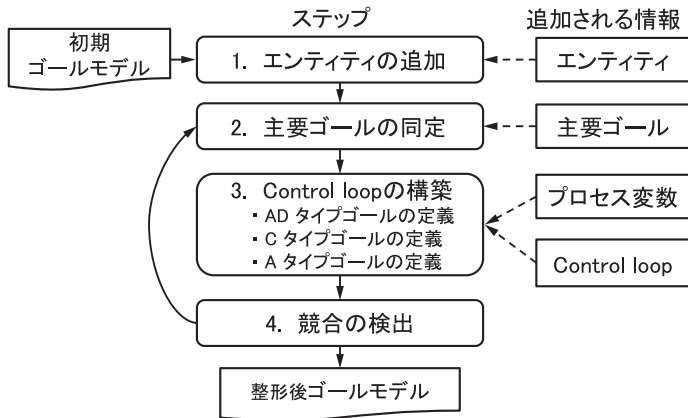
# Gaiaのメタモデル



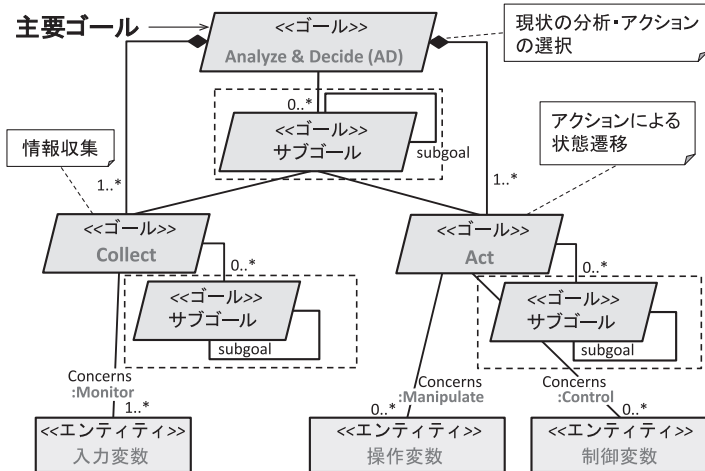
# Troposのメタモデル



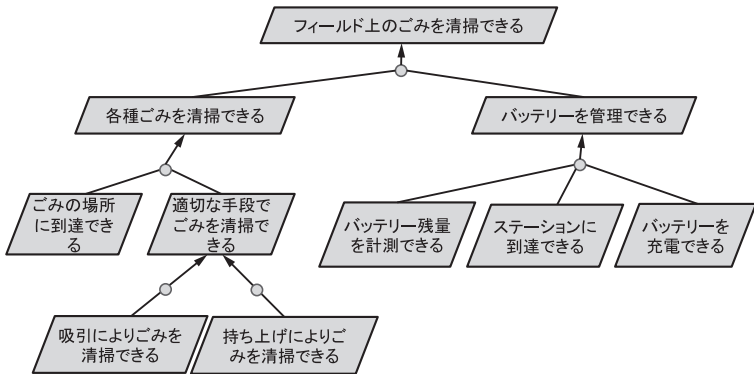
# 研究事例：ソフトウェア変更の局所化 — 整形プロセス —



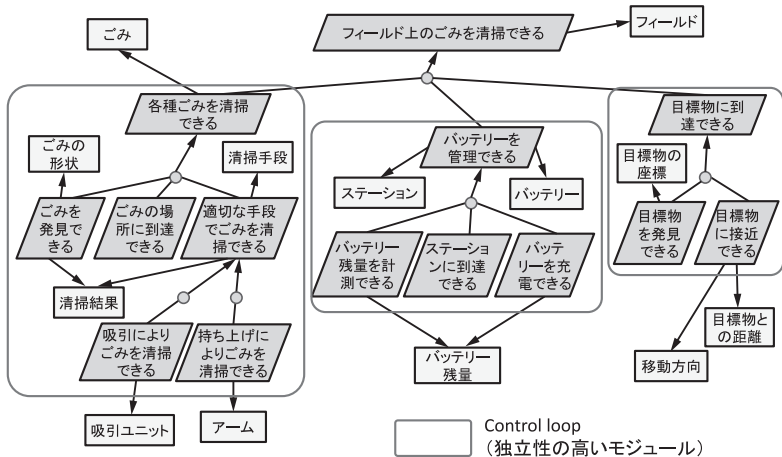
主要ゴール



# 清掃ロボットに対するゴールモデル



# 整形後ゴールモデル





# self-\* 特性の階層化

一般的特性

自己適応

自己管理

主要特性

自己  
構成

自己  
最適化

自己  
修復

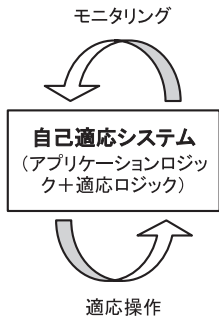
自己  
防衛

基本特性

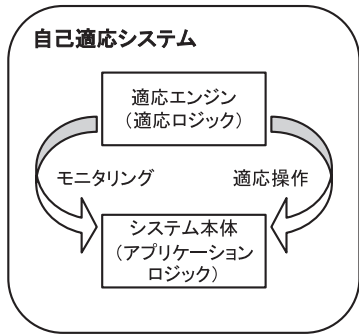
自己認識

文脈認識

# 内部アプローチと外部アプローチ

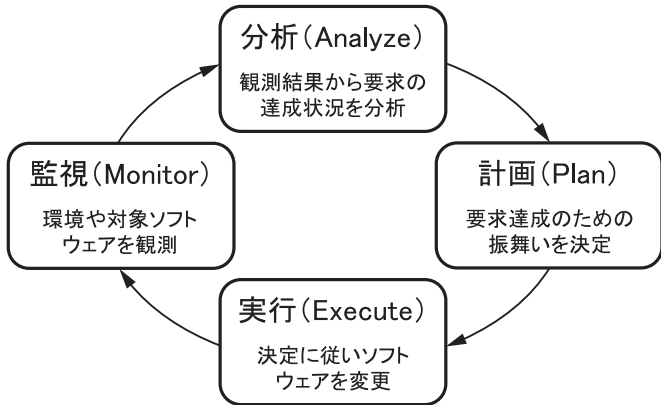


内部アプローチ

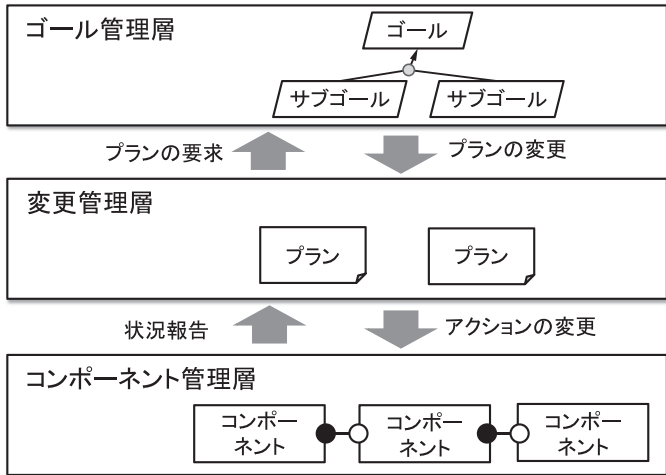


外部アプローチ

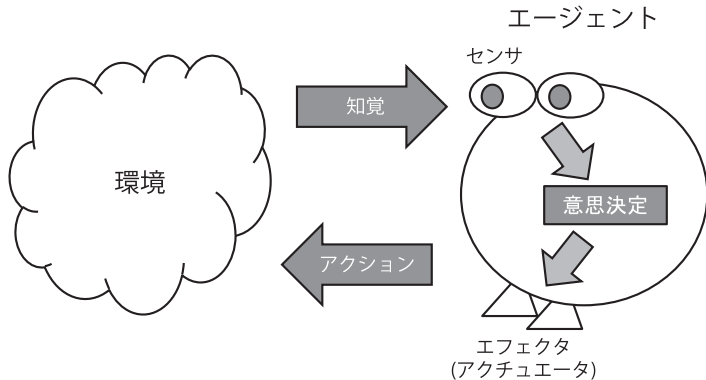
# MAPEループ



# 3層アーキテクチャ

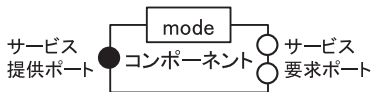


# 伝統的なエージェントモデル

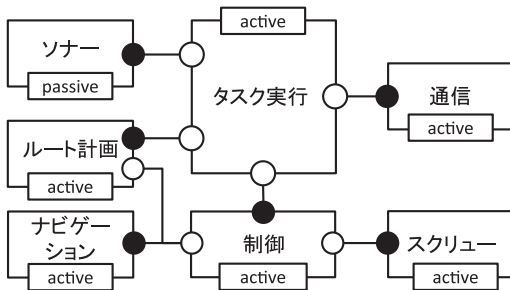


# 拡張Darwinモデル

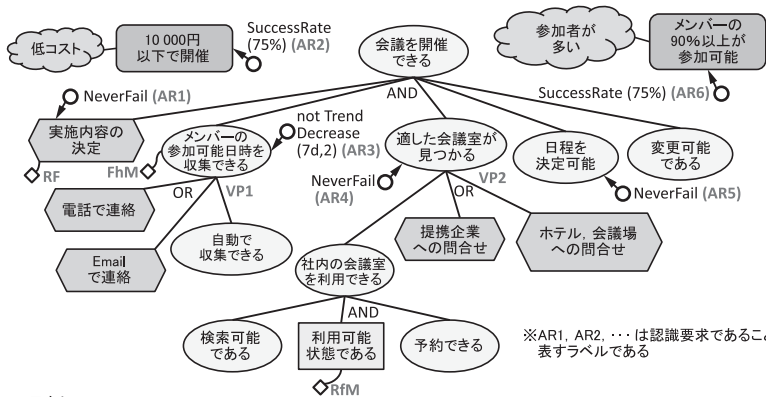
拡張Darwinモデル



拡張Darwinモデル  
の記述例



# 会議調整システムのゴールモデル



凡例:

ゴール

ソフトゴール

タスク

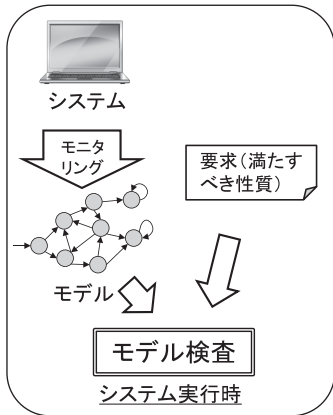
前提

品質制約

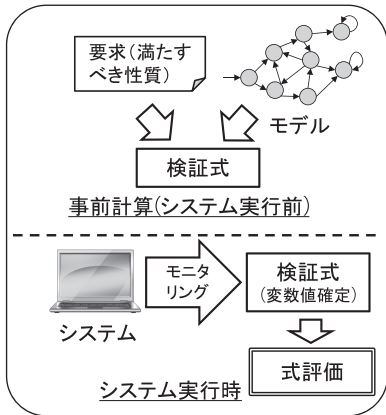
○認識要求

◇制御変数

# 動的検証の効率化



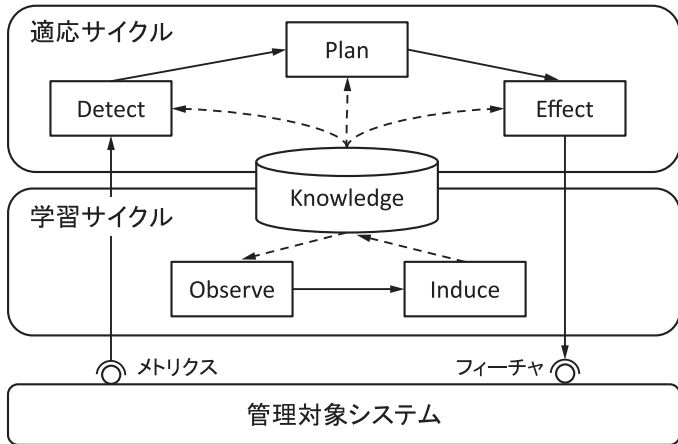
従来の動的検証



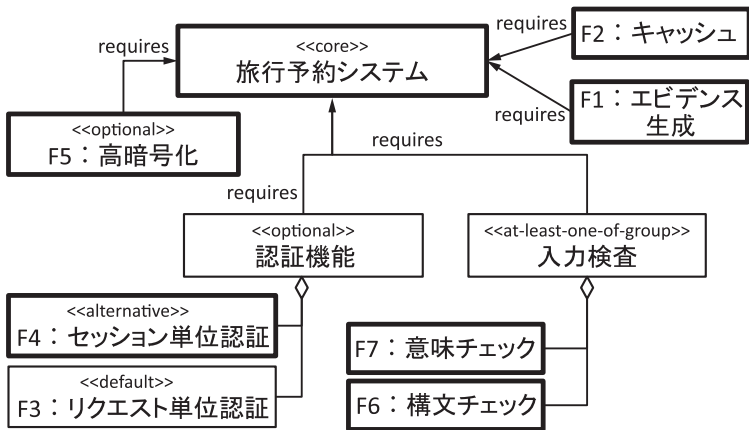
Filieriらの動的検証



# FUSIONフレームワークの概観



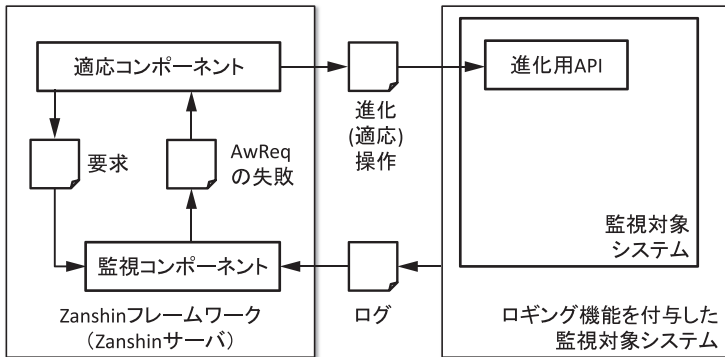
# 旅行予約システムの例： フィーチャモデル



# 旅行予約システムの例： FUSIONによる学習結果

影響変数 (フィーチャ)	誘導関数				
	$M_{G1}$	$M_{G2}$	$M_{G3}$	$M_{G4}$	...
Core	-0.84	-0.16	1.33	0	...
F1	1.55	1.14		2	...
F2	-0.67	-0.94			...
F3	0.71		-0.67		...
F4		-0.17		1	...
F5				4	...
F6			0.24		...
F7			0.59		...
F1F3	0.16				...
...	...	...	...	...	...

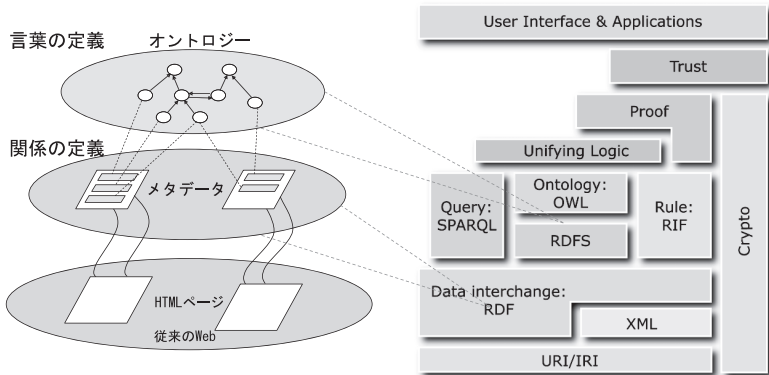
# Zanshinフレームワークの概観



# AR4に対するシナリオの実行結果

- 1: Requirement failed: D\_LocalAvail
- 2: Requirement failed: G\_UseLocal
- 3: Requirement started: T\_CallPartner
- 4: Requirement failed: T\_CallPartner
- 5: Requirement ended: T\_CallPartner
- 6: Requirement started: T\_CallHotel
- 7: Requirement failed: T\_CallHotel
- 8: Requirement ended: T\_CallHotel
- 9: Requirement failed: G\_FindRoom
- 10: Requirement ended: G\_FindRoom
- 11: Requirement failed: G\_SchedMeet
- 12: Requirement ended: G\_SchedMeet
- 13: Processing state change: AR4 (ref. G\_FindRoom) -> failed
- 14: Selected adaptation strategy: ReconfigurationStrategy
- 15: Applying strategy ReconfigurationStrategy(qualia; class-level)...
- 16: Parameters chosen: [CV\_RfM]
- 17: Values to inc/decrement in the chosen parameters: [1.00000]
- 18: Produced new configuration with 1 changed parameter(s)
- 19: RMI Target System Controller forwarding instruction: apply-config(SchedulerGoalModel, it.unitn.disi.zanshin.model.gore.impl.ConfigurationImpl@542d5e4a, class-level)

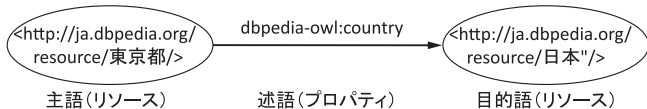
# セマンティックWebの階層構成



# セマンティックWebの発展段階

- フェーズ1：メタデータ活用
  - メタデータはデータに対する付加情報。Webコンテンツにそのコンテンツの管理や処理のためのメタデータを付与することで、整理や統合など効果的なコンテンツ管理が可能になり、また高度なコンテンツ利用が可能になる。
- フェーズ2：メタデータとオントロジー活用
  - オントロジーはメタデータを活用するための知識。オントロジーにより、業界による用語の表現の違いを吸収し、用語間の関係を反映した推論処理を行うことが可能。
- フェーズ3：メタデータとオントロジーと知的エージェントの活用
  - 知的エージェントはWeb上の計算機で行うさまざまな処理。メタデータとオントロジーによってWeb上のデータを連携した高度な処理が可能になる。

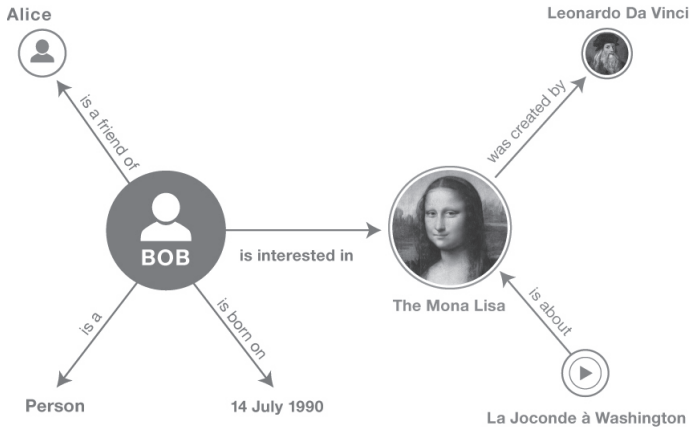
# メタデータRDF



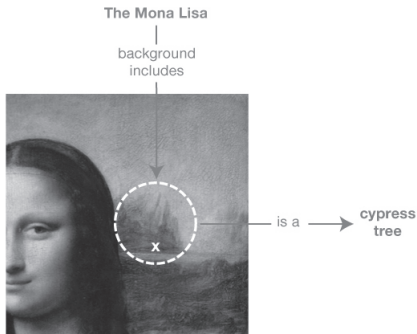
```
1: <?xml version="1.0" encoding="utf-8" ?>
2: <rdf:RDF
3:     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5:     xmlns:dbpedia-owl="http://dbpedia.org/ontology/"
6: <rdf:Description rdf:about="http://ja.dbpedia.org/resource/東京都">
7:   <dbpedia-owl:country rdf:resource="http://ja.dbpedia.org/resource/日本"/>
```



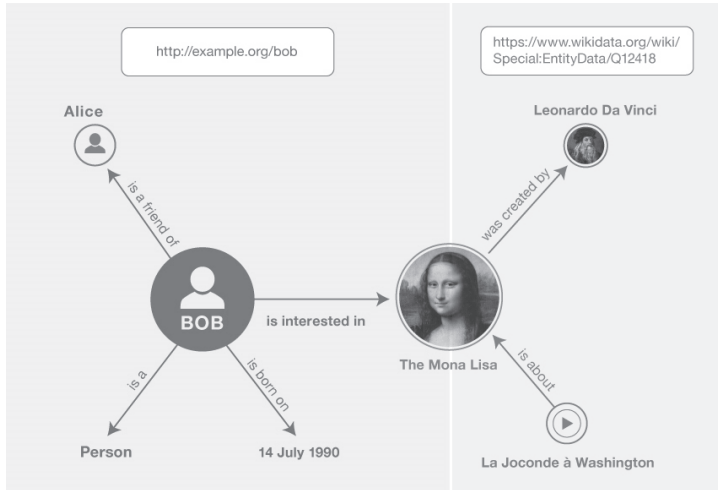
# トリプルが連結されたネットワークグラフ



# モナ・リザの背景にある杉クラスに属する 名前のないリソース



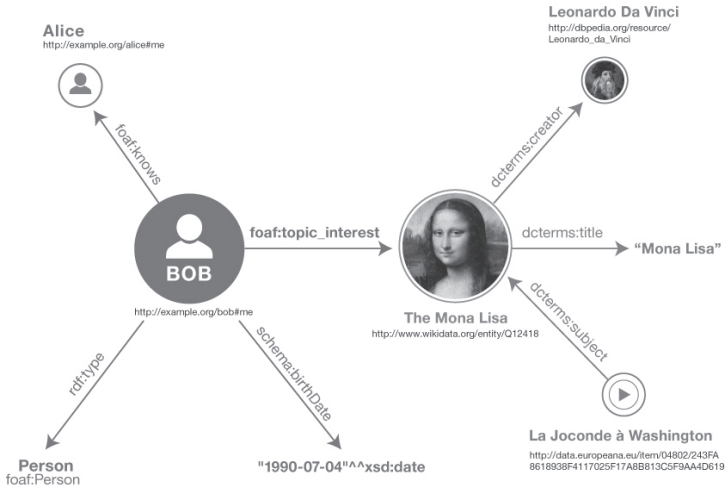
# 名前付きグラフの例



# RDFスキーマ

Construct	Syntactic form	Description
<u>Class</u> (a class)	<b>C</b> rdf:type rdfs:Class	<b>C</b> (a resource) is an RDF class
<u>Property</u> (a class)	<b>P</b> rdf:type rdf:Property	<b>P</b> (a resource) is an RDF property
<u>type</u> (a property)	<b>I</b> rdf:type <b>C</b>	<b>I</b> (a resource) is an instance of <b>C</b> (a class)
<u>subClassOf</u> (a property)	<b>C1</b> rdfs:subClassOf <b>C2</b>	<b>C1</b> (a class) is a subclass of <b>C2</b> (a class)
<u>subPropertyOf</u> (a property)	<b>P1</b> rdfs:subPropertyOf <b>P2</b>	<b>P1</b> (a property) is a sub-property of <b>P2</b> (a property)
<u>domain</u> (a property)	<b>P</b> rdfs:domain <b>C</b>	domain of <b>P</b> (a property) is <b>C</b> (a class)
<u>range</u> (a property)	<b>P</b> rdfs:range <b>C</b>	range of <b>P</b> (a property) is <b>C</b> (a class)

# N-Triplesで表現されたRDFグラフ





# 検索言語 SPARQL

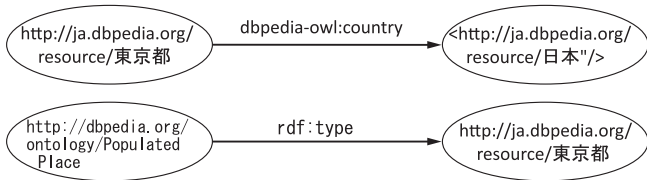


```
1: PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
2: SELECT ?prefecture ?country
3: WHERE {
4:     ?prefecture dbpedia-owl:country ?country .
5: }
```

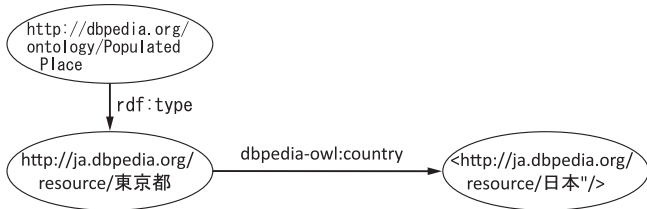
## 検索結果

prefecture	population
<a href="http://ja.dbpedia.org/resource/東京都">http://ja.dbpedia.org/resource/東京都</a>	<a href="http://ja.dbpedia.org/resource/日本">http://ja.dbpedia.org/resource/日本</a>

# Linked Dataの構成

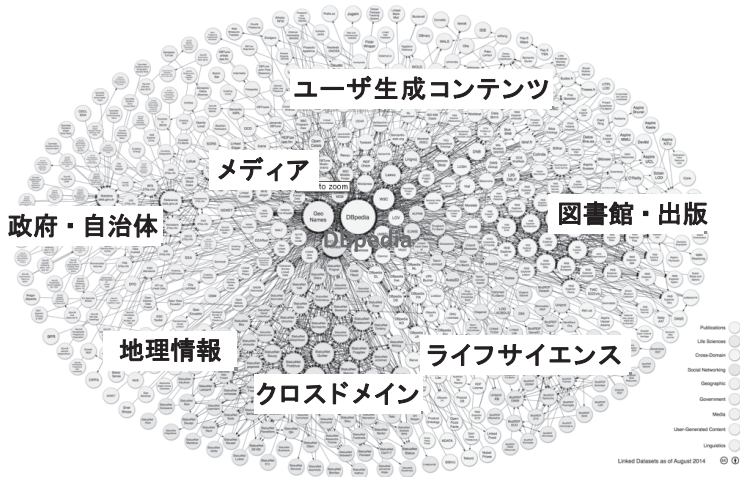


IRIが同じノードで2つのトリプルを結合





# Linked Open Data クラウド



# DBpedia Japaneseの トップページ

[DBpedia Japanese](#)   [ホーム](#)   [SPARQL Endpoint](#)   [Faceted Browser](#)   [iSPARQL](#)   [ダウンロード](#)



DBpediaはWikipediaから情報を抽出してLOD (Linked Open Data)として公開するコミュニティプロジェクトです。本家のDBpediaは主にWikipedia英語版を対象としています。DBpedia Japaneseの目的は、Wikipedia日本語版を対象としたDBpediaを提供することです。

## リソース例

- 東京都
- 森業
- 国立情報学研究所
- 情報・システム研究機構
- 新宿駅
- 達長寺
- サザンオールスターズ
- 難波八幡宮
- 国道1号
- 初音ミク
- ももいろクローバーZ
- 知床半島
- 平清盛
- 姫路城
- レオナルド・ダ・ヴィンチ

## ダウンロード

[データダンプのダウンロード](#)

## SPARQL例

SPARQL Endpoint

### 東京都

```
SELECT DISTINCT *
WHERE {
  <http://ja.dbpedia.org/resource/東京都> ?p ?o .
}
```

SPARQL結果

### ロック音楽のリスト (もしあれば画像uriも)

```
SELECT DISTINCT ?label ?depiction
WHERE {
  ?s prop-ja:genre <http://ja.dbpedia.org/resource/ロック_(音楽)> ;
  rdfs:label ?label .
  OPTIONAL { ?s foaf:depiction ?depiction . }
}
```

SPARQL結果

誕生日が1月1日

# 日本語Wikipediaオントロジーの トップページ

日本語Wikipediaオントロジー

TOP
概要
オントロジーとは
Wikipediaからのオントロジー構築
日本語Wikipediaオントロジー構築手法
利用方法
統計情報
Linked Open Data
アプリケーション
発表資料
ダウンロード
検索
SPARQL
LOD Link

## 日本語Wikipediaオントロジー研究ページへようこそ

本研究チームでは、日本語版Wikipediaを情報資源として、半自動的に大規模で汎用的なオントロジーを構築する事を目的とした研究を行っています。

我々は構築したオントロジーを日本語Wikipediaオントロジー (JWO: Japanese Wikipedia Ontology)と呼んでいます。

このWebサイトでは、日本語Wikipediaオントロジー研究のための情報を配信していく予定です。

本研究チームでは、今後の研究に生かすためのフィードバックを随時募集しております。

- 作成したLODとJWOとをリンクした際のご報告
- 日本語Wikipediaオントロジーの活用事例
- ご意見ご要望

等々のフィードバックを玉川<[s\\_tamagawa@ao.keio.ac.jp](mailto:s_tamagawa@ao.keio.ac.jp)>までご連絡いただければ幸いです。

## 更新情報

2013.11.12

2013.11.7版のWikipediaダンプデータを使用して再抽出。  
一部トリプルが正確にとれていないバグの修正

2013.10.30

2013.10.21版のWikipediaダンプデータを使用して再抽出。

2013.6.25

2013.5.30版のWikipediaダンプデータを使用して再抽出。  
いくつかの企業情報にXBRL-LODへのリンクを追加しました。

2013.5.28

2013年度人工知能学会全国大会 05-10 Linked Dataとオントロジーにて発表予定  
1N4-05-10b-3 日本語WikipediaオントロジーのLinked Open Dataへの取り組み

2013.3.9

日本語Wikipediaオントロジー一括ダウンロードができるようになりました。  
第29回セマンティックウェブとオントロジー研究会にて発表予定

# J-GLOBAL knowledge のトップページ



ログイン

## J-GLOBAL knowledgeとは

もっとオープンに、もっと扱いやすく。革新的なアプリケーションをあなたの手で。

J-GLOBAL knowledgeは国立研究開発法人科学技術振興機構(JST)が維持管理する科学技術情報を一元的に検索・提供するサービスです。セマンティック・ウェブに適合した記述モデルであるRDFをデータの記述に使用し、SPARQLにより検索することができるほか、APIの使用により初心者でも扱いやすいインターフェイスとなっています。

一般公開を開始し、こちらからJ-GLOBALのアカウントを作成いただくことで化学物質データ(約1.5億トリプル)をご利用いただけるようになります。試験公開中のため、急な変更、メンテナンス停止等があります。こちらをお読みいただき、ログインした上でご利用ください。

※ログインしていない場合、データがヒットしません。

7月3日(金) 12:00~13:00メンテナンスのためサービスを停止します。ご迷惑おかけしますがよろしくお願ひ致します。

キーワードを入れて検索してみよう



### SPARQLでさがす

このSPARQLエンドポイントでは、RDFで構成されているデータをSPARQLを指定して問い合わせを行い、データを検索することができます。



### WebAPIをつかう

J-GLOBAL knowledgeでは、SPARQL検索に慣れていない方のためにWebAPIを用意しています。簡易な検索条件を指定して検索することができます。




### 利用方法

RDFに変換したデータのメタデータ構造、SPARQL、APIの利用方法に関する説明をします。

# datahub.ioのトップページ

Takahiro Kawamura 0



The easy way to get, use and share data

Datasets Organizations About Blog Help



## Give your data a home

Publish or register datasets, create and manage groups and communities.

Publish data for free

ecumene Census Division Boundary File [\[link\]](#)

Population by age groups, median age, sex, provinces and territories - 2006

Aboriginal identity population by age group, sex, provinces and territories - 2006

Population by age and... [CSV](#)

2006, 2006 counts

people data

★★★★☆

## Find data

Search for data, and get updates from datasets and groups that you're interested in.

the free, powerful data management platform from the Open Knowledge Foundation

Popular Tags [lod](#) [publications](#) [format-rdf](#) [Senegal](#) [ANSO](#) [menages](#) [donnees](#) [published-by-third...](#)



### Canada

Datasets for <http://www.datadotgc.ca/>. DataDotGC, which launched, in February 2010, is a...

FEATURED

### 1996 Population (Ecumene) Census Data, Canada

The parts of Canada making up the 1996 Settled Area, (or Population Ecumene), represents a selection of the 5984...

### 1996 Population (Ecumene) Census Data, Canada

To offer statistics for analysis for educational purposes and policy making.



### Senegal

Datasets for Senegal, which launched in Mars 2013, is a Senegal, citizen-led effort to promote...

FEATURED

### Banque de Données des Indicateurs Sociaux du Sénégal 2005-2006

(BADIS) AGENCE NATIONALE DE LA STATISTIQUE ET DE LA DEMOGRAPHIE Directeur Général...

### Banque de Données Economiques et Financières 2005

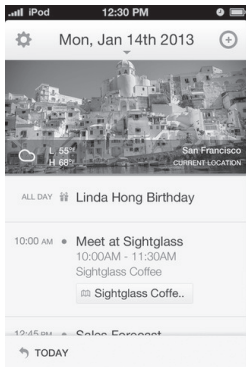
REPUBLIQUE DU SENEGAL UN PEUPLE UN BUT UNE FOI

# IBM ワトソン



- ・ Webから収集した大量の文書中の単語に多重にアノテーション、インデクス化しておく
- ・ 内部知識の20%は事前に構造化（Linked Data化）

# Tempo Smart Calendar



アップル社Siriを生んだSRI International  
によるiPhone向け秘書アプリ

# STAR-CITYのデータ

Type	Sensing	Data Source	Description	Format	Temporal Frequency (s)	Size per day (GBytes)	Data Provider (all open data)
Stream Data	Static	Journey times across Dublin City (47 routes)	Dublin Traffic Department's TRIPS system <sup>a</sup>	CSV	60	0.1	Dublin City Council via dublinked.ie <sup>b</sup>
		Road Weather Condition (11 stations)		CSV	600	0.1	NRA <sup>c</sup>
		Real-time Weather Information (19 stations)		CSV	[5, 600] (depending on stations)	[0.050, 1.5] (depending on stations)	Wunderground <sup>d</sup>
	Dynamic	Dublin Bus Stream	Vehicle activity (GPS location, line number, delay, stop flag)	SIRI: XML-based <sup>e</sup>	20	4-6	Dublin City Council via dublinked.ie <sup>f</sup>
	Dynamic	Social-Media Related Feeds	Reputable sources of road traffic conditions in Dublin City	Tweets	600	0.001 (approx. 150 tweets per day)	LiveDrive <sup>g</sup> Aaroadwatch <sup>h</sup> GardaTraffic <sup>i</sup>
Static Quasi Stream	Dynamic	Road Works and Maintenance		PDF	Updated once a week	0.001	Dublin City Council <sup>j</sup>
		Events in Dublin City	Planned events with small attendance	XML	Updated once a day	0.001	Eventbrite <sup>k</sup>
			Planned events with large attendance			0.05	Eventful <sup>l</sup>
	Static	Dublin City Map (listing of type, junctions, GPS coordinate)		ESRI SHAPE	No	0.1	Open StreetMap <sup>m</sup>

<sup>a</sup> Travel-time Reporting Integrated Performance System - <http://www.advantechdesign.com.au/trips>

<sup>b</sup> <http://dublinked.ie/datastore/datasets/dataset-215.php>

<sup>c</sup> NRA - National Roads Authority - <http://www.nratraffic.ie/weather>

<sup>d</sup> <http://www.wunderground.com/weather/api/>

<sup>e</sup> Service Interface for Real Time Information - <http://siri.org.uk>

<sup>f</sup> <http://dublinked.ie/datastore/datasets/dataset-289.php>

<sup>g</sup> <https://twitter.com/LiveDrive>

<sup>h</sup> <https://twitter.com/aaroadwatch>

<sup>i</sup> <https://twitter.com/GardaTraffic>

<sup>j</sup> <http://www.dublincity.ie/RoadsandTraffic/ScheduledDisruptions/Documents/TrafficNews.pdf>

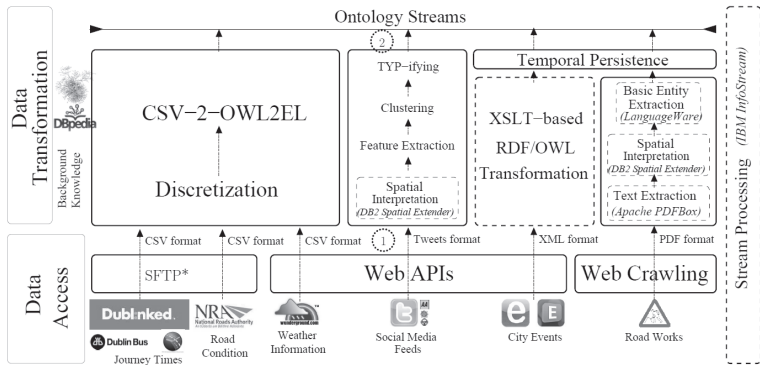
<sup>k</sup> <https://www.eventbrite.com/api>

<sup>l</sup> <http://api.eventful.com>

<sup>m</sup> <http://download.geofabrik.de/europe/ireland-and-northern-ireland.html>



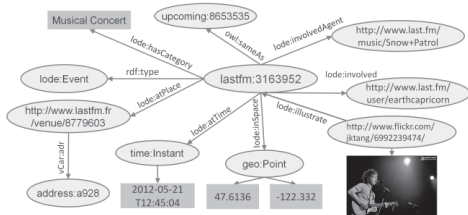
# STAR-CITYの構成



CAPTION:

  : State-of-the-art Tools  
 2 : Data Flow Ordering  
 \* : The most up-to-date data stream is (on May 17, 2013) only available through SFTP

# EventMediaLive



SPARQL  
REST API (Elda)

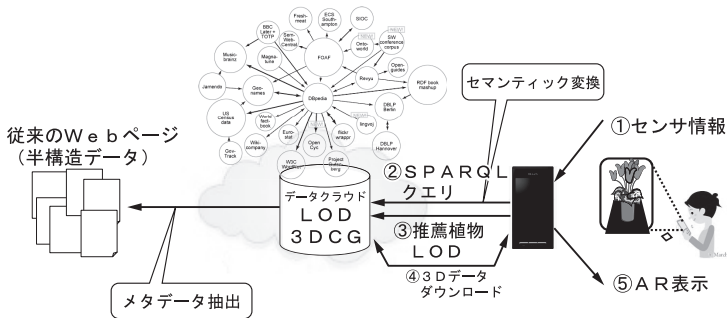


# BBC Ontologies



- ・ ニュース記事のメタデータ記述に利用
- ・ 人，組織，場所，日時，テーマ，イベントなどの要素で構造化し，RDFaを利用してWebページへ埋め込む

# 花咲かめらの構成



スマートフォンを用いて環境をセンシングし、クラウド上のLODを検索することで、環境に合う植物をARで実空間に重畳表示（植物育成支援、景観確認）する

# 花咲かめらの画面例



1. 植物を植えたい場所を見つける



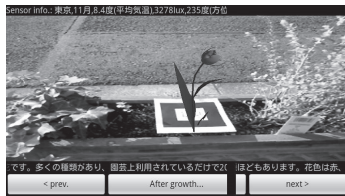
2. マーカーを置く



3. アプリをかざす



4. 環境条件に合った植物の提案



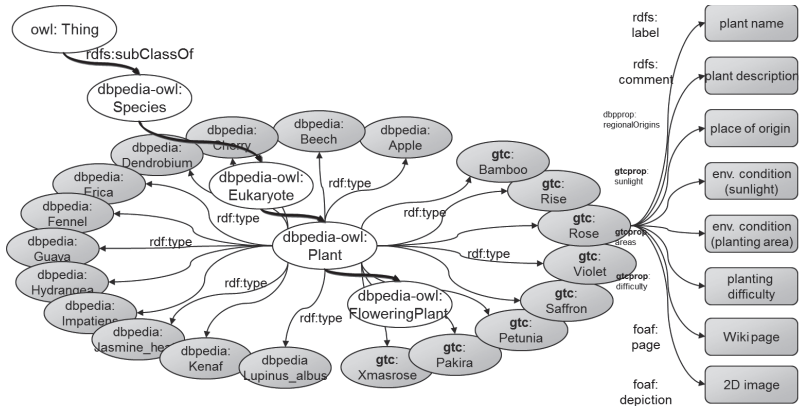
です。多くの種類があり、園芸上利用されているだけで20ほどあります。花色は赤、



フリック

5. 気に入らなければ3つまで表示

# 植物LODの概要



# 花之声の画面例



まわりに人がおらず、手が汚れているので音声操作が向いている

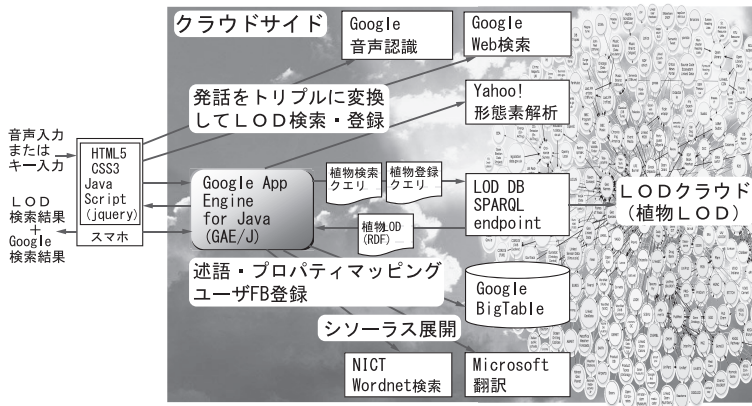


LOD検索結果

Google検索結果



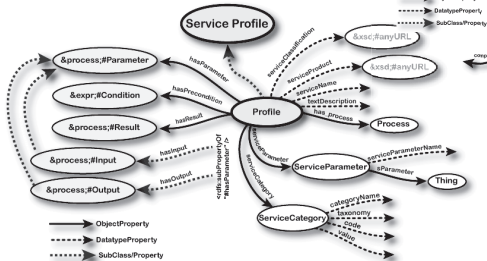
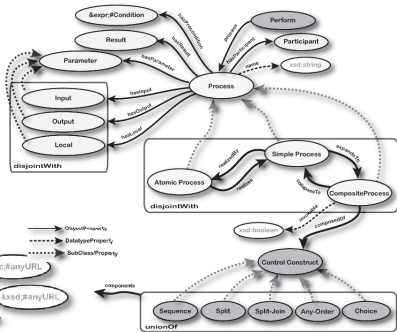
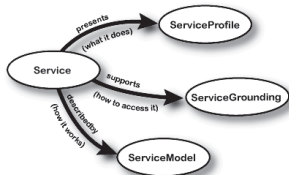
# 花之声の構成



音声を用いた園芸・農作業向け情報検索・記録サービス,  
自然発話から全世界630億トリプル(情報の単位)あるLODを検索

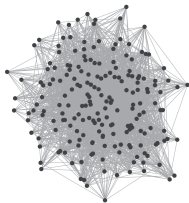


# セマンティックWebサービスの構成



# 自律 Web アプリケーション

Web 上の大量の情報



情報抽出・予測

台風の進路



○月△日に～で...のコンサート



情報推薦

この商品をチェックした人はこんな商品もチェックしています



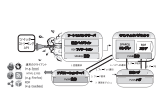
# 情報抽出・予測手法

震源地と台風の進路の特定



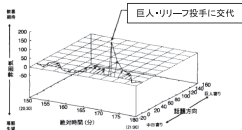
[Sakaki et al., 2010]

センサとしてのマイクロブログ利用



Linked Open Social  
Signals プロジェクト

実況の盛り上がりの抽出



[岡本 ほか, 2005]

動画からのシーン抽出



[多胡 ほか, 2010]

意見・評判抽出

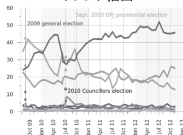
抽出した意見を表示



肯定意見が多く、  
否定意見が少ない  
ほど浮上する

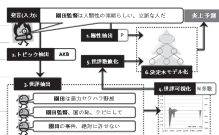
[田中 ほか, 2011]

トレンド抽出



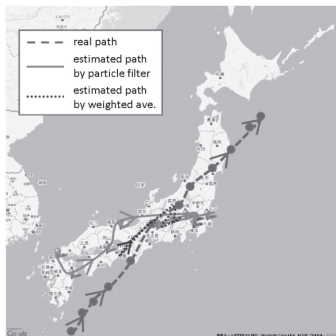
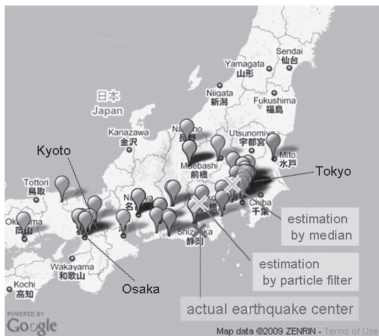
[橋本 ほか, 2011]

ツイッター炎上予測



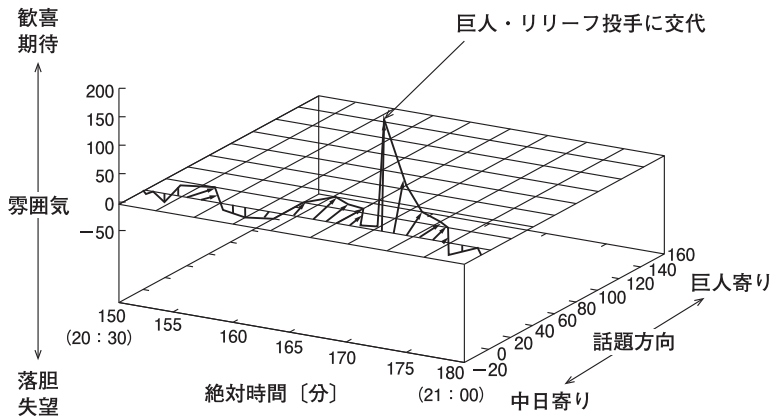
[岩崎 ほか, 2015]

# 震源地と台風の前経路の特定

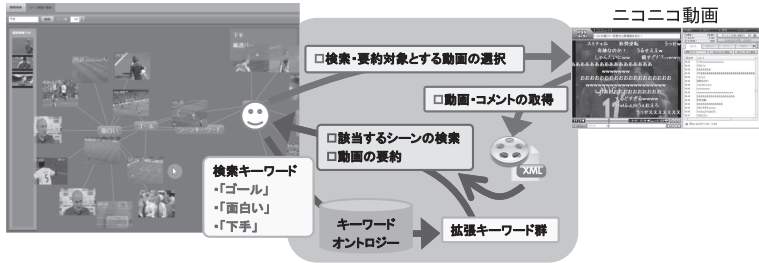




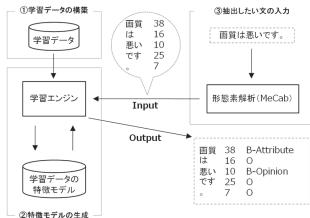
# 実況の盛り上がりの抽出



# ソーシャルアノテーションを用いた 動画シーン情報の取得



# 意見抽出による評判比較システム

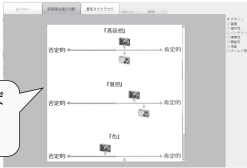


抽出した意見を表示



肯定意見が多く、  
否定意見が少ない  
ほど浮上する

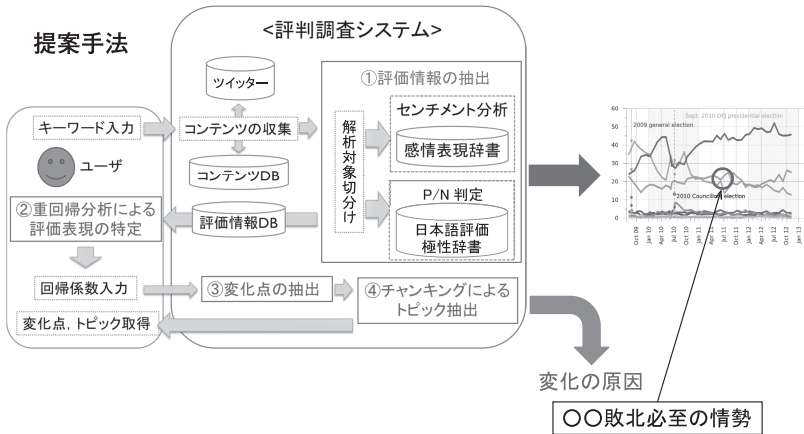
肯定意見が多ければ  
右へ、否定意見が  
多ければ左へ移動



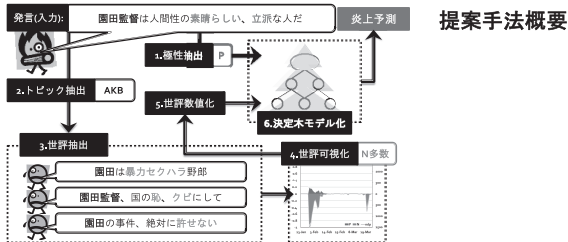


# 評判傾向抽出

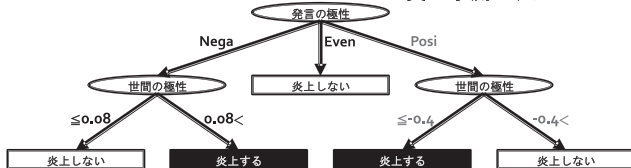
## 提案手法



# 炎上予測



## 炎上予測モデル



# 情報推薦手法

## ネット販売商品



## 災害時行動

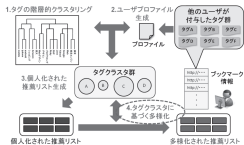


## イラスト



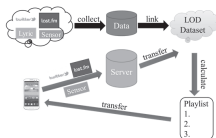
[住元 ほか, 2011]

## Web ページ



[藤原 ほか, 2013]

## 音楽



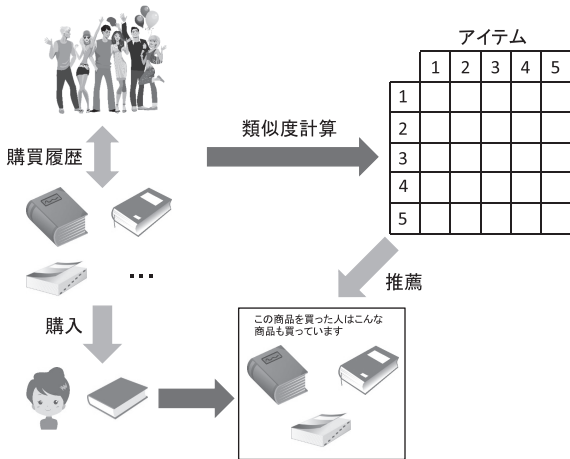
[Wang et al., 2014]

## 料理のレシピ



[池尻 ほか, 2014]

# amazon.com の商品推薦



# 震災時避難行動推薦



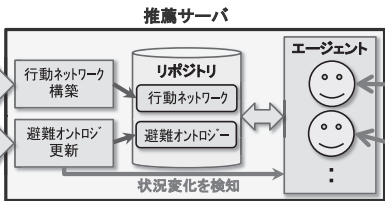
推薦端末

ツイッター



行動抽出

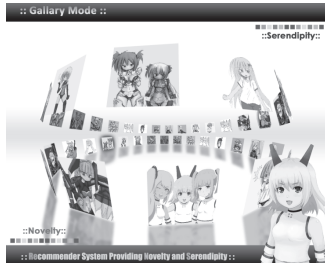
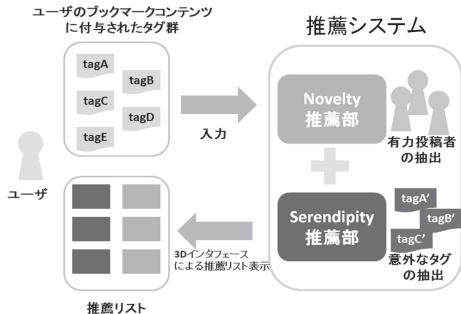
イベント抽出



ユーザ

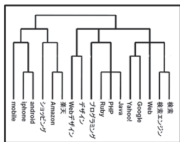


# 未知性・意外性を考慮した イラスト推薦

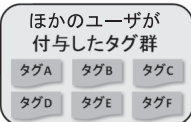


# 未知性を考慮した Web サイト推薦

## 1. タグの階層的クラスタリング



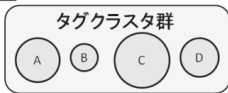
## 2. ユーザプロフィール生成



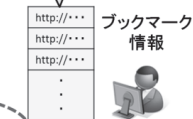
## 3. 個人化された推薦リスト生成



個人化された推薦リスト

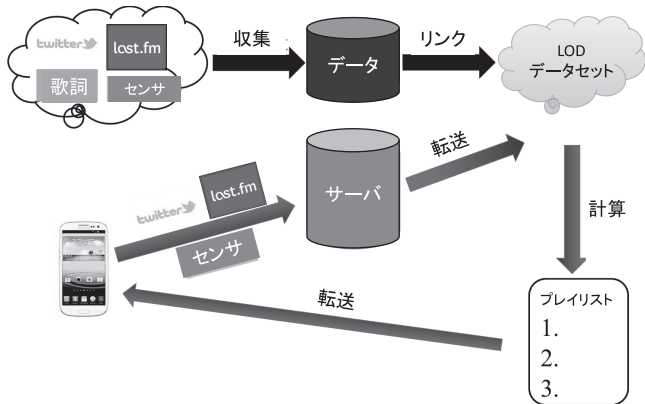


## 4. タグクラスタに基づく多様化



多様化された推薦リスト

# 連想関係を用いた音楽推薦





# 希少性に基づく意外性レシピ推薦

いつもの味に飽きた...  
でも新しい料理に挑戦  
するのは敷居が高い...



例) ハンバーグの意外食材

希少度のみ → コーfeeミンチ  
熟味酢

希少度  
+  
一般度 → きゅうり  
りんご



ユーザ

料理mの  
意外レシピ

料理m指定

全レシピ

↓ レシピ抽出

料理m  
レシピ群

↓ 食材抽出

料理m  
全食材

↓ 食材意外度算出

料理mの食材の  
意外度リスト

↓ レシピ意外度算出

料理mのレシピ  
の意外度リスト



Webサービス