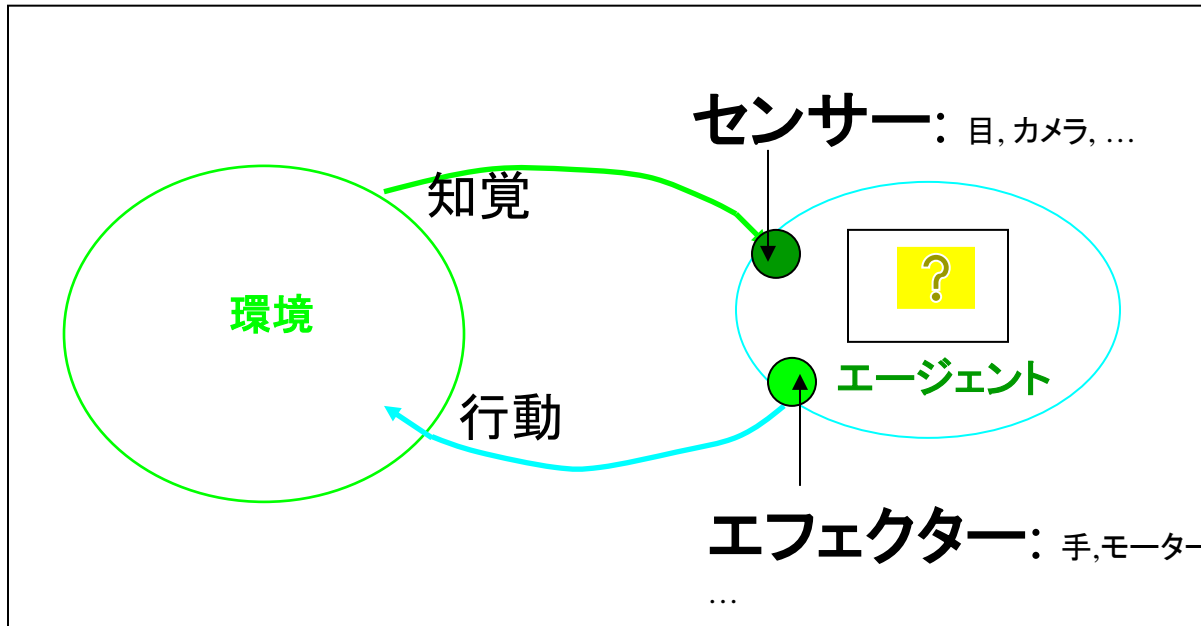


L2. Introduction (2)

- 知的エージェントとは？
- 知的エージェントの特徴とは？
- どのように知的エージェントをデザインするか？
- 知的エージェントの例
- デモ

エージェントとは何か？

- ◆ エージェントはセンサーを通じて環境を知覚し、エフェクターを通じて環境に行動することができるものです。



例:

人による運転

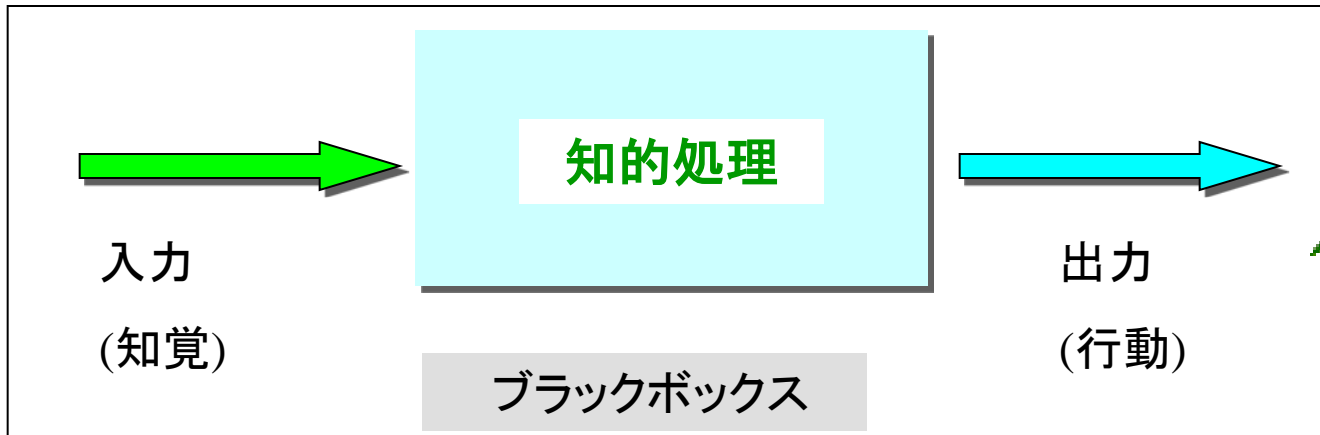
ロボットによる運転

プログラムによる運転

一般的なエージェントの仕組み

知的エージェントとは何か

- ◆ エージェントはタスクを実行するために常に一定量の知識を必要とします。従って知的エージェントと呼ばれます。



- 最上位レベルでは、エージェントは主に以下の3つのカテゴリに区別されます。
人間エージェント・ハードウェアエージェント・ソフトウェアエージェント。
- 例えば、旅行代理店、ロボット、自動化されたタクシー

知的なソフトウェアエージェント

- ◆ 知的なソフトウェアエージェントは、ユーザーのために特定のタスクを果たすことができ、また有効な方法で自律的にタスクの部分的な処理を実行する知性を備えているソフトウェア・プログラムであると定義されます。
- ◆ ソフトウェアエージェントに活用される知能とは何か？
 - 人のように考えるプログラム
 - 人のように行動するプログラム
 - 合理的に考えるプログラム
 - 合理的に行動するプログラム

知性における非常に広いバリエーションは、低い知能を持つ単純なエージェントから高い知能を持つ複雑なエージェントまであらゆるものが想定されます。

- ◆ AI (artificial intelligence) の分野は知性の本質の理解に取り組んでいます。



知的エージェントの特徴

◆ 内部の特徴

– 学習 / 推論:

エージェントは、これまでの経験から学び、順次自分の行動を環境に適応させることができます。

– 反応性:

エージェントは、環境からの情報に影響を受けたり、適切に反応することができなければなりません。

– 自律性:

エージェントは、自分の内部状態と行動の両方を制御する必要があります。エージェントの自律性の度合いを指定することもでき、重要な意思決定のためにユーザーの介入が必要になる場合もあります。

– 目的指向:

エージェントは、目的が明確に定義されていて、徐々に環境に影響を及ぼしながら自分の目的を達成していきます。

続き

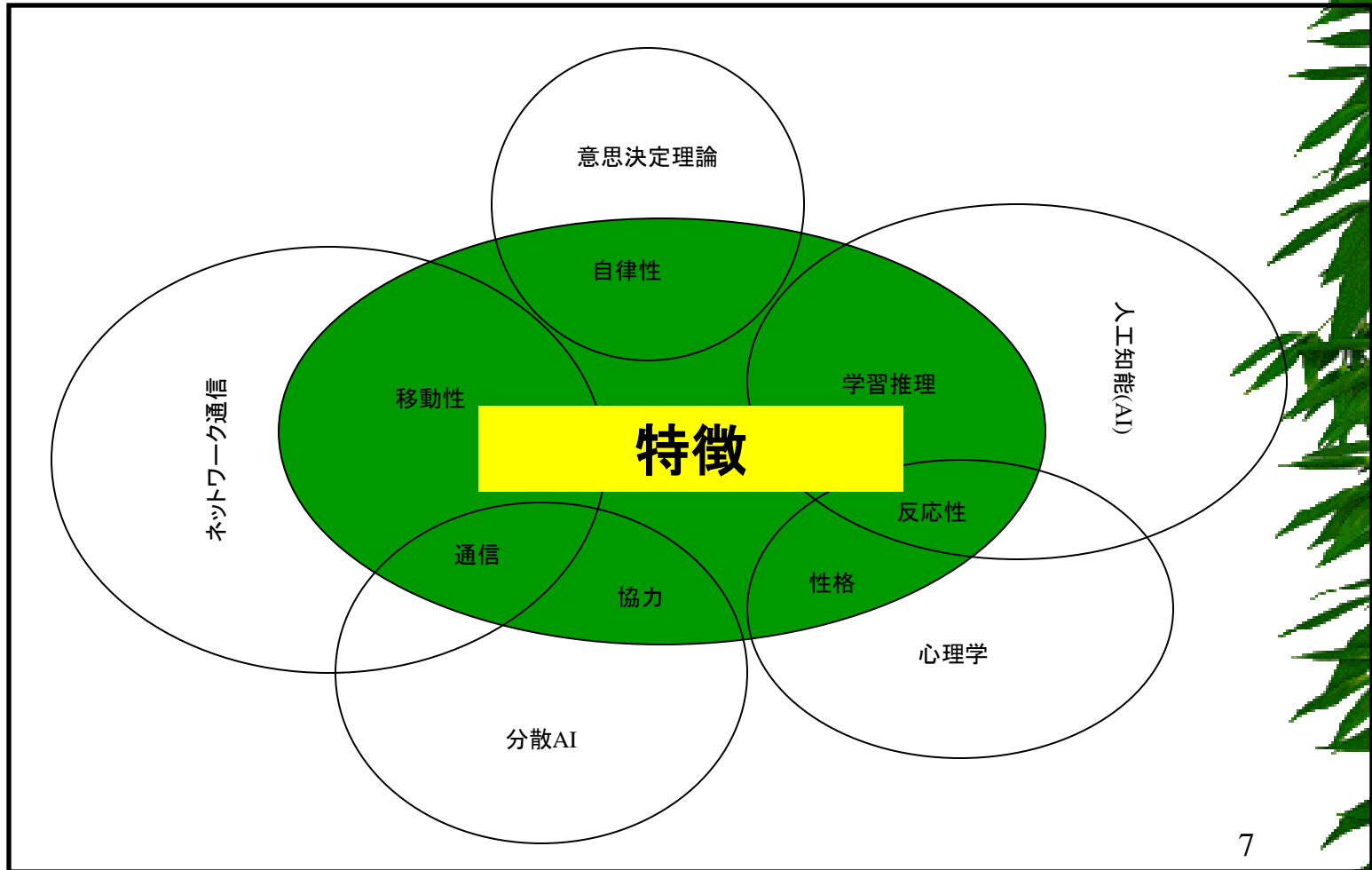
◆ 外部の特徴

- 通信:
エージェントは、タスクを成し遂げるために環境に対する、例えば人間、他のエージェントなどの任意の情報源などに対し相互作用をしばしば要求します。
- 協力:
複数のエージェントが協力することにより、単一のエージェントの能力を超えて複雑なタスクに対し、より早く、より良いソリューションを可能にします。
- 移動性:
エージェントは電子通信ネットワーク内を移動することができます。
- 性格:
人のように、エージェントは可能な限り多くの人間の性格に合った外的な振る舞いを示すかもしれません。

◆ 従来のソフトウェアプログラムとソフトウェアエージェントは何が違うのか？

- 認識: ソフトウェアエージェントは環境の変化を認識します。
- 自律性: ソフトウェアエージェントは主に自律的に仕事を果たします。
- 学習: ソフトウェアエージェントは知識を吸収(経験を通じて学習)します。
- 通信: ソフトウェアエージェントはユーザーや他のエージェントと通信します。

影響範囲



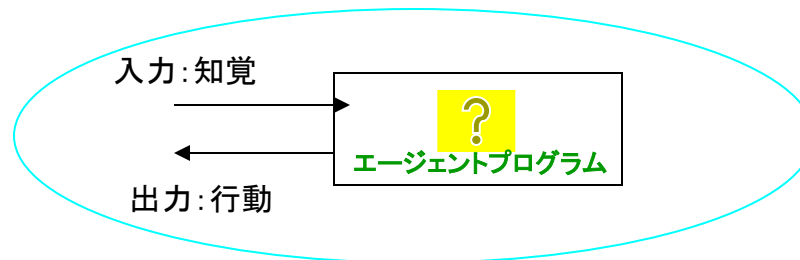
応用例

- ◆ 情報検索とフィルタリング
- ◆ ニュースの購読
- ◆ 友人発見支援エージェント
- ◆ 買い物支援エージェント
- ◆ スケジューリング支援エージェント
- ◆ Webドキュメント補修エージェント
-



どのようにエージェントをデザインするか

- ◆ エージェント = アーキテクチャ + エージェントプログラム
 - 一般的にアーキテクチャは以下の機能を持ちます。
 - ☞ 使用可能なセンサーからプログラムに知覚を生成する
 - ☞ プログラムを実行する
 - ☞ プログラムの行動の選択をエフェクターに供給する
 - アーキテクチャは以下のものです
 - ☞ 普通のコンピューター
 - ☞ 専用のハードウェア
 - ☞ いくつかのソフトウェア
 - エージェントプログラムは知覚から行動にエージェントのマッピングを実装する機能です。これはアーキテクチャ上で実行されます。



例：自動化されたタクシーのデザイン

知覚するもの	カメラ, 速度計, GPS, ソナー(超音波探知機)
行動するもの	ハンドル, アクセル, ブレーキ
目的	目的地までの安全に運転すること
環境	信号、他の車、歩行者、日本

タクシー運転エージェントとそのPAGEの説明

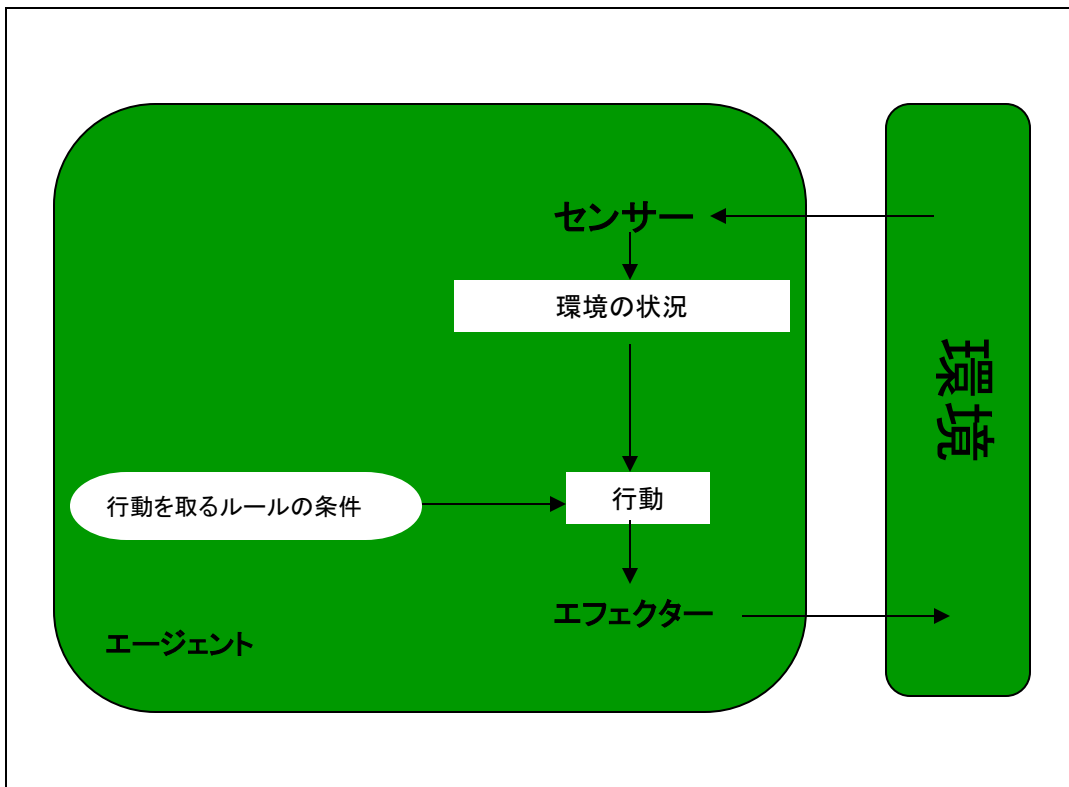
↑ Percepts, Actions, Goals, Environment

エージェントプログラムには以下の4つがあります

- 単純反射エージェント
- 環境保存エージェント
- ゴール主導エージェント
- 効用主導エージェント

単純反射エージェント

現在の状況がルール条件と一致すると、そのルールに関連付けられている行動を実行します。



```
function SRAgent(percept)
return action
```

```
static: rules, a set of condition-action rules
```

```
state <- INTERPRET-INPUT(percept)
```

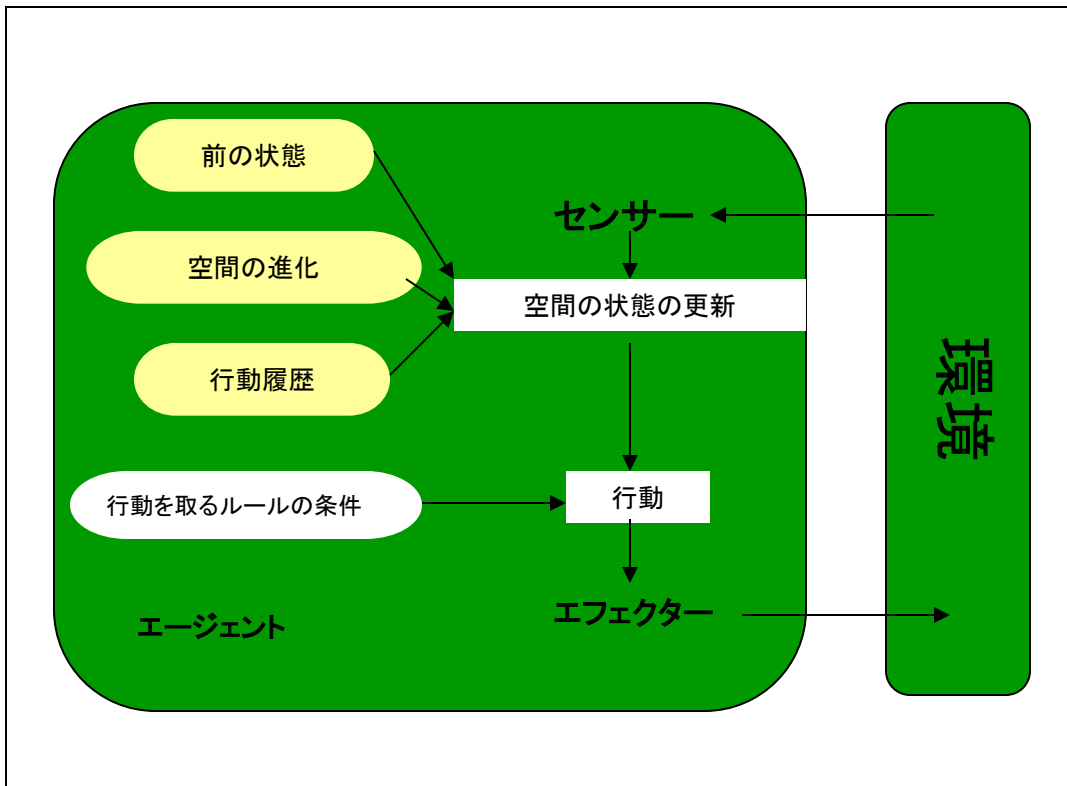
```
rule <- RULE-MATCH(state, rules)
```

```
action <- RULE-ACTION[rule]
```

```
return action
```

環境保存エージェント

(知覚と格納された内部状態によって定義された)現在の状況がルール条件と一致すると、そのルールに関連付けられている行動を実行します。(car model evolving)



```
function SRSAgent(percept) return  
action
```

```
static: current world state
```

```
rules, a set of condition-action rules
```

```
state <- UPDATE-STATE(state, percept)
```

```
rule <- RULE-MATCH(state, rules)
```

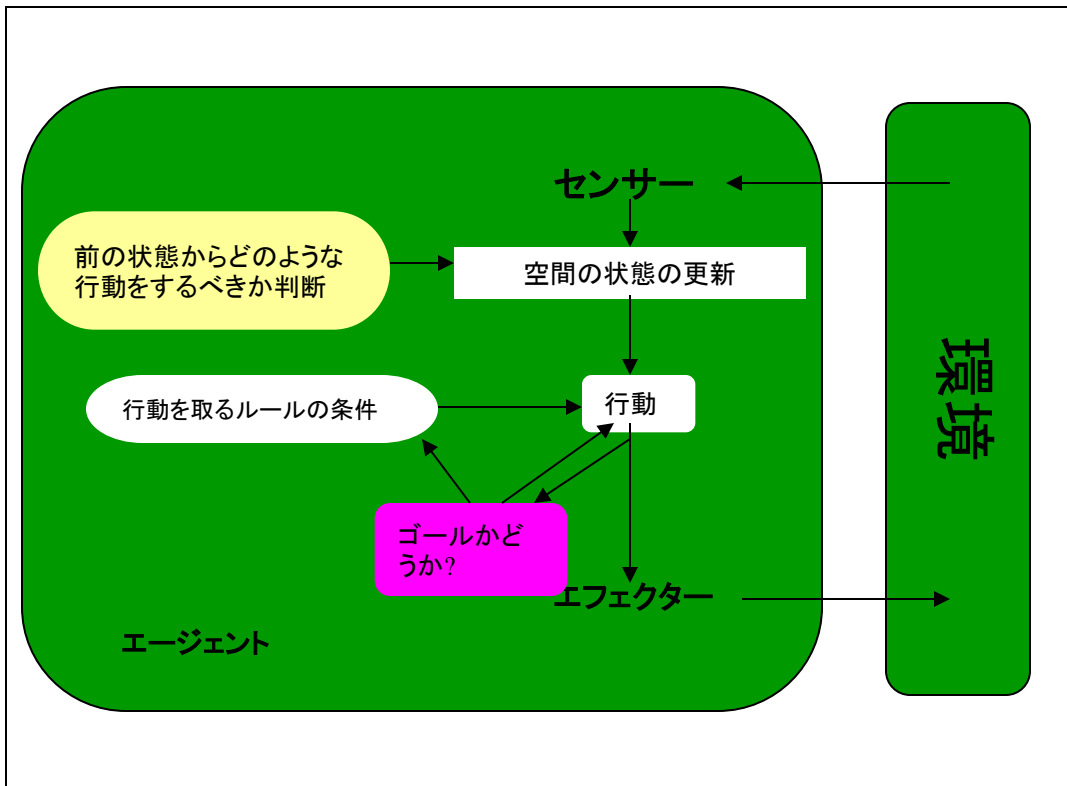
```
action <- RULE-ACTION[rule]
```

```
state <- UPDATE-STATE(state, action)
```

```
return action
```

ゴール主導エージェント

最終目標に向かって意思決定をしていきます



```
function Goal-Agent(percept) return  
action
```

```
static: current world state
```

```
rules, a set of condition-action rules
```

```
state <- UPDATE-STATE(state, percept)
```

```
do { rule <- RULE-MATCH(state, rules)
```

```
action <- RULE-ACTION[rule]
```

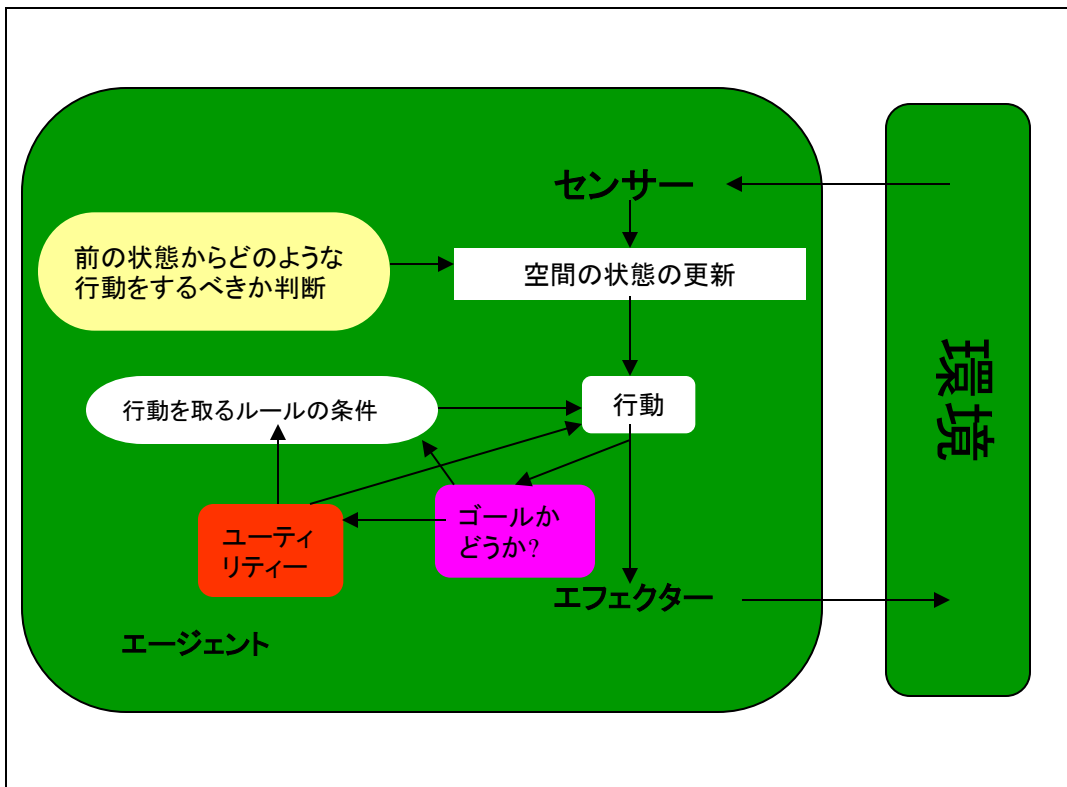
```
state <- UPDATE-STATE(state, action)
```

```
}while (not goals? <- state)
```

```
return action
```

効用主導エージェント

意思決定を、量で評価することで動作します。ユーティリティは、状態を質を表す実数にマッピングします。



```
function Utility-Agent(percept) return action
```

static: current world state

rules, a set of condition-action rules

```
state <- UPDATE-STATE(state, percept)
```

```
do { rule <- RULE-MATCH(state, rules)
```

```
action <- RULE-ACTION[rule]
```

```
state <- UPDATE-STATE(state, action)
```

```
} while ((not goals? <- state)|| (not good quality?))
```

```
return action
```

環境の特性

- ◆ アクセス可能 vs. アクセス不可能
- ◆ 決定論的 vs. 非決定論的
- ◆ 一時的 vs. 非一時的
- ◆ 静的 vs. 動的
- ◆ 離散 vs. 連続



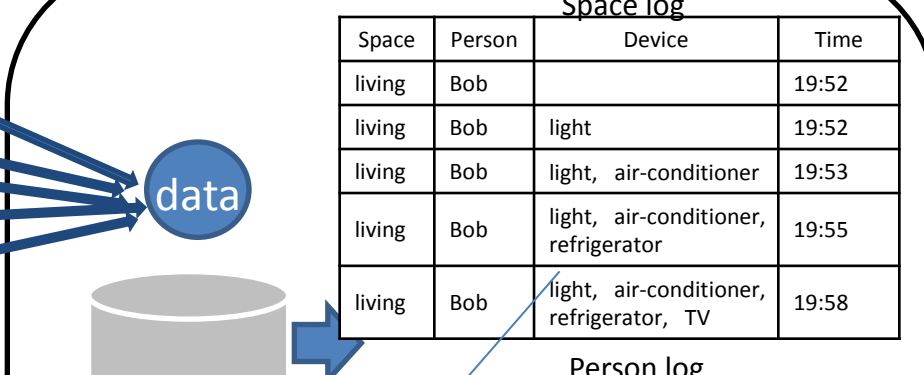
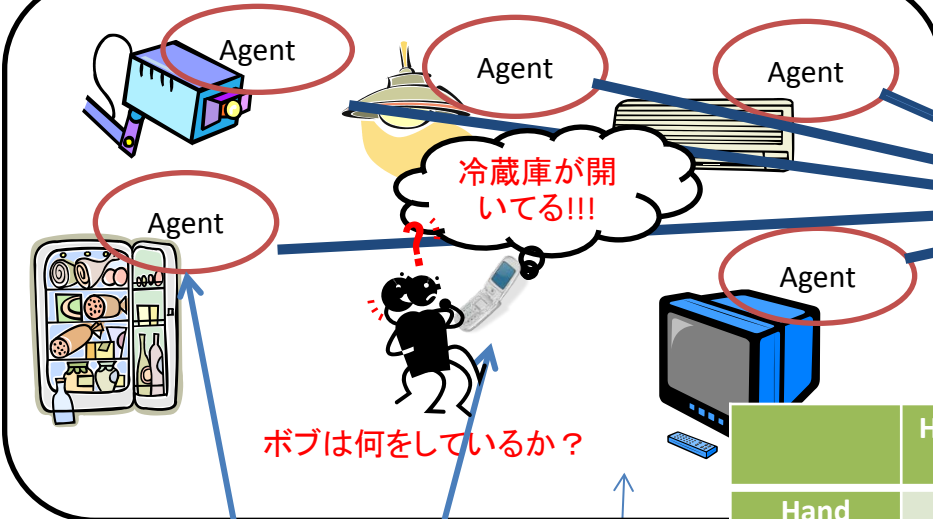
マルチエージェントシステム

- ◆ 田村さんのスマートエコホーム

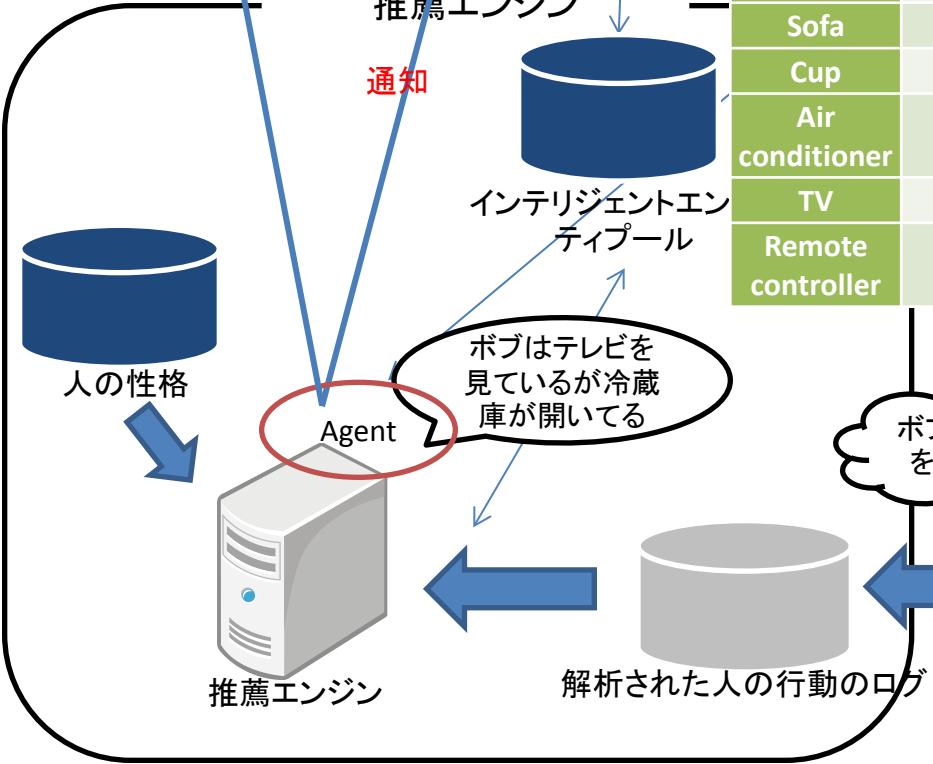


家のシミュレーション

データの集合



閉じるように提案



	Hand	Hip	Sofa	Cup	Air conditioner	TV	Remote controller
Hand	1	0	0	0	0	0	1
Hip	-	1	1	0	0	0	0
Sofa	-	-	1	0	0	0	0
Cup	-	-	-	1	0	0	0
Air conditioner	-	-	-	-	1	0	1
TV	-	-	-	-	-	1	1
Remote controller	-	-	-	-	-	-	1

ボブはテレビを見ている

人の行動の抽出

	Hand	Hip	Sofa	Cup	Air conditioner	TV	Remote controller
Hand	1	0	0	0	0	0	0
Hip	-	1	1	0	0	0	0
Sofa	-	-	1	0	0	0	0
Cup	-	-	-	1	0	0	0
Air conditioner	-	-	-	-	1	0	0
TV	-	-	-	-	-	1	0
Remote controller	-	-	-	-	-	-	1

Drinking

	Hand	Hip	Sofa	Cup	Air conditioner	TV	Remote controller
Hand	1	0	0	0	0	0	0
Hip	-	1	1	0	0	0	0
Sofa	-	-	1	0	0	0	0
Cup	-	-	-	1	0	0	0
Air conditioner	-	-	-	-	1	0	0
TV	-	-	-	-	-	1	0
Remote controller	-	-	-	-	-	-	1

Watching TV

実世界でのアプリケーション

◆ デモ

- 探索問題の解決
- ゲーム: コンピューターがプレイヤーに
- スマートガーデン
- 宝を見つけるエージェント
- 迷路ロボット: 出口を発見する
- 自動的な会話



Work in Class

- ◆ ホームページにある1～6のアプレットを実行してみましょう
<https://cis.k.hosei.ac.jp/~rhuang/>

Read:

<http://www.popsci.com/cars/article/2013-09/google-self-driving-car>

<http://www.northeastern.edu/news/2012/05/driverless-car/>

Think:

Image if there are all driverless cars are on road in future, what may be the problems and how to solve the problems?

もし将来すべての車が自動で運転されるようになったら何が問題になるか、またその問題をどのように解決するか想像してみましょう。

Home work

The Google driverless car is a project by Google that involves developing technology for autonomous cars. The software powering Google's cars is called Google Chauffeur.

Google自動運転カープロジェクトはグーグルによる車の自動運転技術に関連したプロジェクトです。

この車を制御しているソフトウェアを"Google Chauffeur"といいます。

Q: How does AI technology play important role in developing software Google Chauffeur? Please write a report and submit it on **10/24** (before lecture).

"Google Chauffeur"において、AI技術はどのような重要な役割を果たしていますか？レポートにまとめて**10月24日**の講義が始まる前に提出してください