神戸大学大学院人間発達環境学研究科 数理情報環境論コース

デジタルゲームにおける人工知能の構造と運動

三宅　陽一郎

y.m.4160@gmail.com

http://www.facebook.com/youichiro.miyake

twitter: @miyayou

2013.3.11

神戸大学

感謝と自己紹介

- 桐生先生、長坂先生、ありがとうございます。
- １９７５年生まれ　兵庫県芦屋市出身
- 自己紹介
  2004-2011　AI Programmer (FROM SOFTWARE)
  2011-Present　Lead AI Researcher (SQUARE ENIX)

  2007-Present　国際ゲーム開発者協会 日本支部 SIG-AI チェア（ IGDA JAPAN SIG-AI)
  2008-Present　日本デジタルゲーム学会（DiGRA JAPAN）研究委員
  2011-Present　CEDEC （日本ゲームカンファレンス）委員
- 所属学会
  人工知能学会、AAAI, IEEE CIS, ACM, 日本デジタルゲーム学会（DiGRA JAPAN）

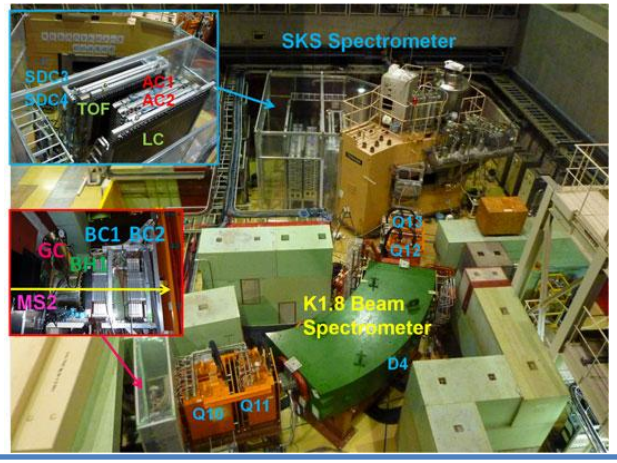
  情報処理学会 EC研究会 委員→アドバイザー（何もしなかった）
  ACM マルティメディア国際学会 委員（何もしなかった）
  情報処理学会　AIコンテスト　SamurAI Coding 委員　（これから頑張ります）
  IEEE WCCI 2014 (Computational Intelligence 国際会議)　委員　（これから頑張ります）
  経済産業省 DCEXPO 委員（ちょっと頑張った）

学歴

京都大学（数学）

大阪大学（原子核実験物理）

東京大学
（エネルギー工学/人工知能）



高エネルギー加速器研究所（半年ぐらい。修士論文）

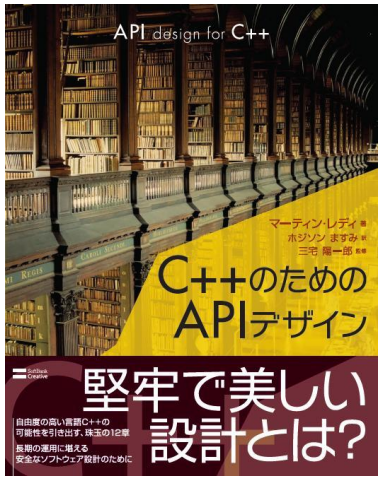
http://www.facebook.com/youichiro.miyake

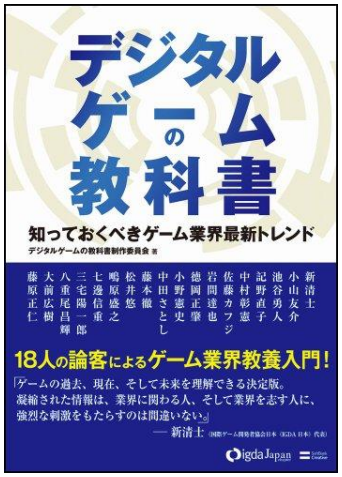
Works (2006-2012)

Chrome Hounds (2006)
Demon's Souls (2009)
PokaPoka Airu Village (2011)
Armored Core V (2012)

AI for Game Titles

Books





ゲーム開発者(＝三宅)は何を考えてゲームを作っているか？

- ゲームを作って人のために役立つことが僕たちの使命だ。
- 電気やガスやエスカレーターみたいに、直接役立つって難しいけど、ゲームが美しいビジョンを伝えたり、嫌なことを吸収してあげたりして、人の心の状態を綺麗に保つことで、社会をよくしているのだと信じている。

ゲーム開発者(＝三宅)は何を考えてゲームを作っているか？

- ゲームに命を吹き込みたい。
- エンジニアとしてはゲームに命を吹き込む方法をクリエーターに提供したい。
- AIエンジニアとしては、キャラクターに命を吹き込みたい。

ということで、今回は、キャラクターAIに命を吹き込む技術、ゲームに命を吹き込む技術を解説して行きます。

今日の内容

第一章

AIはどのように世界を結びつくか？

第二章

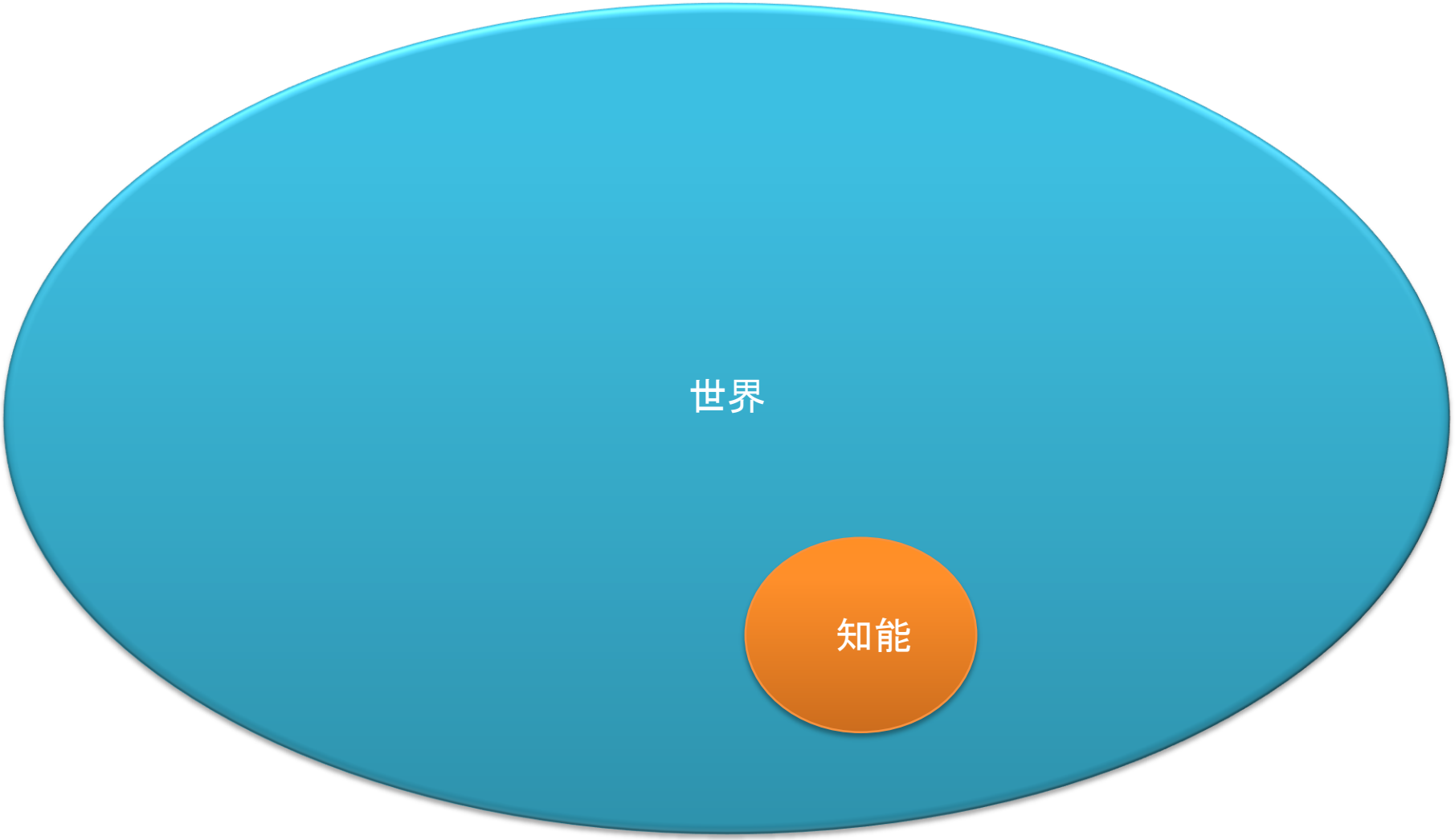
エージェントモデル

第三章

行動表現と世界表現の双対性

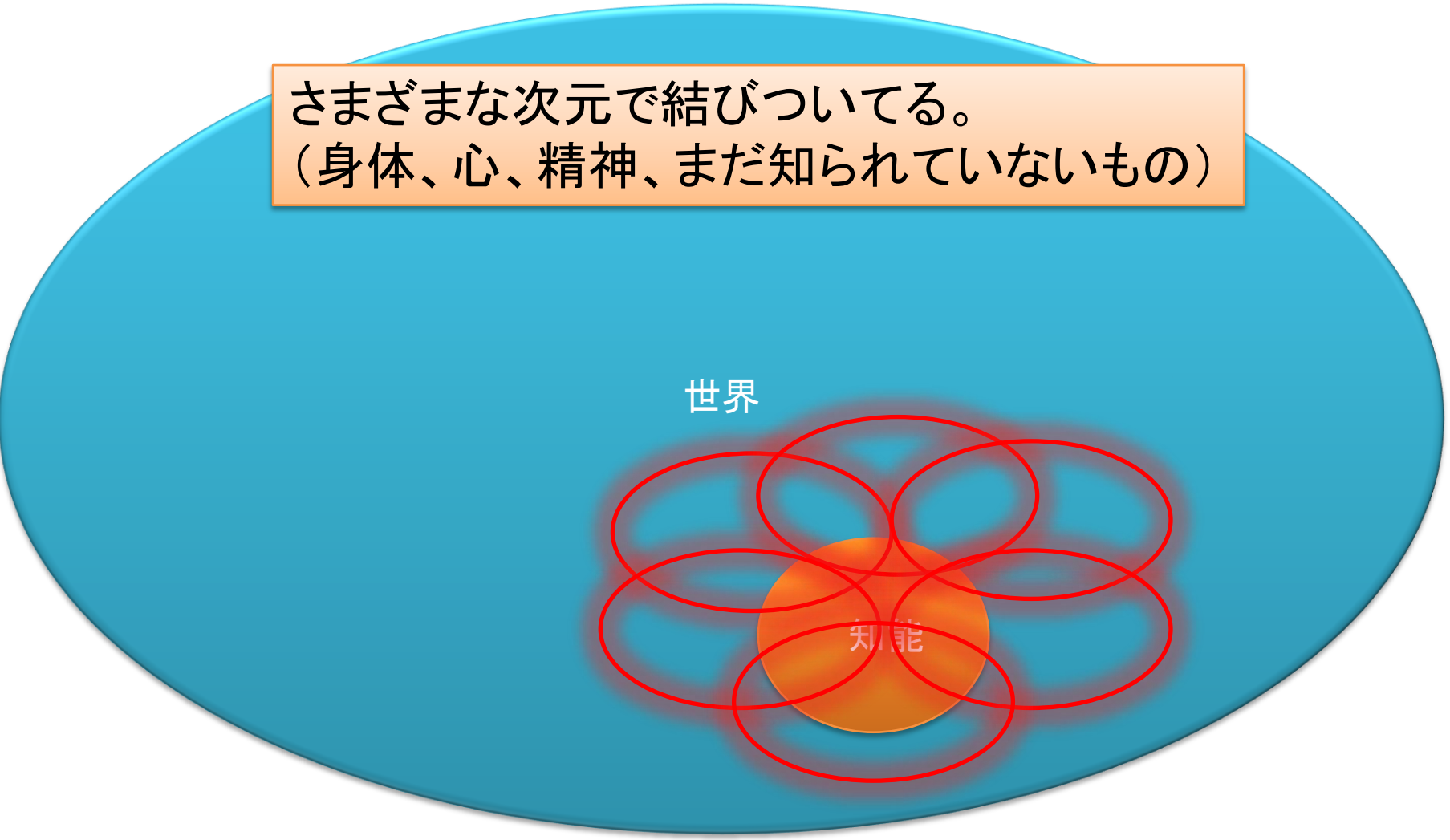
第一章
AIはどのように世界を結びつくか？

テーマ：人工知能はどのように世界と結びつくか？



テーマ：人工知能はどのように世界と結びつくか？

さまざまな次元で結びついてる。
（身体、心、精神、まだ知られていないもの）



テーマ：人工知能はどのように世界と結びつくか？

むしろ関係の集積が存在を作っている。

知能と世界の結び付きはいつできたんだろう？

- 生命は何万年という進化の中で、世界の中で生まれてきた。
- 生命はむしろ孤立しているのではなくて、そもそも世界との結びつきの上で成立している。
- 高度から低次までさまざまな結びつきがある。

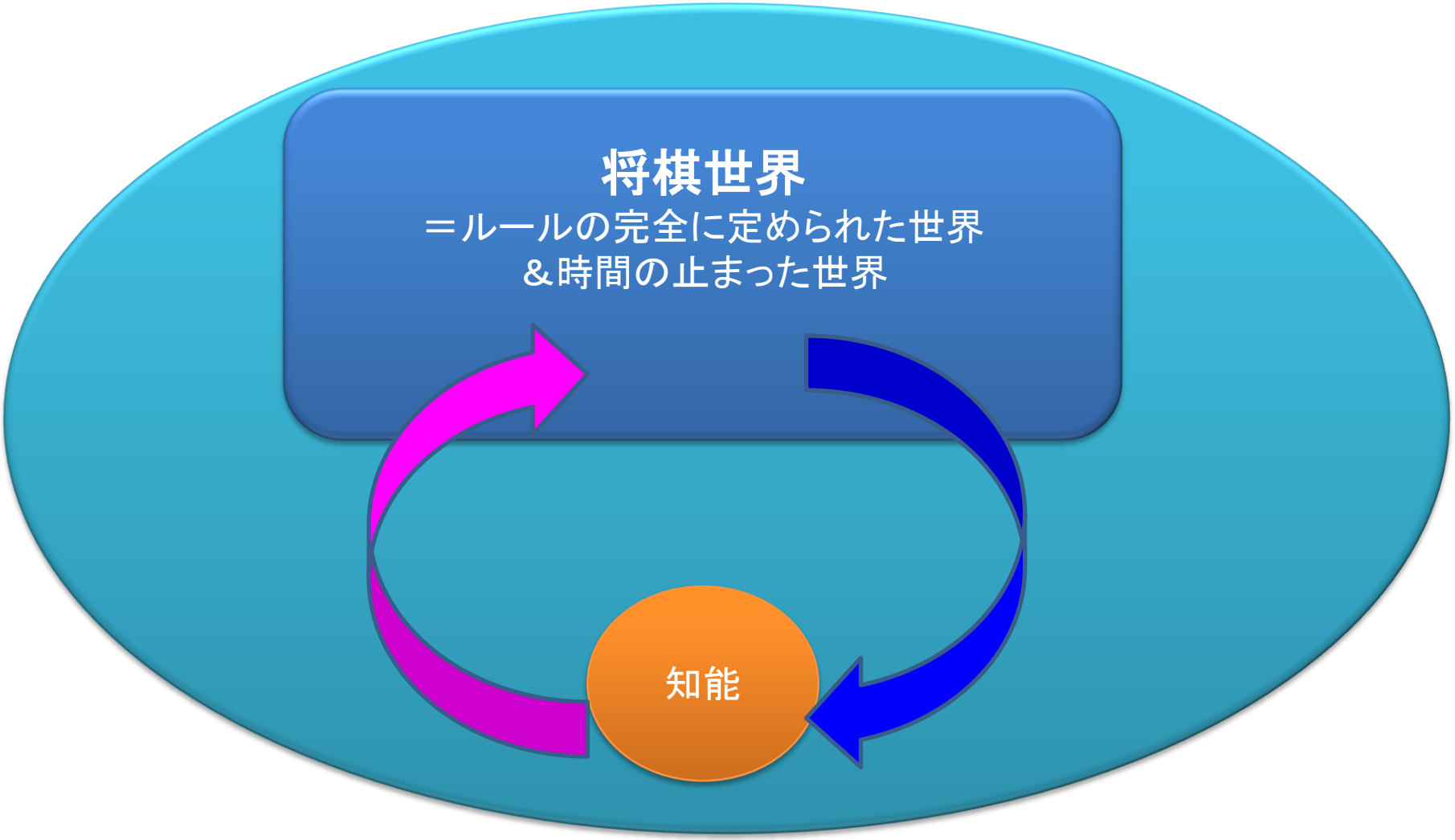
知能と世界の結び付きはいつできたんだろう？

- 生命は何万年という進化の中で、世界の中で生まれてきた。
- 生命はむしろ孤立しているのではなくて、そもそも世界との結びつきの上で成立している。
- 高度から低次までさまざまな結びつきがある。

人工知能はどうだろうか？
人工知能は世界とどのような結びつきを獲得しているだろうか？

知能と世界の結びつきのあり方を探求するのも人工知能の重要なテーマ

人工知能初期の研究



初期のゲームAIの研究(1970-)

世界をボードゲーム（チェス、囲碁、将棋）に見立てて、人工知能を探求する。

閉じた問題。フレームの回避。

完全にロジカルな問題　＝純粋なアルゴリズムの探究

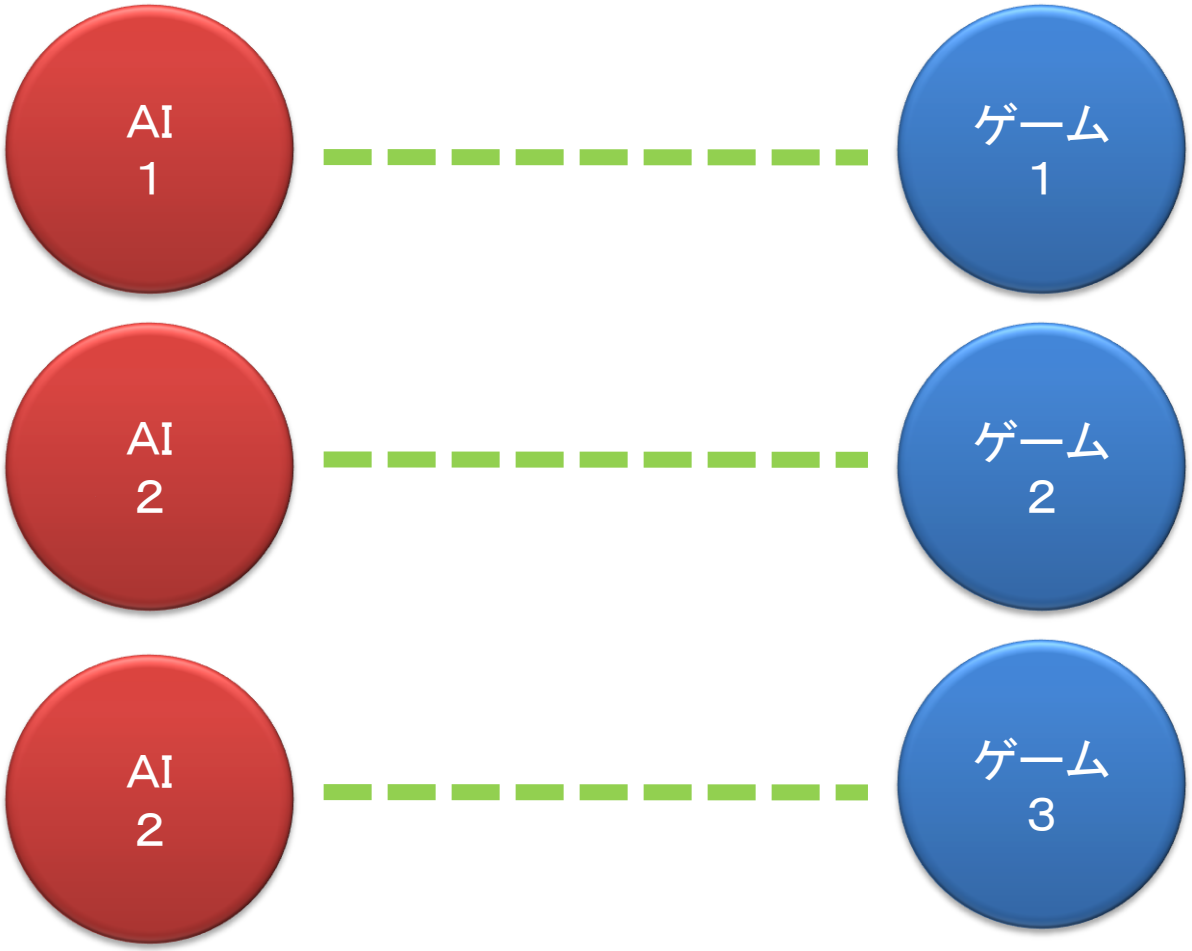
人工知能から情報科学の課題へ移行（アルゴリズムの探究）

問題の表現を問い直すことはない。

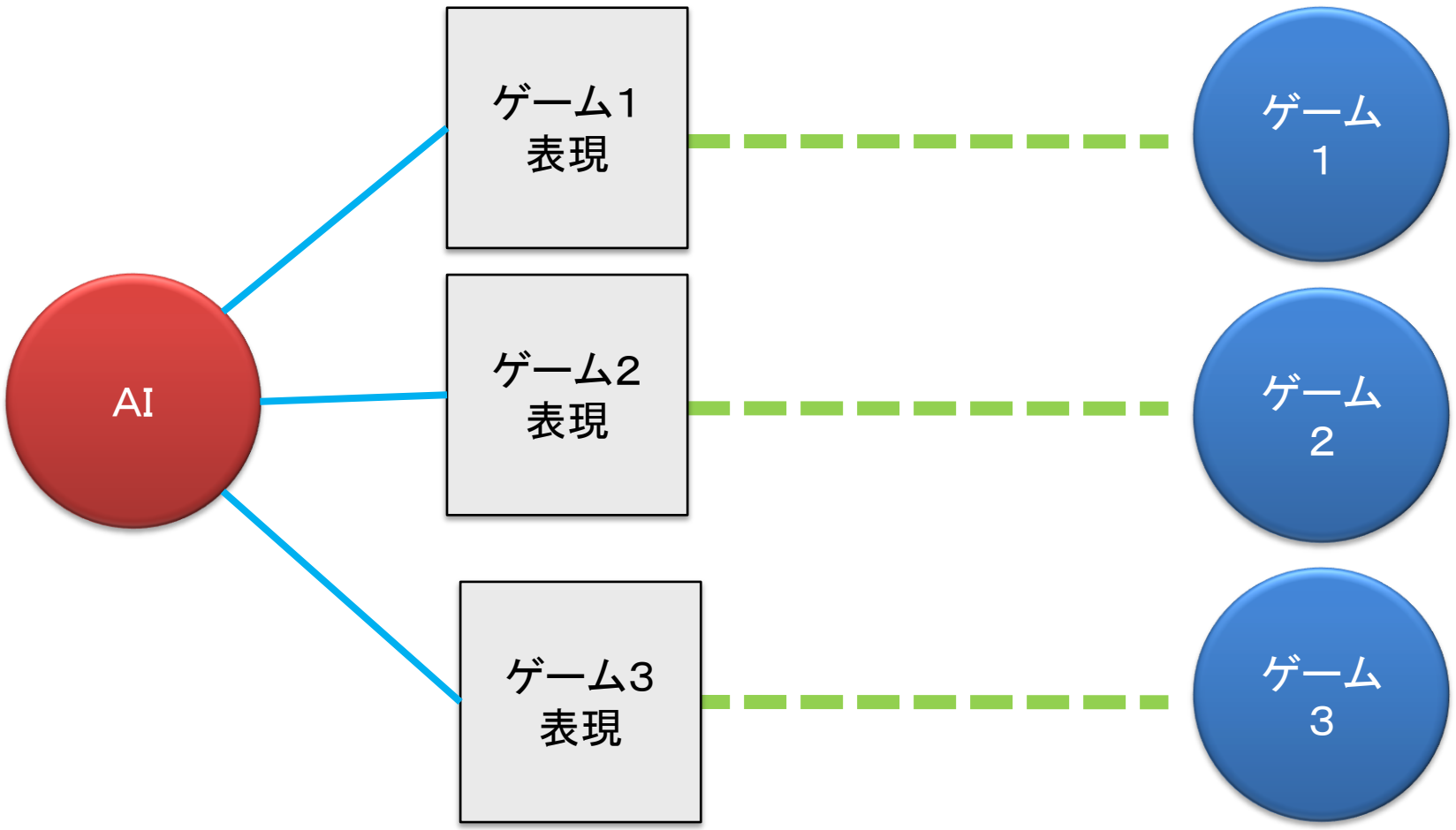
一般化ゲームAIの探究
(General Game Playing)

- なぜ一つのゲームだけなのか？
- ゲームのルールを明示的に理解できないAIなのではないか。
- ゲームのルール自体を理解できるAIを作るべきではないのか？
- そうすれば、いろいろなゲームを討てるAIが作れるのではないか。

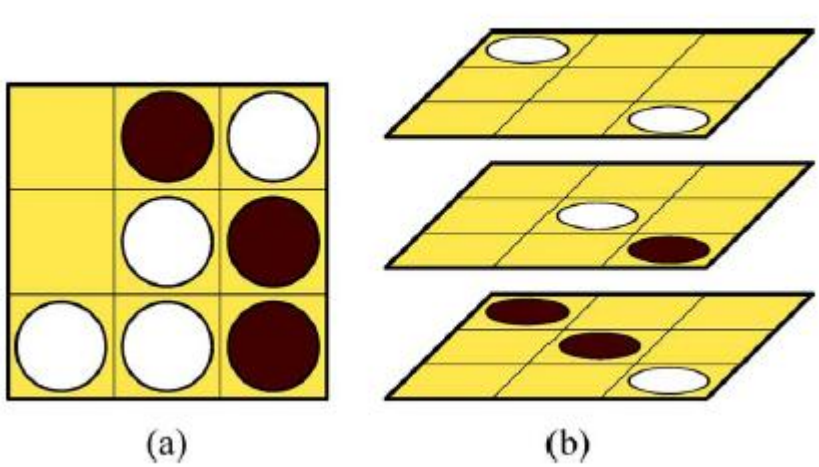
ゲームごとのAI

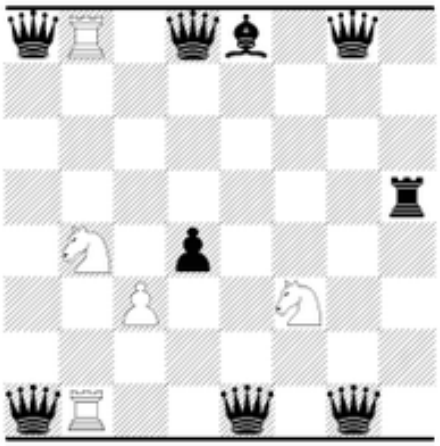


どんなゲームにおけるAI
（General Game Playing）



そのためには？





いろいろなゲーム自身　（＝ルール、ステージ）を汎用的に記述して、AIに理解させてやらなければならない。

ゲームを定義する。

Encoding and generating videogame mechanics
http://www.kmjn.org/notes/generating_mechanics_bibliography.html

General Game Playing

チェスのプログラムはチェスしかプレイできない。
将棋のプログラムは将棋しかプレイできない。

どんなゲームもプレイできるAI（＝ゲームを超えたAI）を作ろう。

Ludi システム

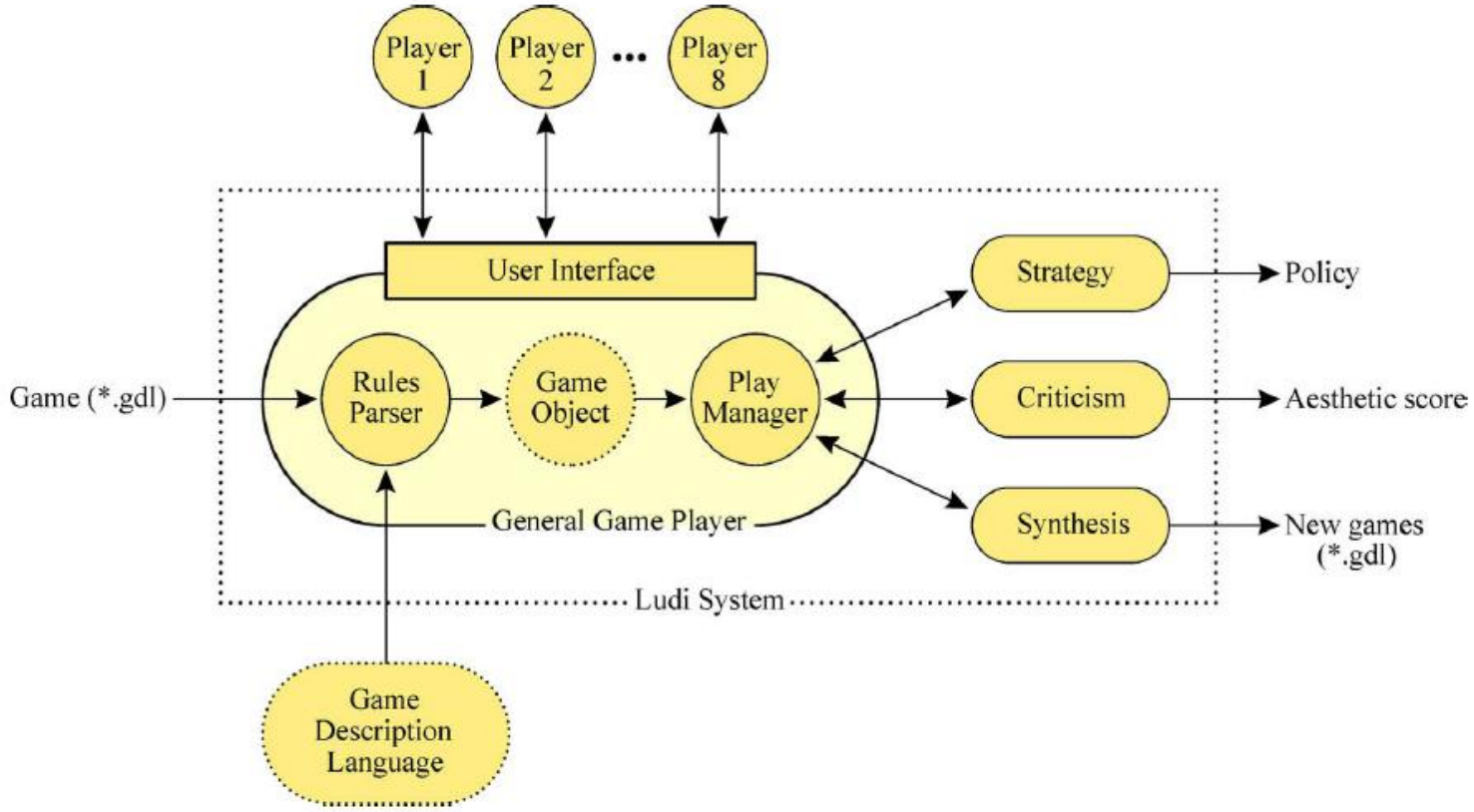


Fig. 2. Framework of the Ludi system.

Cameron Browne & Frederic Maire (2010). **Evolutionary game design.** *IEEE Transactions on Computational Intelligence and AI in Games* 2(1): 1-16.

Ludi システム

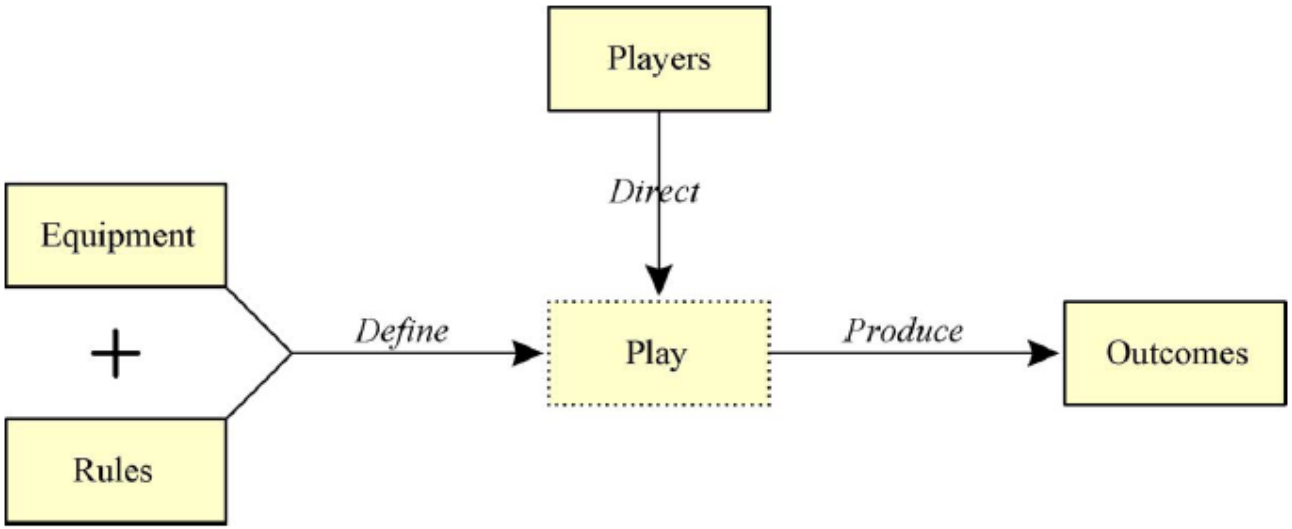


Fig. 3. Basic game model.

Cameron Browne & Frederic Maire (2010). **Evolutionary game design.** *IEEE Transactions on Computational Intelligence and AI in Games* 2(1): 1-16.

Ludi システム

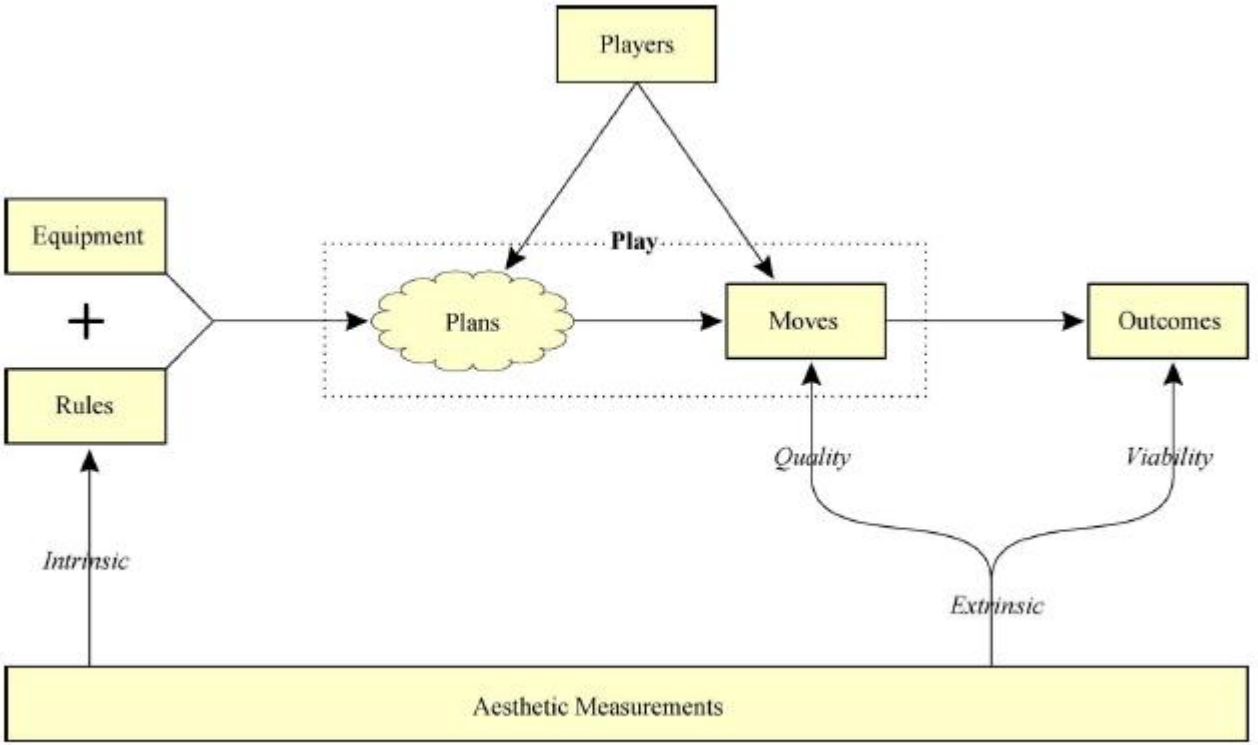


Fig. 6. Player-centric aesthetic model of games.

Cameron Browne & Frederic Maire (2010). **Evolutionary game design.** *IEEE Transactions on Computational Intelligence and AI in Games* 2(1): 1-16.

ゲームそのものを進化させる

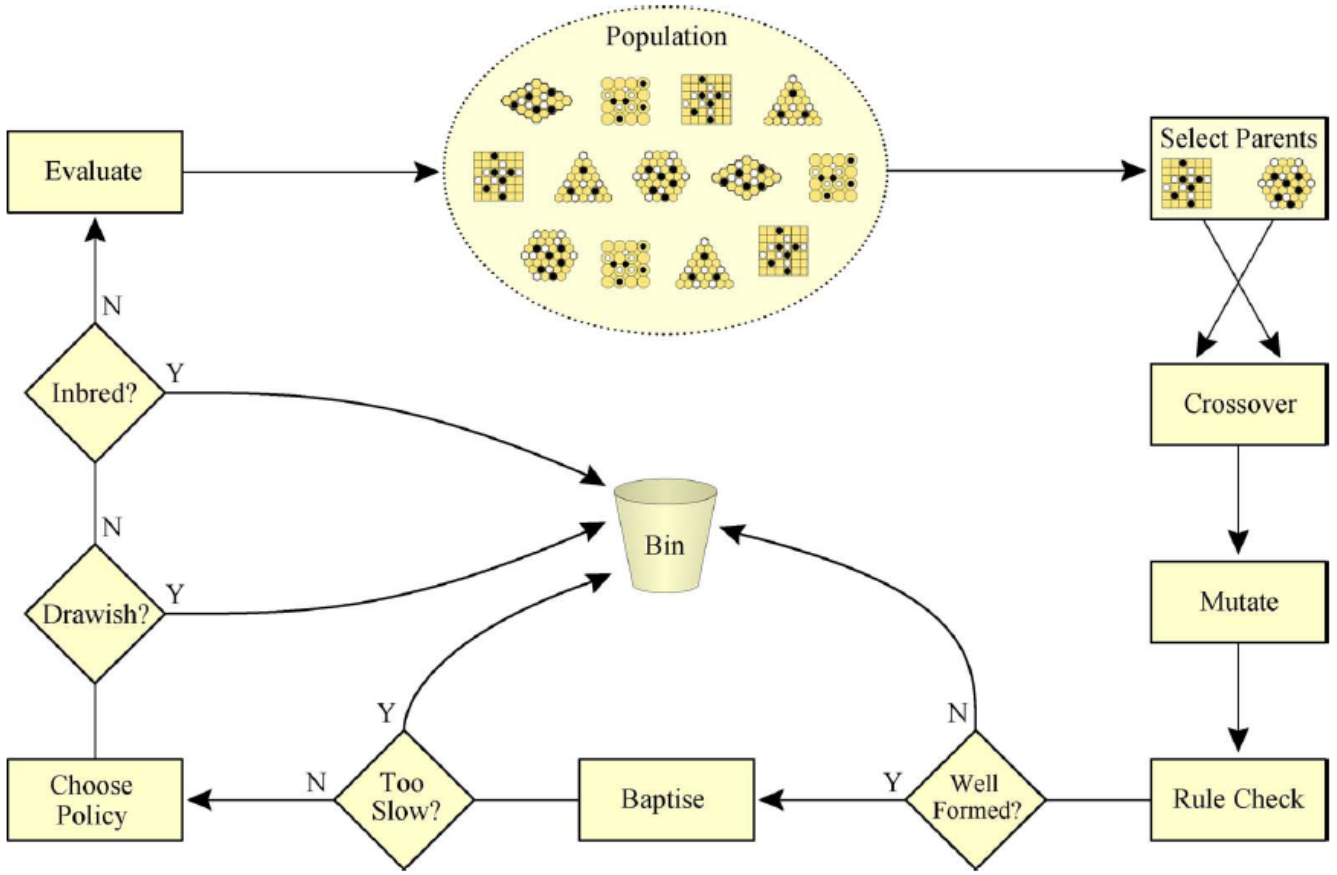


Fig. 9. Game life cycle.

素材から新しいゲームを生成する。

Cameron Browne & Frederic Maire (2010). **Evolutionary game design.** *IEEE Transactions on Computational Intelligence and AI in Games* 2(1): 1-16.

ゲーム自体を表現する


```
GAME            american_checkers
GOALS           stalemate opponent
BOARD_SIZE      8 BY 8
BOARD_TYPE      planar
PROMOTE_RANK 8
SETUP           man AT {(1,1)(3,1)(5,1)(7,1)(2,2)(4,2)
                        (6,2)(8,2)(1,3)(3,3)(5,3)(7,3)}
CONSTRAINTS     must_capture

DEFINE man
  MOVING
    MOVEMENT
      LEAP
      (1,1) SYMMETRY {side}
    END MOVEMENT
  END MOVING
  CAPTURING
    CAPTURE
      BY {hop}
      TYPE [{opponent} any_piece]
      EFFECT remove
      MOVEMENT
        HOP BEFORE [X = 0]
            OVER   [X = 1]
            AFTER  [X = 0]
        HOP_OVER [{opponent} any_piece]
        (1,1) SYMMETRY {side}
      END MOVEMENT
    END CAPTURE
  END CAPTURING
  PROMOTING
    PROMOTE_TO king
  END PROMOTING
  CONSTRAINTS continue_captures
END DEFINE

DEFINE king
  MOVING
    MOVEMENT
      LEAP
      (1,1) SYMMETRY {forward side}
    END MOVEMENT
  END MOVING
  CAPTURING
    CAPTURE
      BY {hop}
      TYPE [{opponent} any_piece]
      EFFECT remove
      MOVEMENT
        HOP BEFORE [X = 0]
            OVER   [X = 1]
            AFTER  [X = 0]
        HOP_OVER [{opponent} any_piece]
        (1,1) SYMMETRY {forward side}
      END MOVEMENT
    END CAPTURE
  END CAPTURING
  PROMOTING
    PROMOTE_TO king
  END PROMOTING
  CONSTRAINTS continue_captures
END DEFINE

END GAME.
```

Figure 3: Definition of *American Checkers* as a symmetric chess-like game.

Barney Pell (1992). **METAGAME in symmetric chess-like games.** In *Heuristic Programming in Artificial Intelligence 3 – The Third Computer Olympiad*, Ellis Horwood.

新しいチェスのゲーム

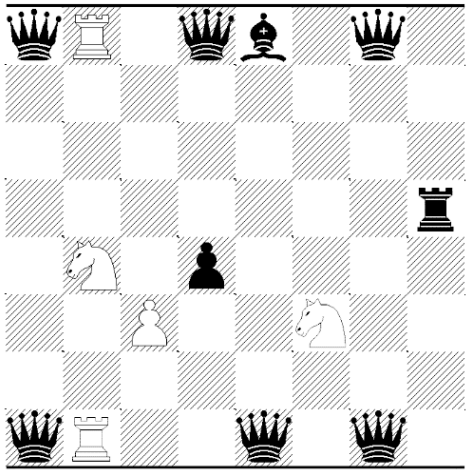


Figure 1: A *vertical-cylinder* board and capture movements

A Class Definition

This grammar for *symmetric chess-like games* is presented in an extended-*BNF* notation. Capitalised and quoted words are terminal symbols, except for IDENTIFIER and NUMBER, which stand for any identifier and any number, respectively. Text in unquoted braces is optional. The grammar is not case-sensitive, so that game definitions may use uncapitalised words for clarity.

Comments can appear within game definitions. Comments begin with percent symbols ('%') and end with the start of a new line.

```
game --> GAME IDENTIFIER
         goal_defs
         board
         SETUP assignment_list
```



```
         {CONSTRAINTS MUST_CAPTURE}
         piece_defs
         END GAME '.'

goal_defs --> GOALS goals

goals -->  goal | goal goals

goal --> ARRIVE description AT square_list
       | ERADICATE description
       | STALEMATE player

description --> '[' player_gen piece_names ']'

player_gen --> '{' player '}' | ANY_PLAYER

player -->  PLAYER | OPPONENT

piece_names --> '{' identifiers '}'
              | ANY_PIECE

identifiers --> IDENTIFIER | IDENTIFIER identifiers

square_list --> '{' squares '}'

squares --> square | square squares

square --> '(' NUMBER ',' NUMBER ')'

board --> BOARD_SIZE NUMBER BY NUMBER
          BOARD_TYPE board_type
          PROMOTE_RANK NUMBER

board_type --> PLANAR | VERTICAL_CYLINDER
```

Barney Pell (1992). **METAGAME in symmetric chess-like games.** In *Heuristic Programming in Artificial Intelligence 3 – The Third Computer Olympiad*, Ellis Horwood.

新しいパズルゲーム

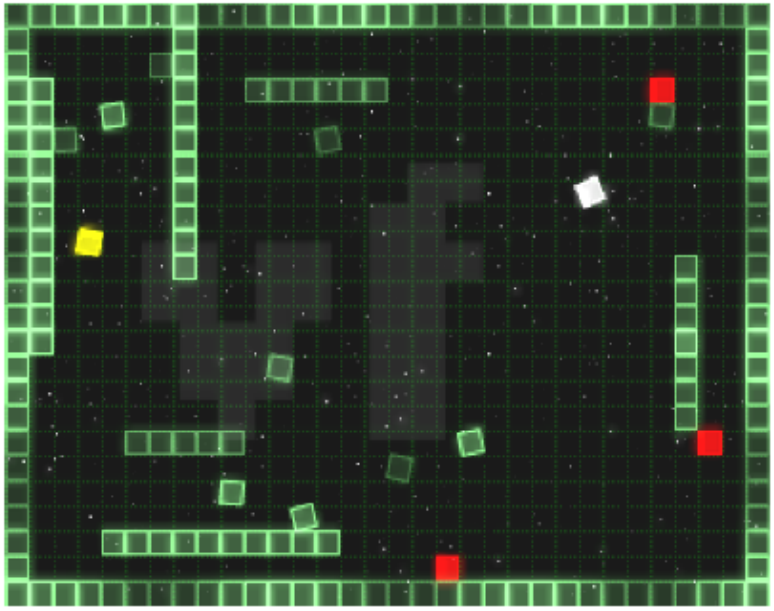


Fig. 1. **Screenshot of gameplay for a generated mini-game in the *Variations Forever* prototype.** The player controls the white character using an Asteroids-inspired movement model, trying to touch all red characters which move via a Pac-Man-inspired movement model. The encircling walls, random-walls and random-blocks algorithms have generated dangerous obstacles which can harm the player's square. This particular game's rules also define the stars and grid backdrop details as well as a third kind of character (yellow) which drifts on its own.

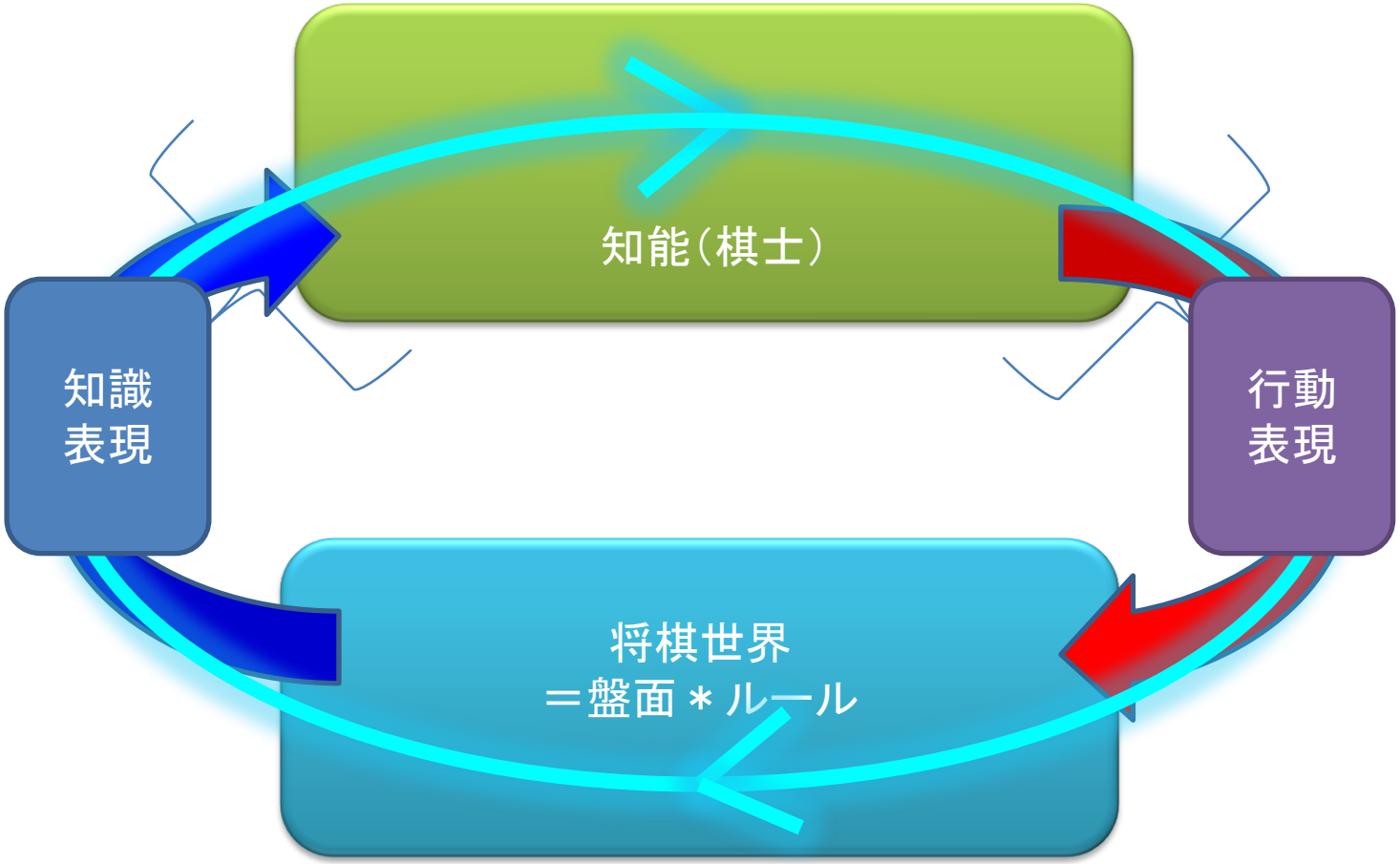
Fig. 1. The example game at $t = 0$. The cyan circle is the agent, the other circles are things of different colours. Note that all the thing positions are randomized, whereas the layout of the walls is hard-coded and subject to neither evolution nor changes during gameplay.

Aam M. Smith & Michael Mateas (2010). **Variations Forever: Flexibly generating rulesets from a sculptable design space of mini-games.** In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pp. 273-280.

Julian Togelius & Jürgen Schmidhuber (2008). **An experiment in automatic game design**.
In *Proceedings of the 2008 IEEE Conference on Computational Intelligence and Games*, pp. 111-118.

将棋

- どのように運動が知覚と世界を形成していくか。



http://userdisk.webry.biglobe.ne.jp/005/038/27/5/todai8_001.jpg

将

- どのように運動が知覚

（駒）

知識
表現

行動
表現

将棋世界
＝盤面＊ルール

http://yamagata6.blog118.fc2.com/blog-entry-316.html

http://japan5374.seesaa.net/article/39641352.html

世界内、世界外

- 騎士 ＝ 世界外 ＝ 世界を俯瞰する
- 駒＝世界内 ＝ 世界を体験する。

スーパーマリオAIコンテスト

(1) 国際学会併設のAIコンテスト

※最近、海外ではデジタルゲームにおける人工知能技術の研究が盛んになっており、大きな大会では、論文発表と並行してコンテストが行われる場合が多い。ほとんどは、大学の研究室やゲーム企業などからエントリーされる。

(2) 毎年、恒例化

http://www.marioai.org/

(3) 目的:　デジタルゲームの題材を提供する
＝研究を実際のゲームに結びつける
＝デモンストレーション、産業へアピール

競技設定

(1) Infinite Mario によって、ステージは自動生成

(2) マリオは決められた周囲の環境の情報を取って来ることができる。

Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

優勝作品：Robin Baumgarten (2K Games)

アイデア

- スーパーマリオの全ての敵の挙動を解析して数式に落しておく。
- 数式に基づいて、常に次の環境（世界）の状態を予測する。
- 各ノードの適合値を予測に基づいて計算する。
- A*のヒューリスティック関数：目的の場所までにかかる時間
- 穴に落ちたり、ダメージを食らったりするケースはペナルティを加える。
- コイン、敵撃墜、パワーアップはとりあえず今のところ無視。

Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

A* アルゴリズム

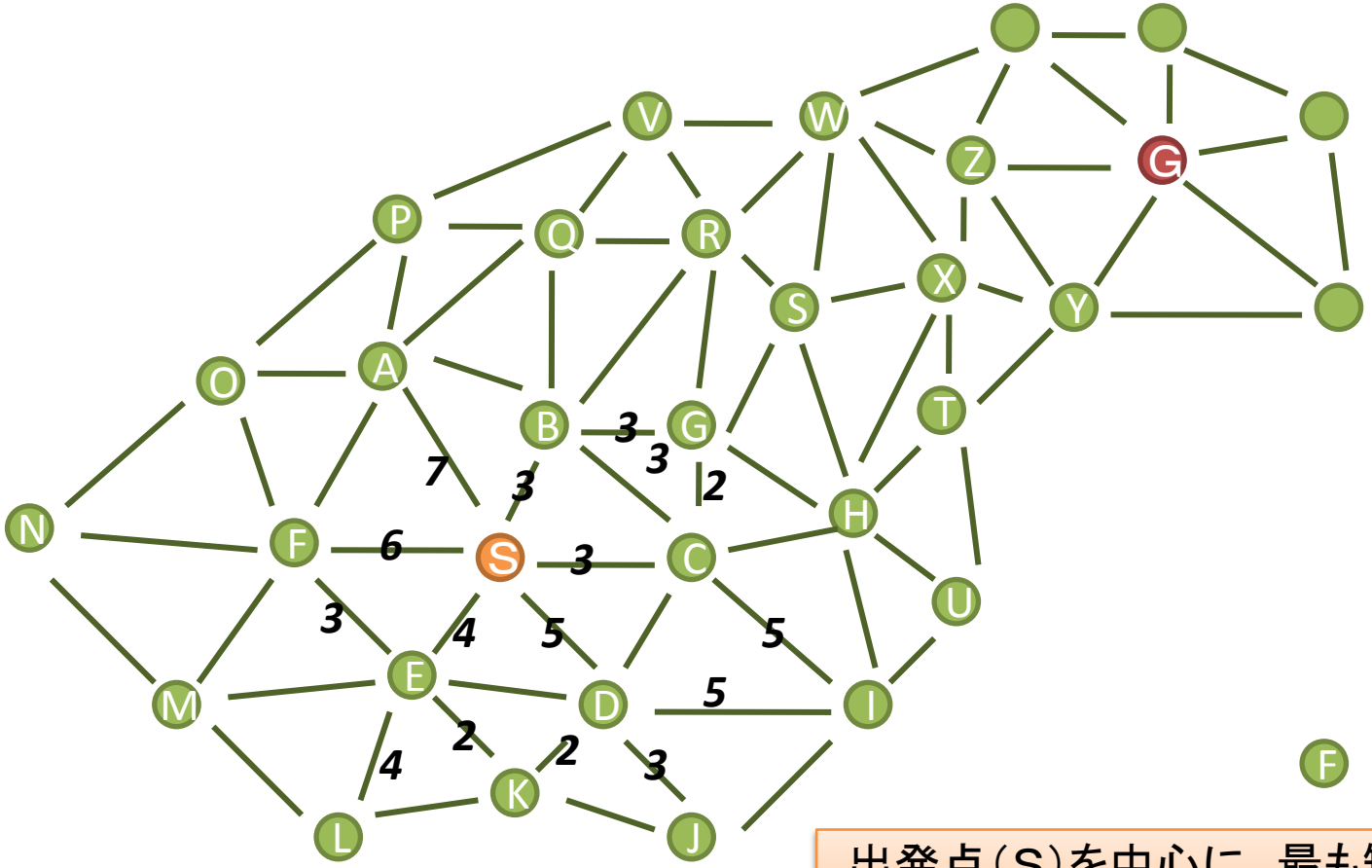
- Best-First Search アルゴリズム
- ゴールまでの残りコストを推定するヒューリスティック関数
- オープンノード群を保持する(最初はスタートノードだけ)
- オープンセットが空でない間
  - *オープンセットの中でトータル推定コストが最も低いノードをピックアップする*
  - *ノード＝ゴールになれば、その経路が求めるべき経路。*
  - *子ノードをオープンセットに追加（もし既存のノードでも、経路推定がよりよければ、入れ替える）*
- ヒューリスティック関数が適切ならば、正解を求めることができる。（ヒューリスティック関数は実際より下に見積もるものを取ることが大切）

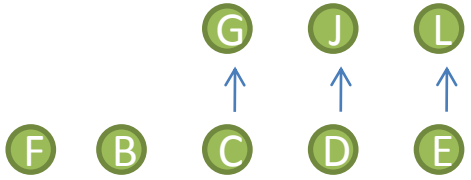
Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

ネットワーク上のグラフ検索法

ダイクストラ法

各ノードの評価距離＝出発点からの経路



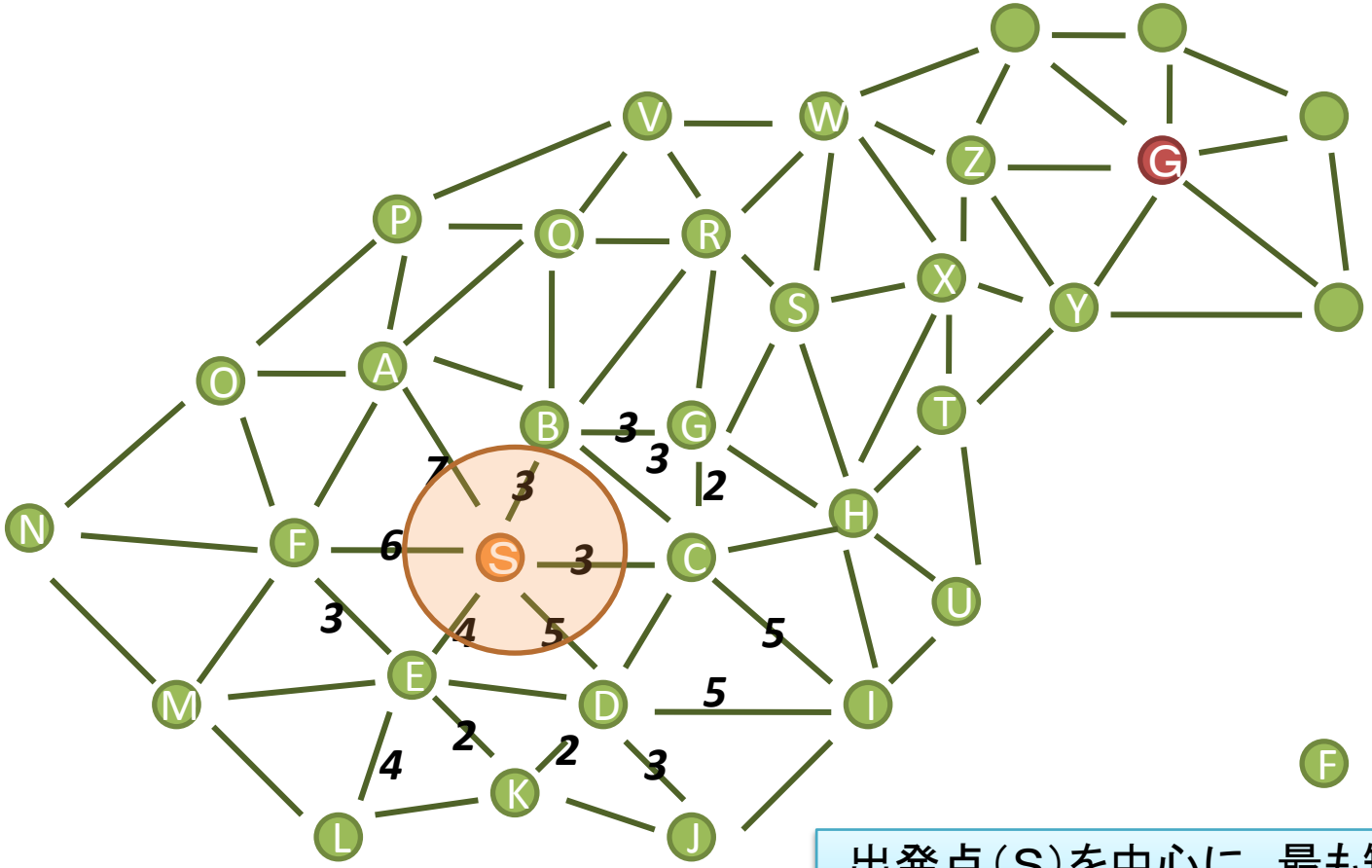


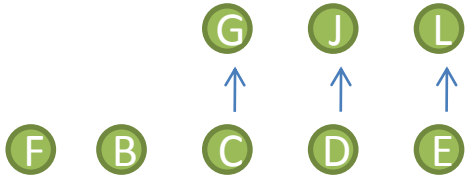
出発点（S）を中心に、最も短い経路を形成して行く。Gにたどり着いたら終了。

ネットワーク上のグラフ検索法

ダイクストラ法

各ノードの評価距離＝出発点からの経路





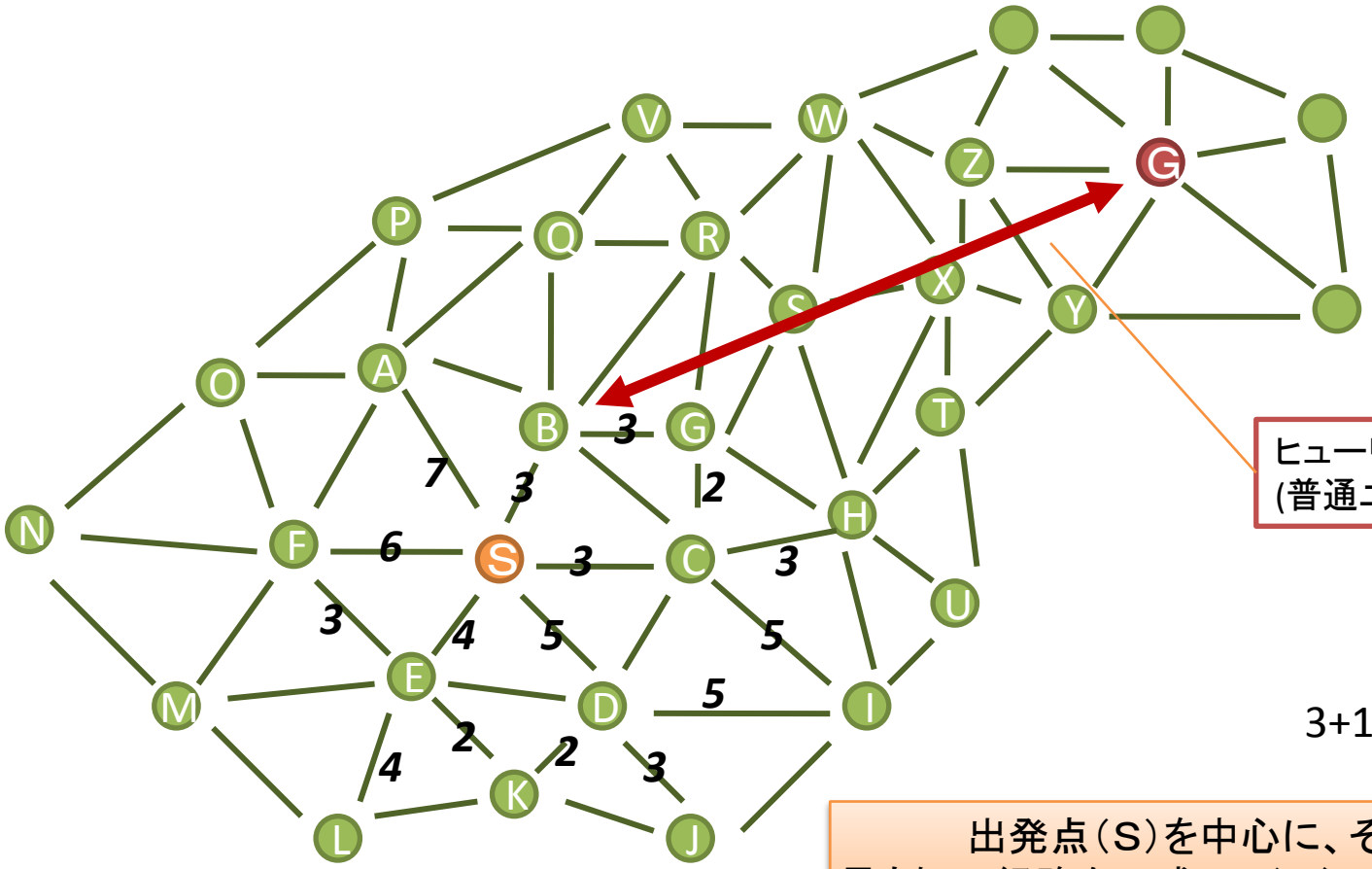
出発点（S）を中心に、最も短い経路を形成して行く。Gにたどり着いたら終了。

ネットワーク上のグラフ検索法

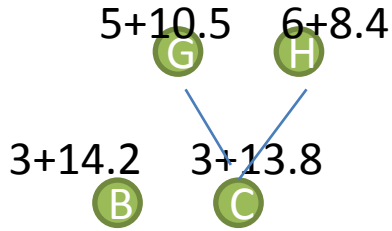
A*法

ゴール地点がわかっている場合、現在のノードとゴールとの推定距離（ヒューリスティック距離）を想定して、トータル距離を取り、それが最少のノードを探索して行く。

各ノードの評価距離＝出発点からの経路＋ヒューリスティック距離



ヒューリスティック距離
(普通ユークリッド距離を取る)



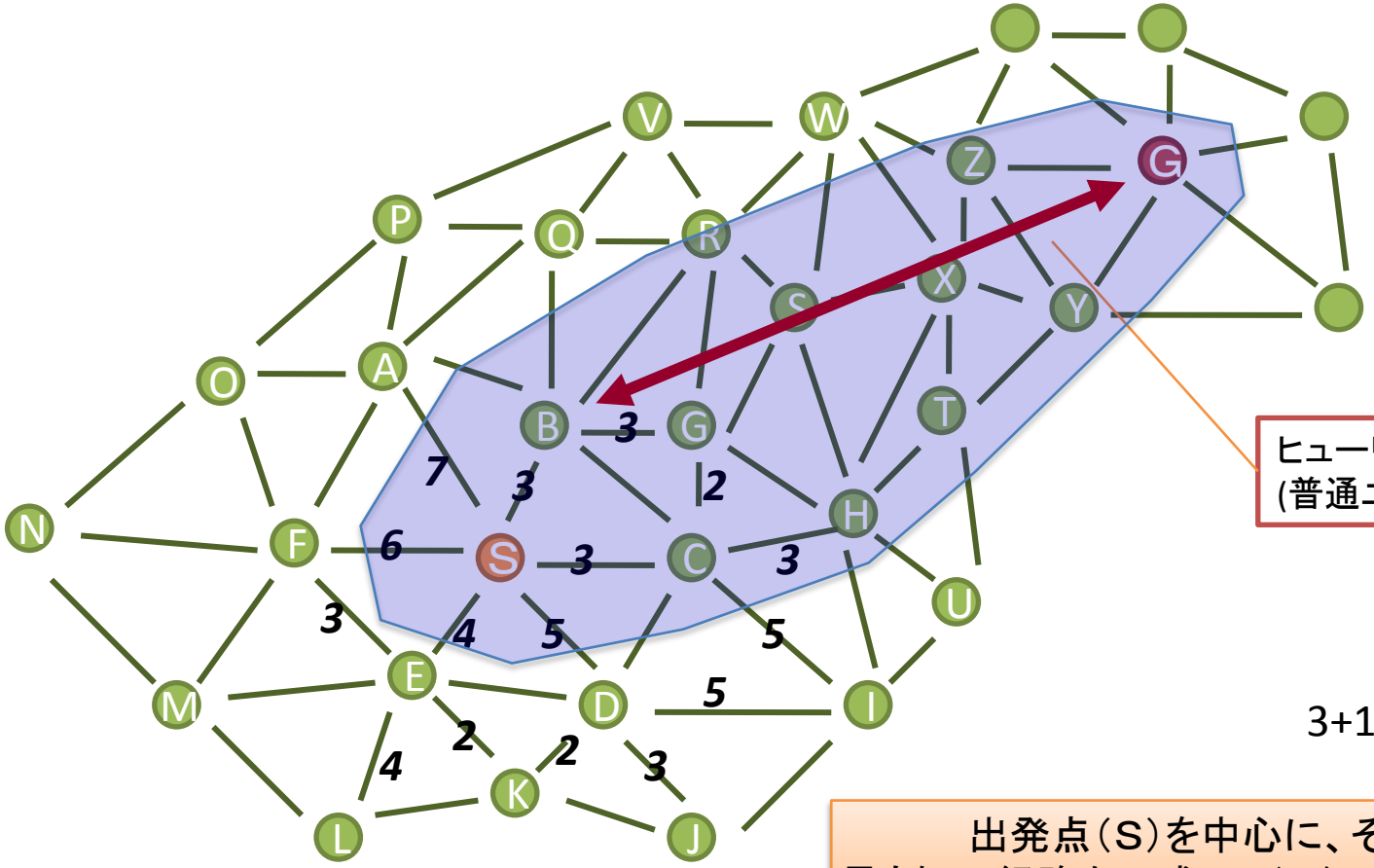
出発点（S）を中心に、そのノードまでの
最も短い経路を形成して行く。Gにたどり着いたら終了。

ネットワーク上のグラフ検索法

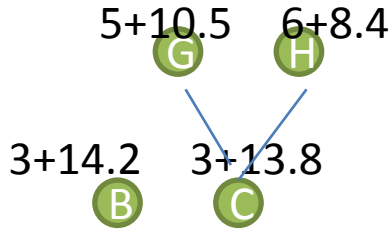
A*法

ゴール地点がわかっている場合、現在のノードとゴールとの推定距離（ヒューリスティック距離）を想定して、トータル距離を取り、それが最少のノードを探索して行く。

各ノードの評価距離＝出発点からの経路＋ヒューリスティック距離



ヒューリスティック距離
(普通ユークリッド距離を取る)



出発点（S）を中心に、そのノードまでの
最も短い経路を形成して行く。Gにたどり着いたら終了。

A* IN MARIO: CURRENT POSITION

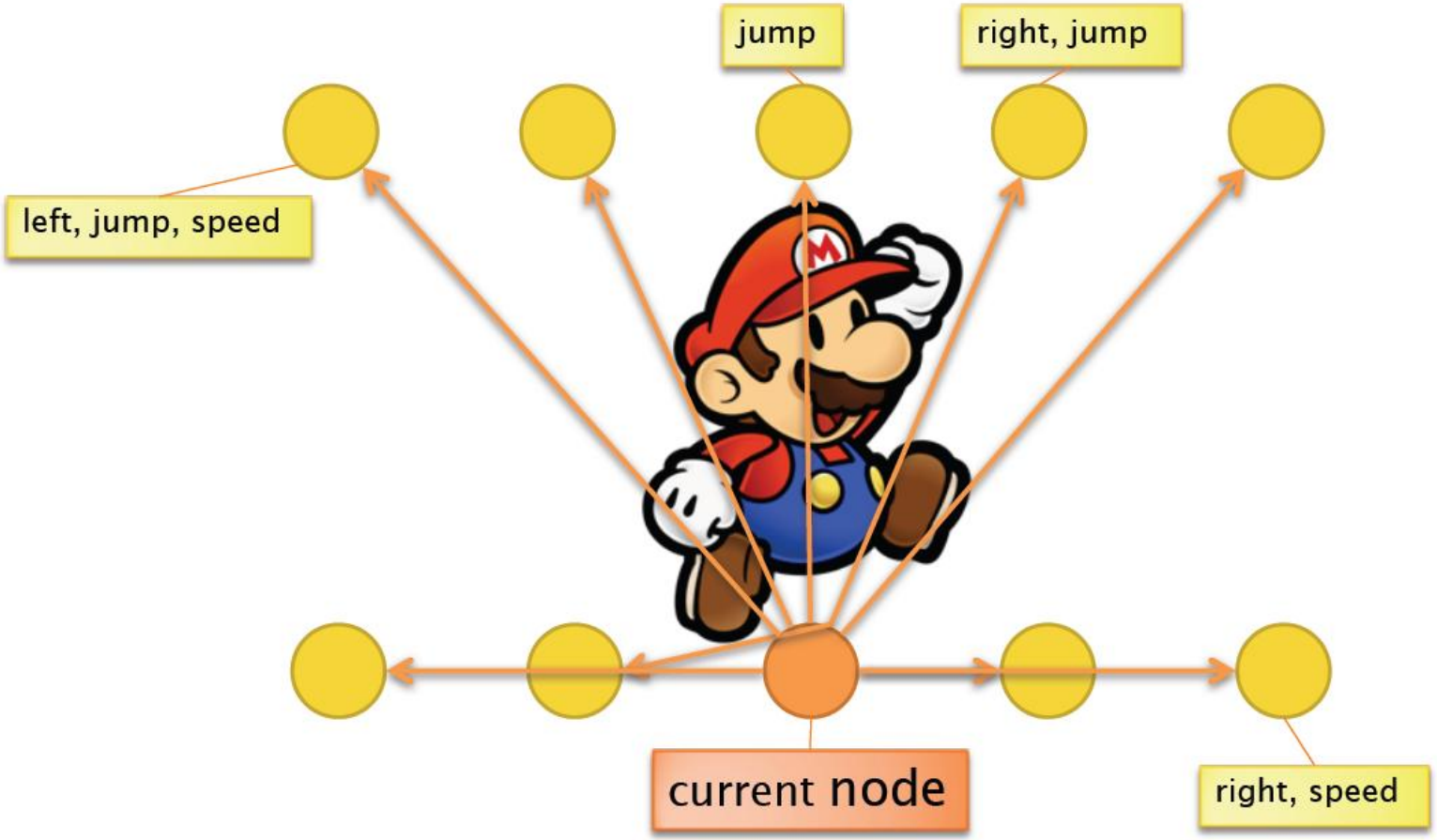
current node

Goal:
right border
of screen

ゴールは常に画面右端。

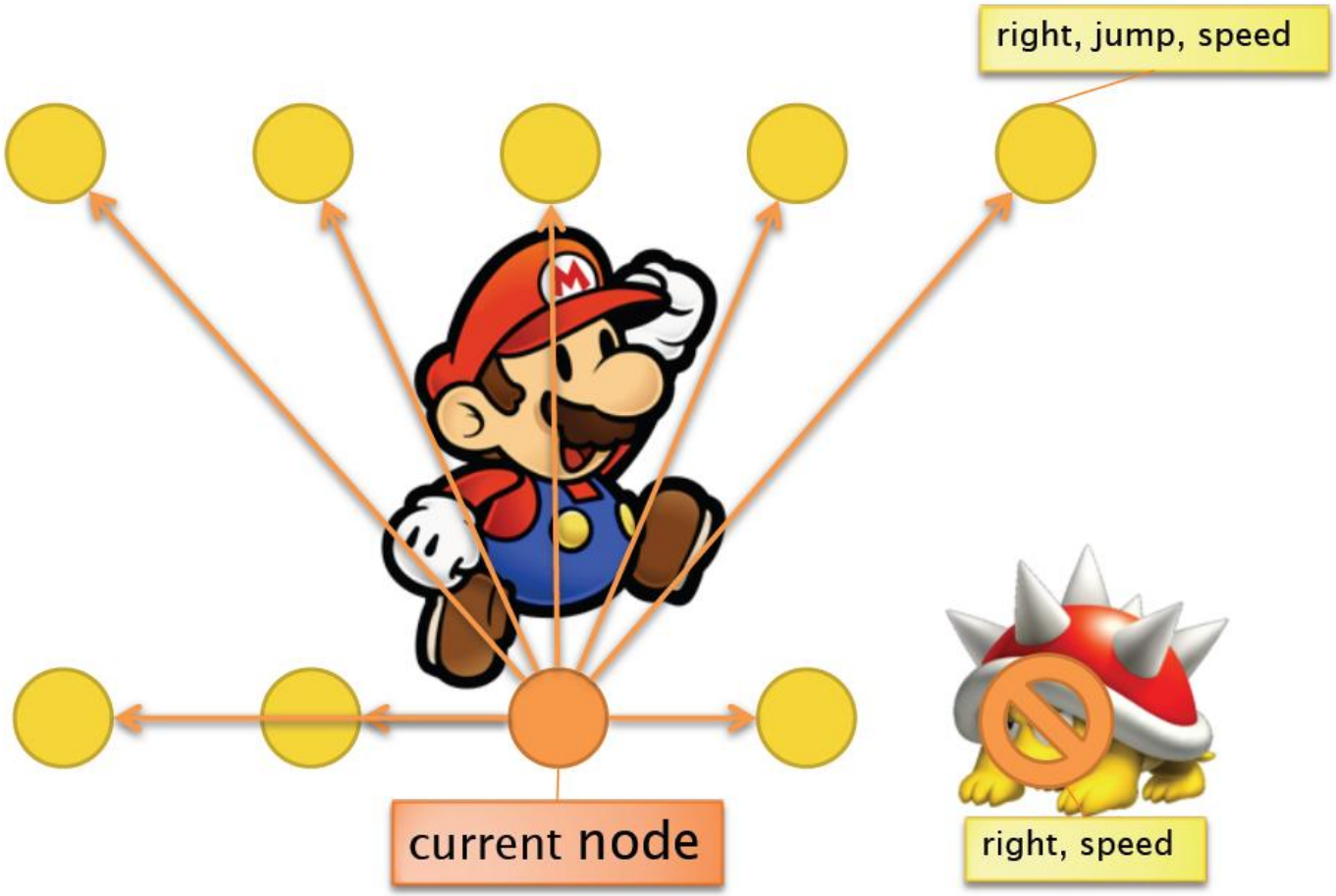
Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

A* IN MARIO: CHILD NODES



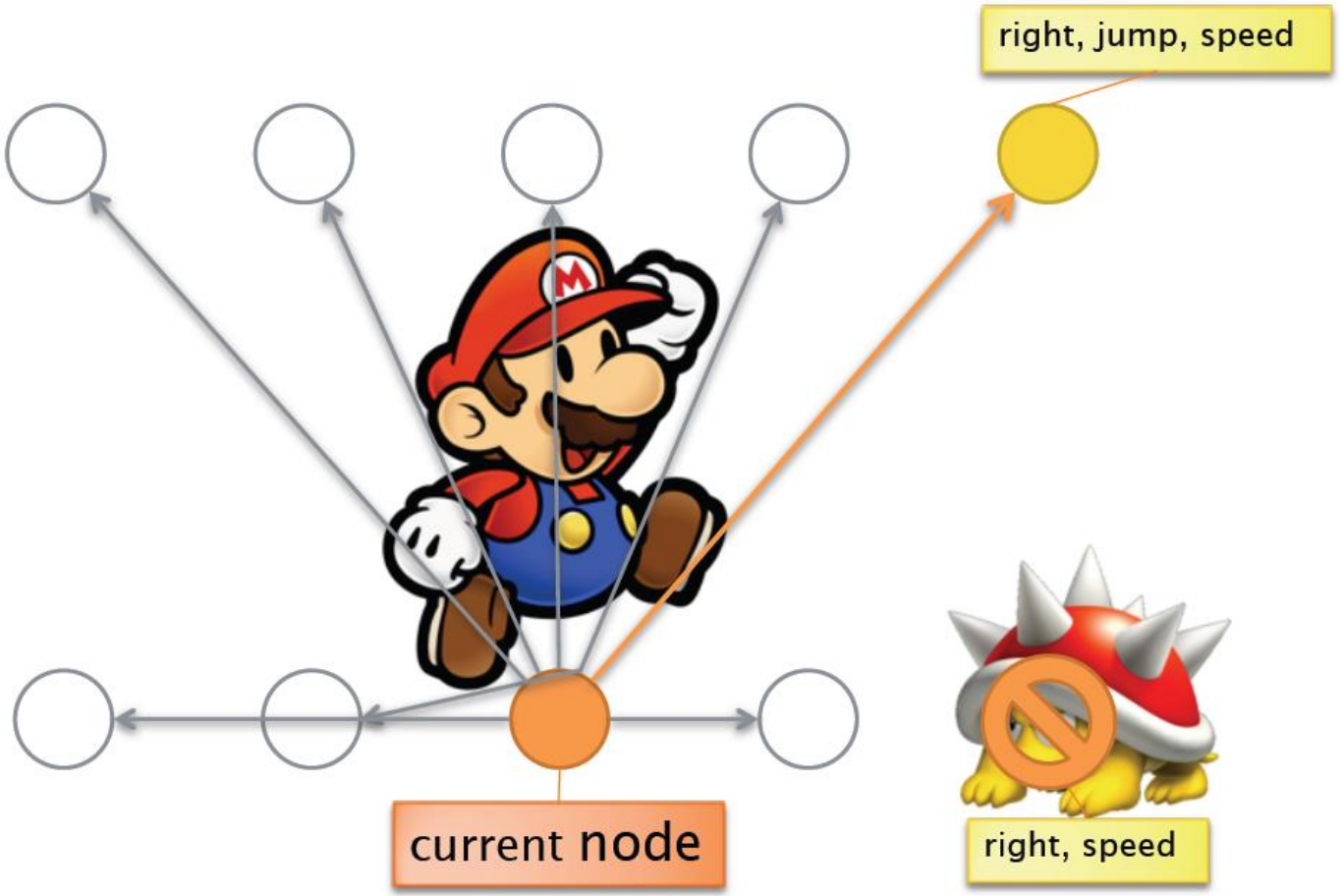
Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

A* IN MARIO: BACKTRACK



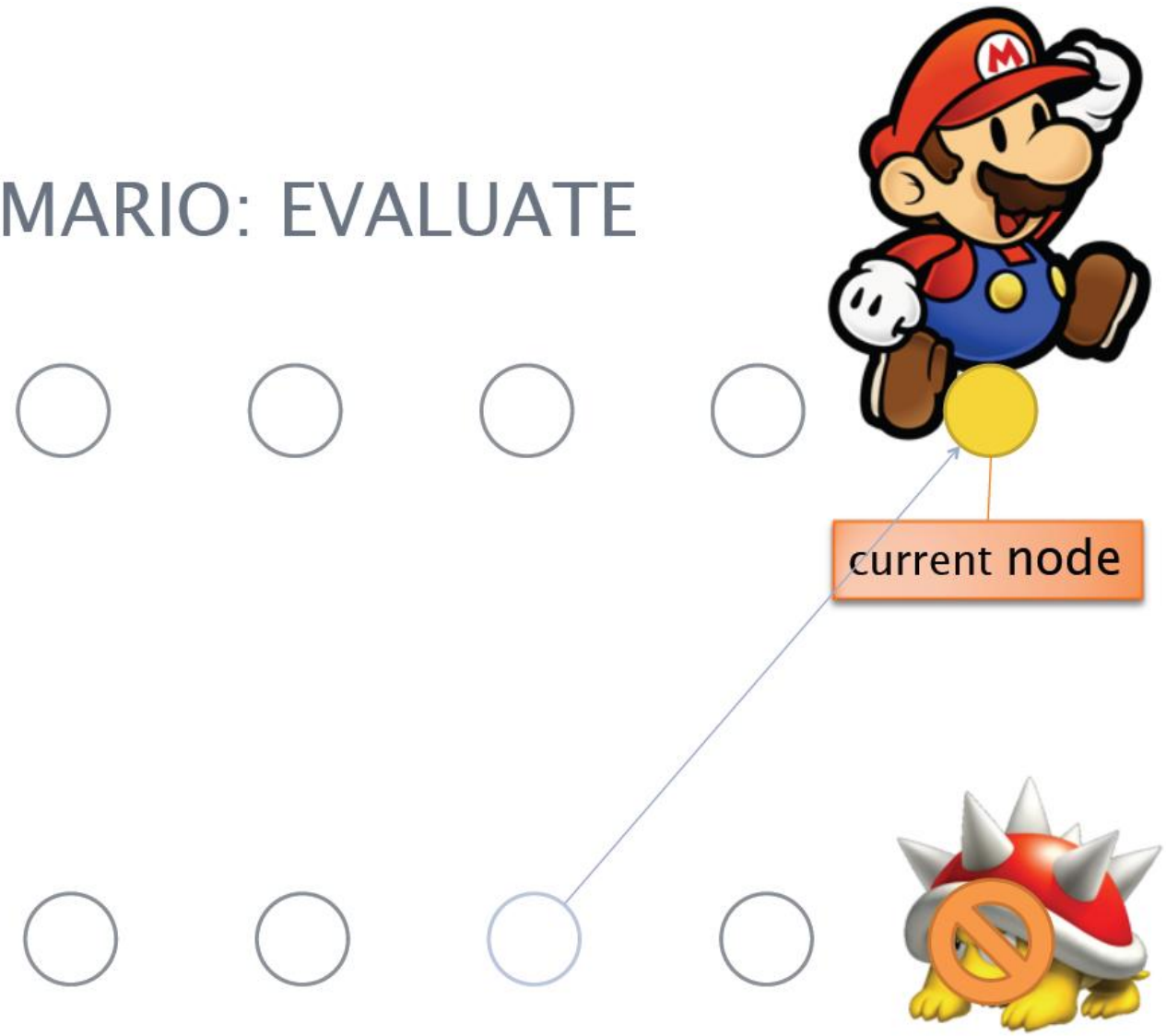
Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

A* IN MARIO: BEST FIRST



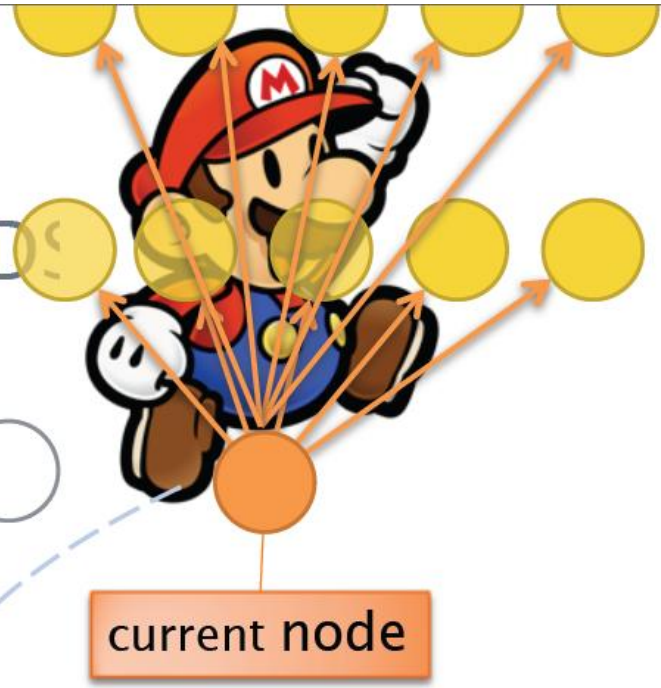
モーションプランニング

A* IN MARIO: EVALUATE



Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

A* IN MARIO: CREATE CHILDS



Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

A* IN MARIO: BEST FIRST

Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

HEURISTIC

- Using Mario's current speed and acceleration, how long does it take to reach the goal?
- Assume maximum acceleration and no obstacles (admissible heuristic!)

$$xa = xa + 1.2$$

$$x = x + xa$$

$$xa = xa * 0.89$$

- Optimisation: Find a closed form for this.

Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

ヒューリスティック関数

- ゴールまで辿り着くまでの時間は速度と加速度から計算する。
- 障害物がない場合、最大加速は以下の式。

$$xa = xa + 1.2$$

$$x = x + xa$$

$$xa = xa * 0.89$$

Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

ステージで起こるイベントとの同期

- 2フレームごとに更新
- オブジェクトと敵の位置をノードに反映

Mario AI Competition@ ICE-GIC 2009
Sergey Karakovskiy and Julian Togelius
http://julian.togelius.com/mariocompetition2009/GIC2009Competition.pdf

まとめ

(1) ステージの変化を予測する。

(2) マリオ自身の動きをシミュレートできる。

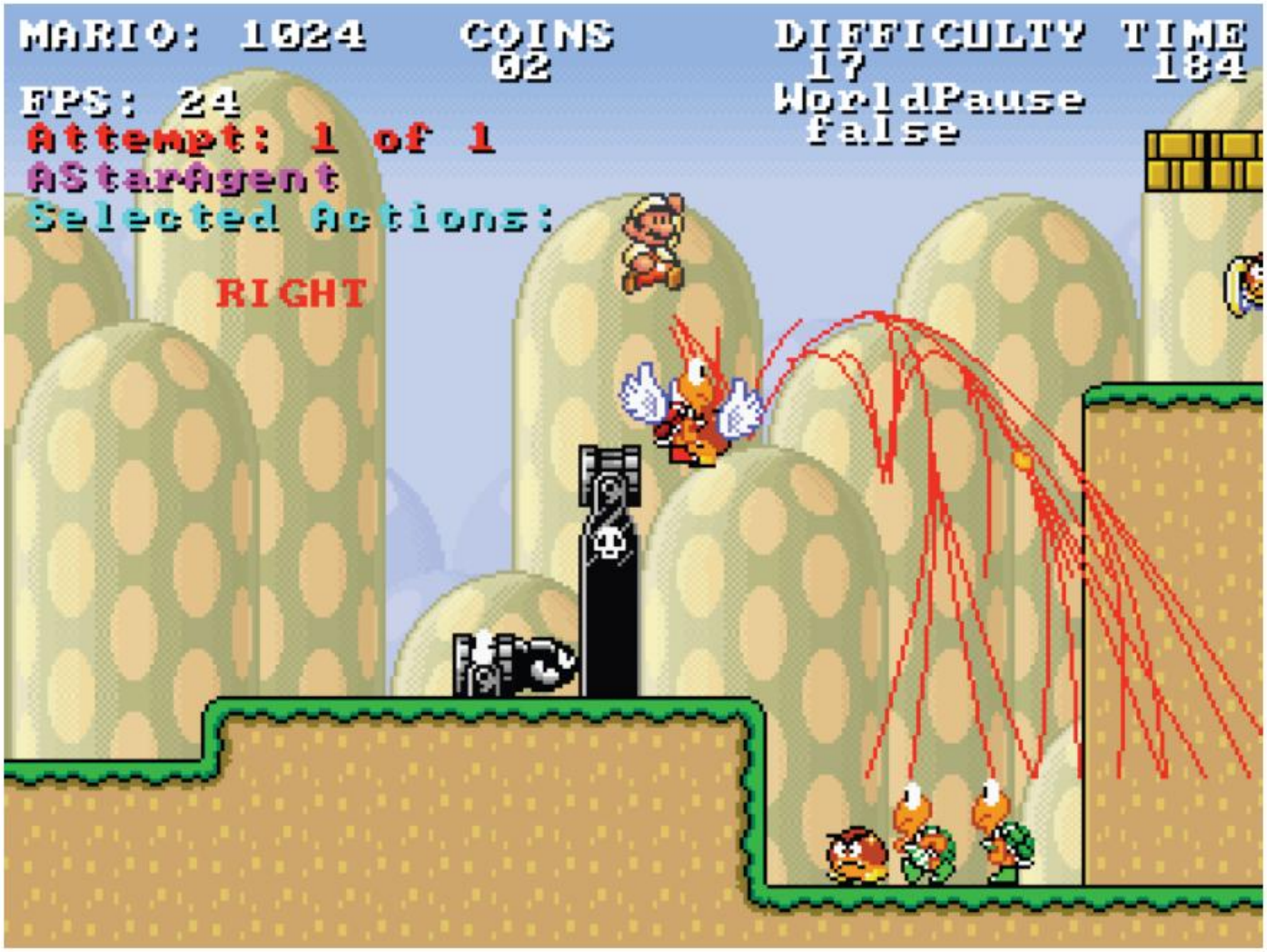
(3) A*検索を使う。（目標は右端）

(4) 経路だけでなくマリオの動きと共にアクションプランを作る（モーションプランニング）

(5) 敵の重みをA*検索のノードに加える。

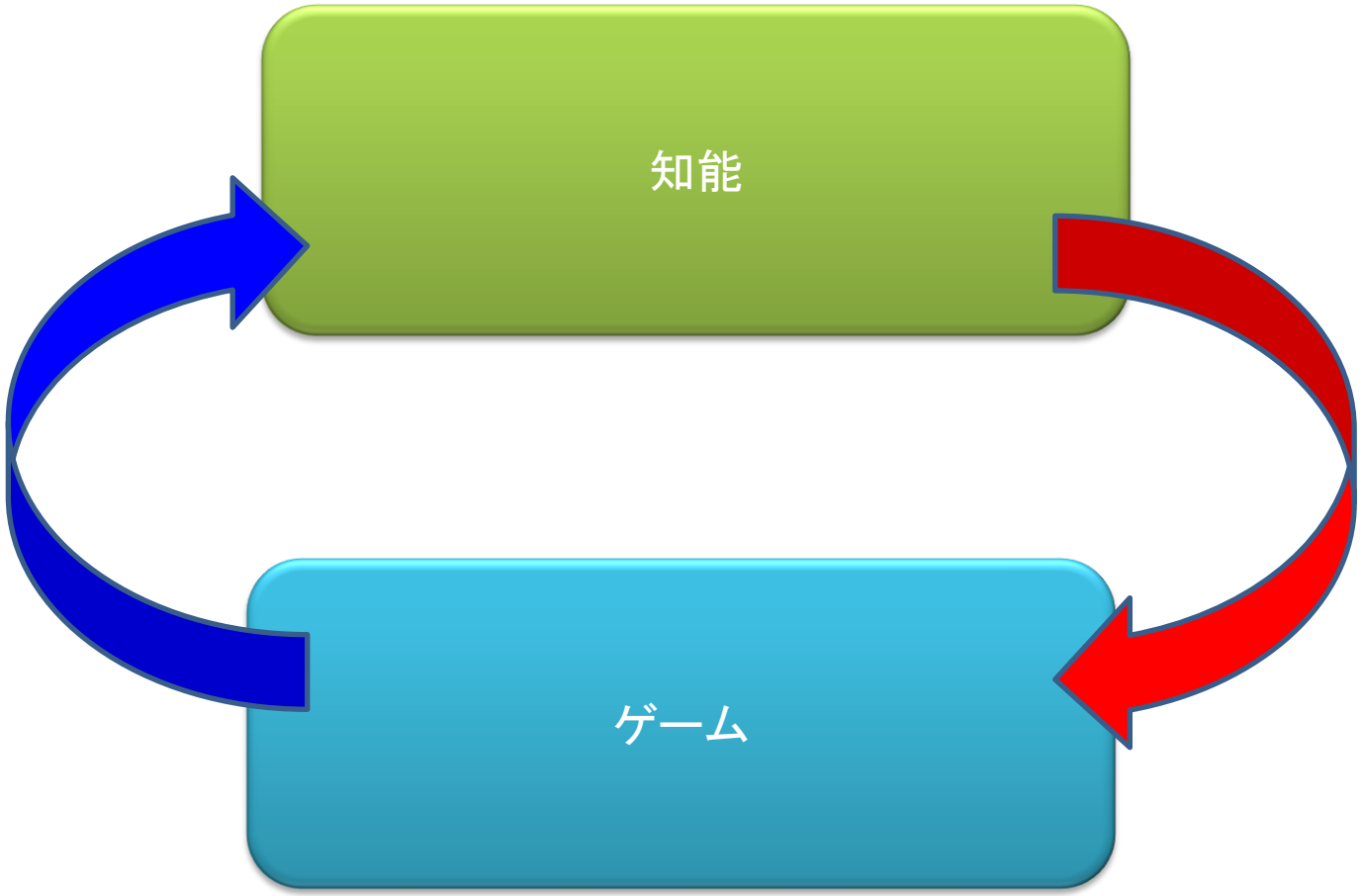
(6) 2フレームごとにパス検索をリプラニングする。

VIDEO



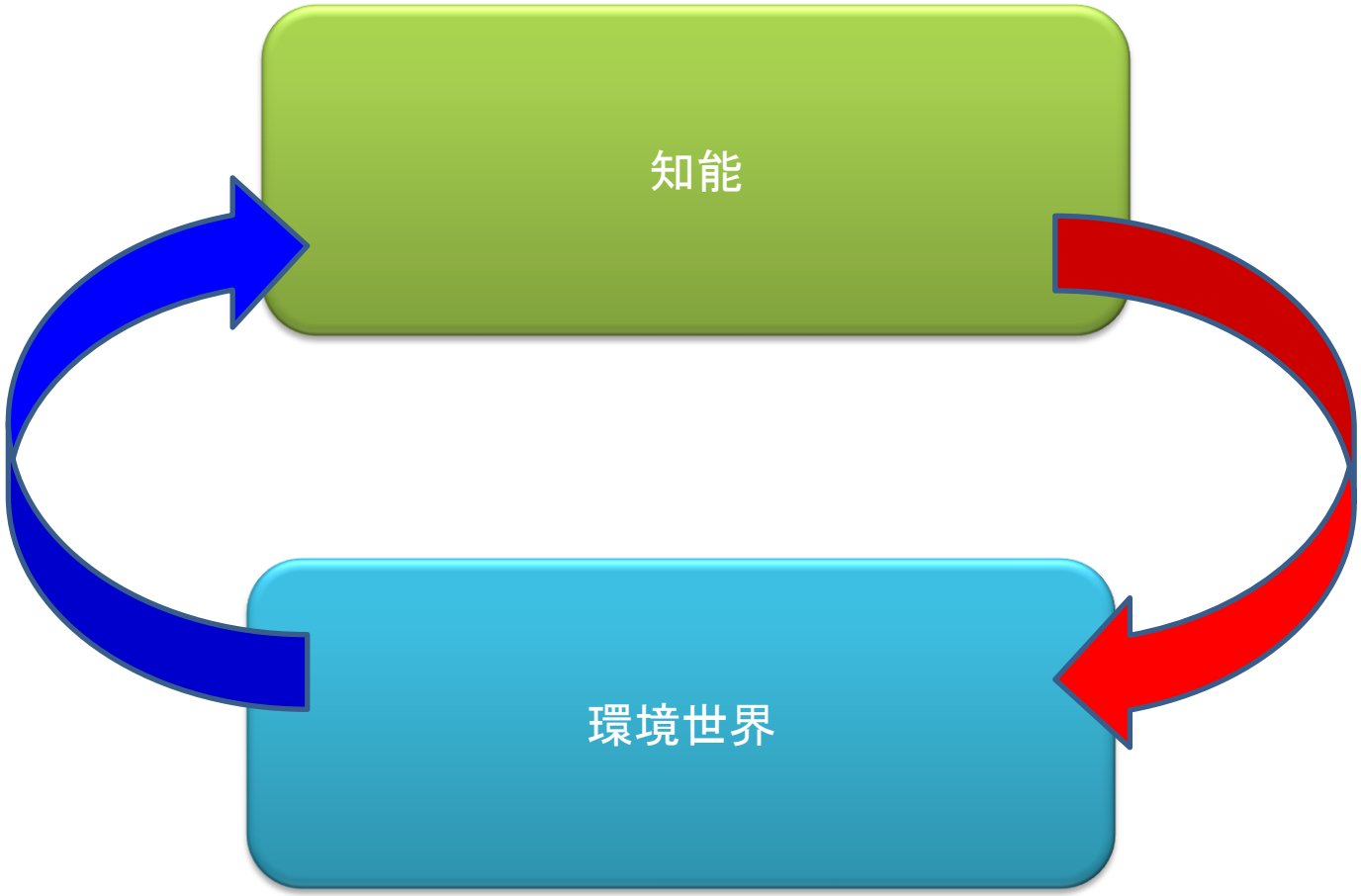
http://www.youtube.com/watch?v=DlkMs4ZHHr8

人工知能とゲームの新しい関係



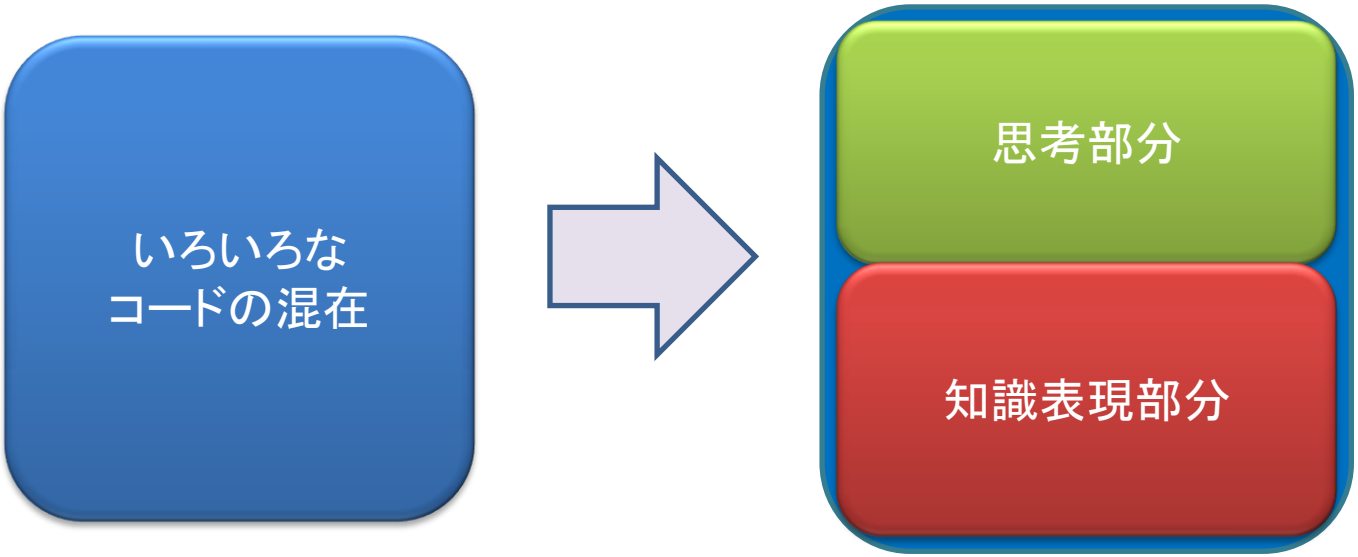
ゲームと知能を明確に分けましょう。人間がゲームを理解してAIを記述するのではなく、AIにゲームを理解させて問題を解かせましょう。

人工知能と世界の新しい関係

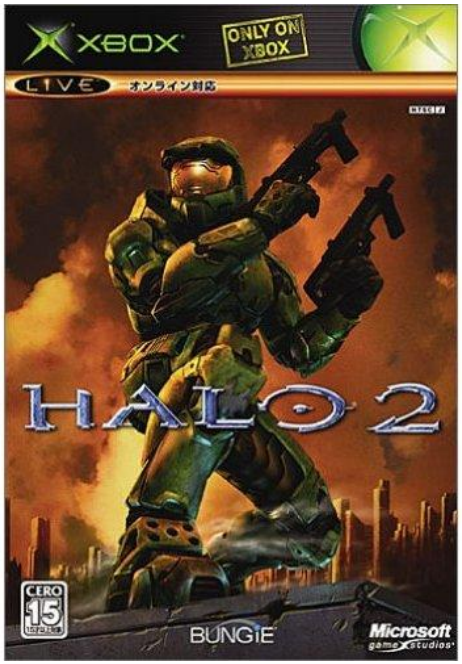


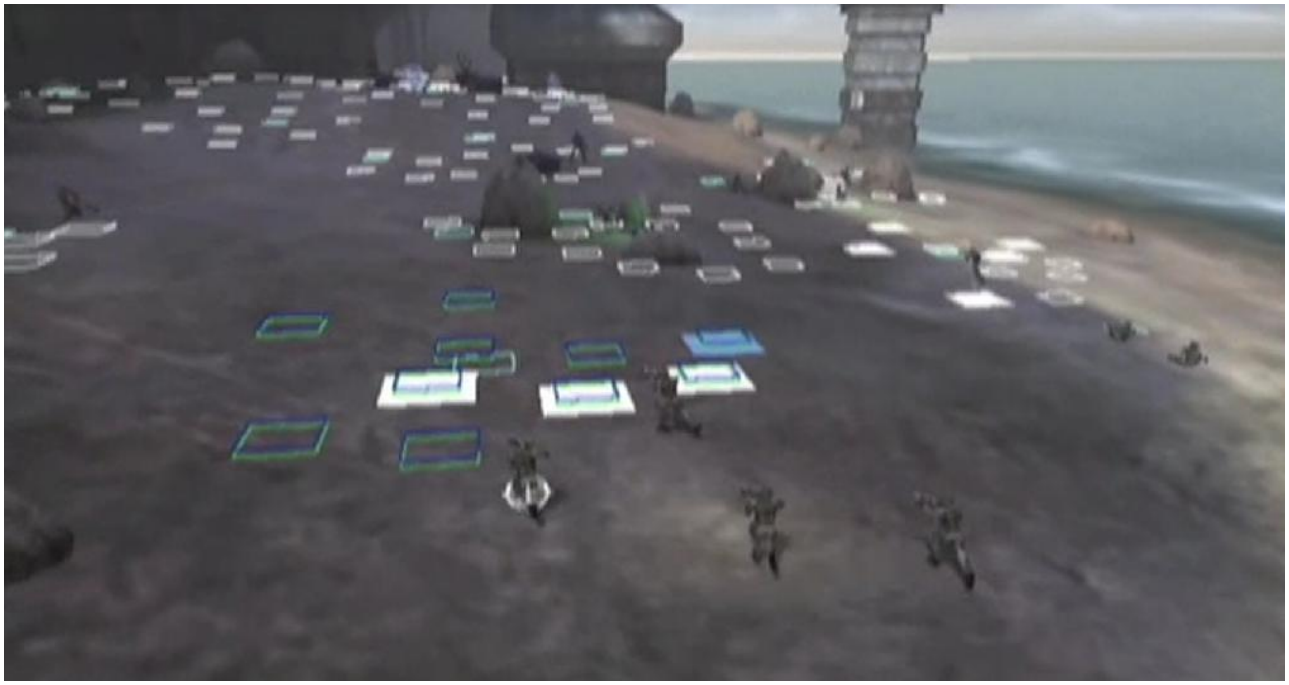
環境と知能を明確に分けましょう。人間が環境を理解してAIを記述するのではなく、AIに環境を理解させて問題を解かせましょう。

知識表現



デモ（Halo2）





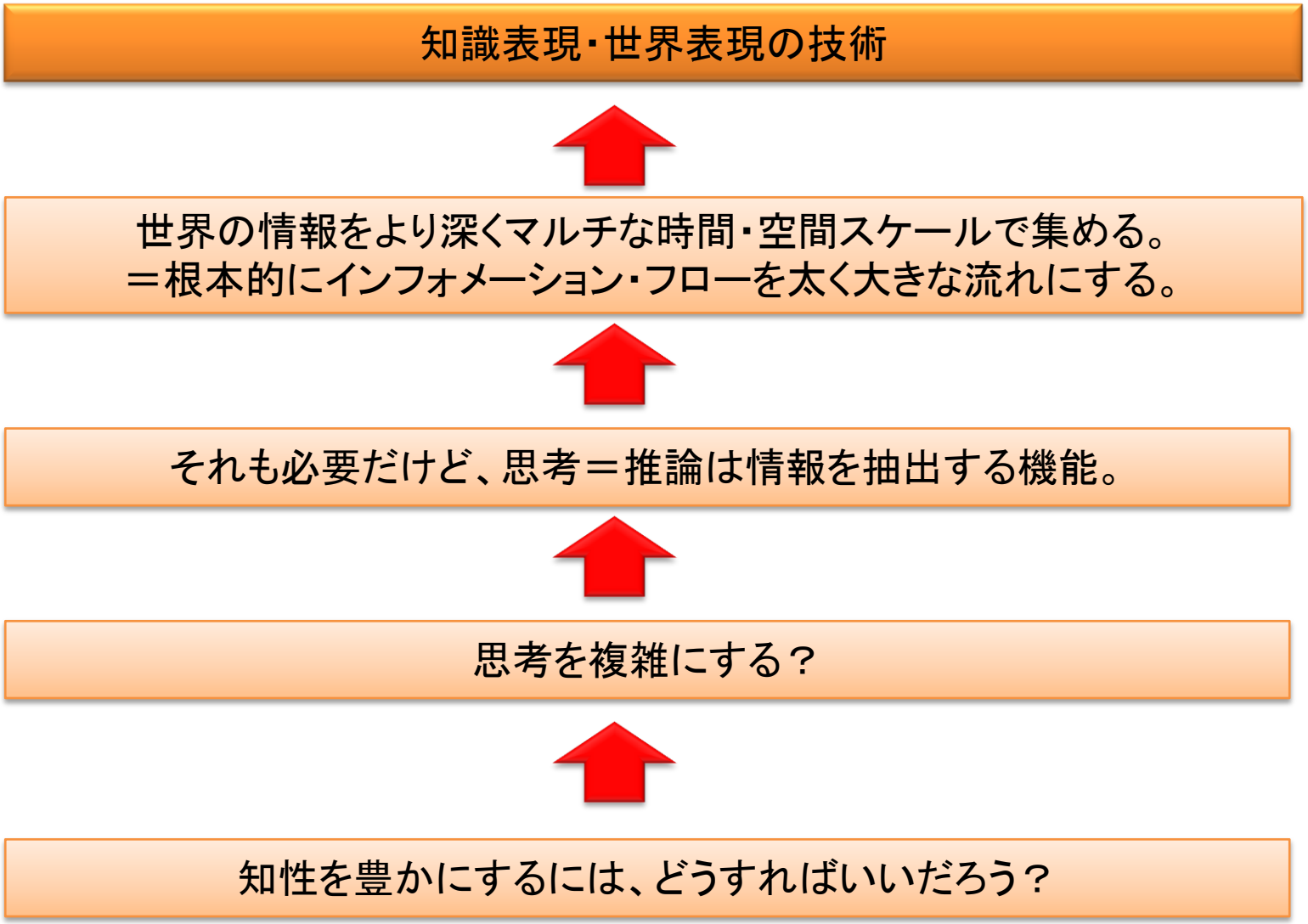
- Genre：SciFi-FPS
- Developer：　BUNGIE Studio
- Publisher　：Microsoft
- Hardware: Xbox, Windows, Mac
- Year：　2004

Halo AI Retrospective: 8 Years of Work on 30 Seconds of Fun
Author: Damian Isla (AI Engineering Lead)
http://www.bungie.net/Inside/publications.aspx

Demo

Halo AI Retrospective: 8 Years of Work on 30 Seconds of Fun
Author: Damian Isla (AI Engineering Lead)
http://www.bungie.net/Inside/publications.aspx

知性を豊かにするには？


知識表現・世界表現の技術
世界の情報をより深くマルチな時間・空間スケールで集める。
＝根本的にインフォメーション・フローを太く大きな流れにする。
それも必要だけど、思考＝推論は情報を抽出する機能。
思考を複雑にする？
知性を豊かにするには、どうすればいいだろう？

知識表現・世界表現

仮想世界の知性
＝人工知能

WORLD

人工知能は生物のように世界をそのまま認識・解釈できるだろうか？

仮想世界の知性
＝人工知能

知識表現
（KR）

WORLD

AIが世界(物・事・空間など)を解釈できるように、世界をうまく情報表現する
＝ **知識表現 (KR、Knowledge Representation)**

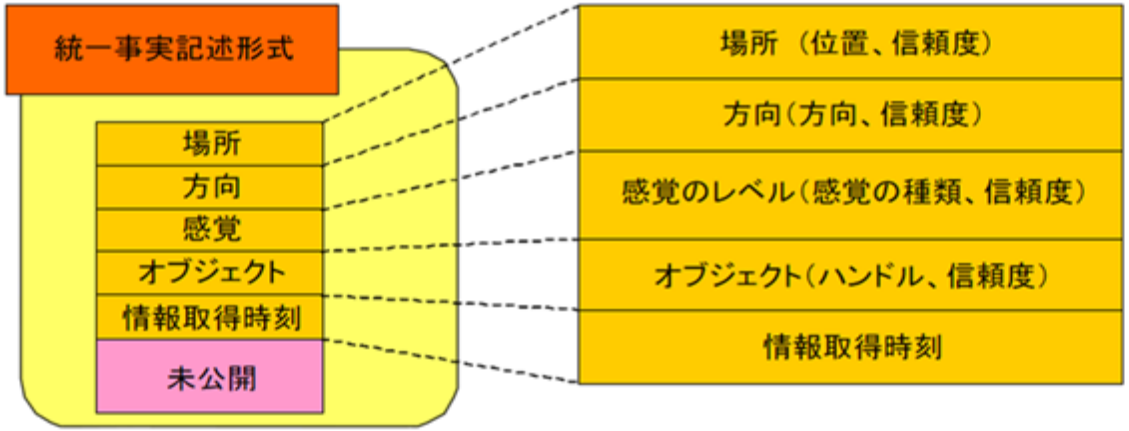
知識表現 (F.E.A.R.(2004年)における例)

統一事実表現（WMF, Working Memory Fact）

In F.E.A.R.

キャラクター | オブジェクト | 任務 | 事件 | パス | 欲求 | ノード

全て以下の形式（フォーマット）で記述する。



全部で本当は16個の属性がある

```
WorkingMemoryFact
{
   Attribute<Vector3D>      Position
   Attribute<Vector3D>      Direction
   Attribute<StimulusType>  Stimulus
   Attribute<Handle>        Object
   Attribute<float>         Desire
   ...
   float                fUpdateTime
}
```

```
Attribute<Type>
{
   Type  Value
   float fConfidence
}
```

値

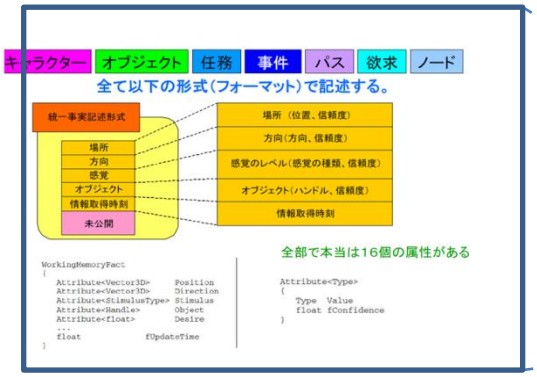
情報の信頼度

Jeff Orkin, "Three States and a Plan: The A.I. Of F.E.A.R(GDC2006) (Document,PPT,Movie)", 2006 http://www.jorkin.com/gdc2006_orkin_jeff_fear.zip

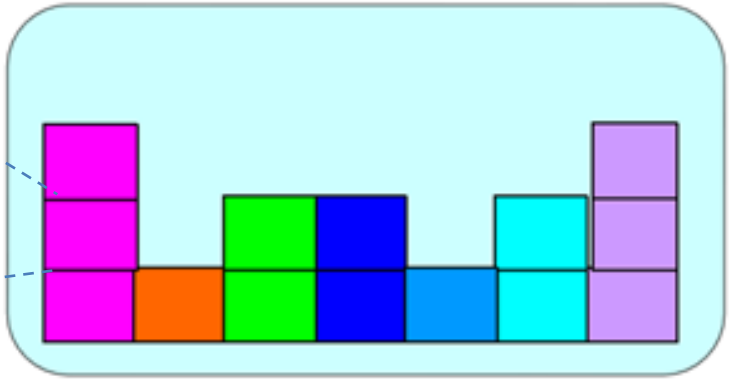
知識表現（F.E.A.R.(2004年)における例)

統一事実表現（WMF, Working Memory Fact）

In F.E.A.R.



記憶領域に認識した事実をWorking Memoryへ蓄積する



Working Memory

敵Cが時刻Mに位置Zにいた
信頼度= 0.6

敵Aが時刻Lに位置Xにいた
信頼度 = 1.0

敵Bが時刻Mに位置Yにいた
信頼度 = 0.0

ターゲットを決める

Blackboard
ターゲットはAにする

知識表現

知識表現の多様さは知性（AI）に柔軟性を与え、
その豊富さは高い知性を作る足場を与える。

すべてを表現して持っておく必要はない。
AIが必要とする知識表現を持っておけばおくほどよい。

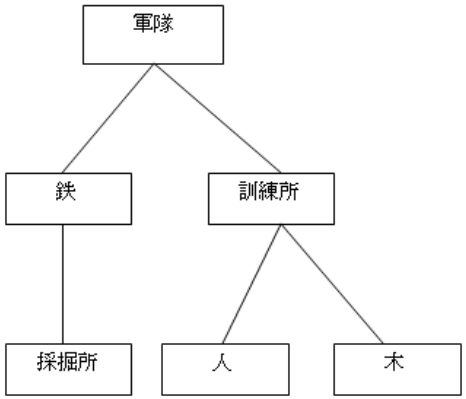
同じ物、同じマップに対しても複数の表現を持てば持つほどよい。

「物（オブジェクト）」の数ではない。AIが解釈するべき事実の数だけある。

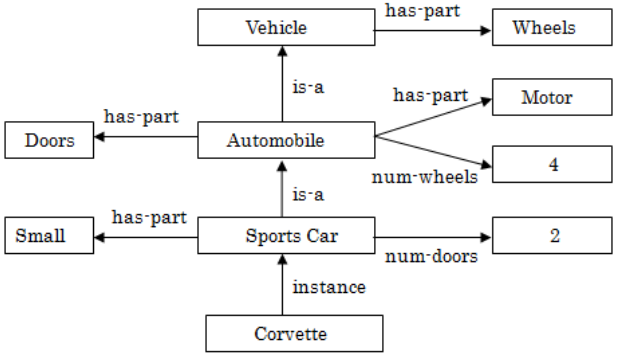
知識表現はいくつあるのだろう？

いろいろな知識表現

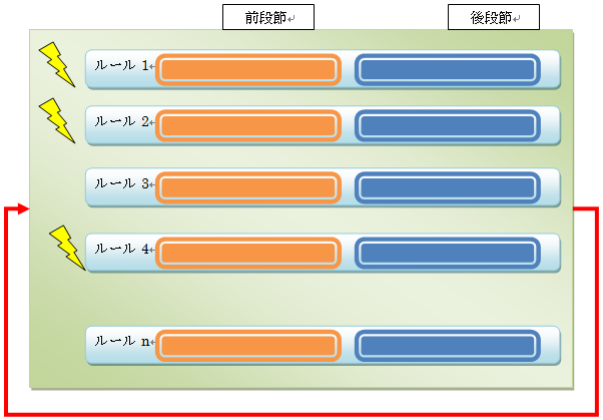
依存グラフ



意味ネットワーク

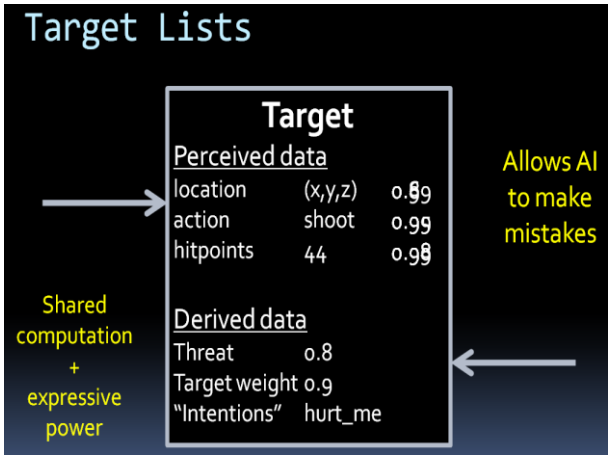


ルールベース表現

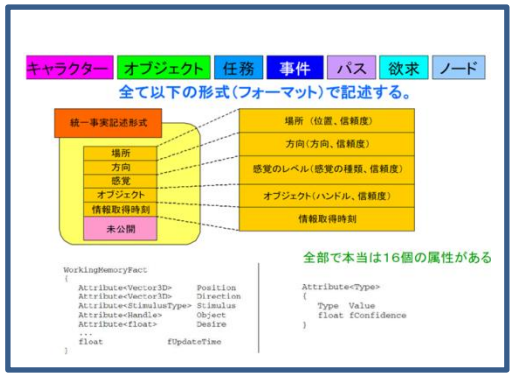


世界表現

敵表現リスト



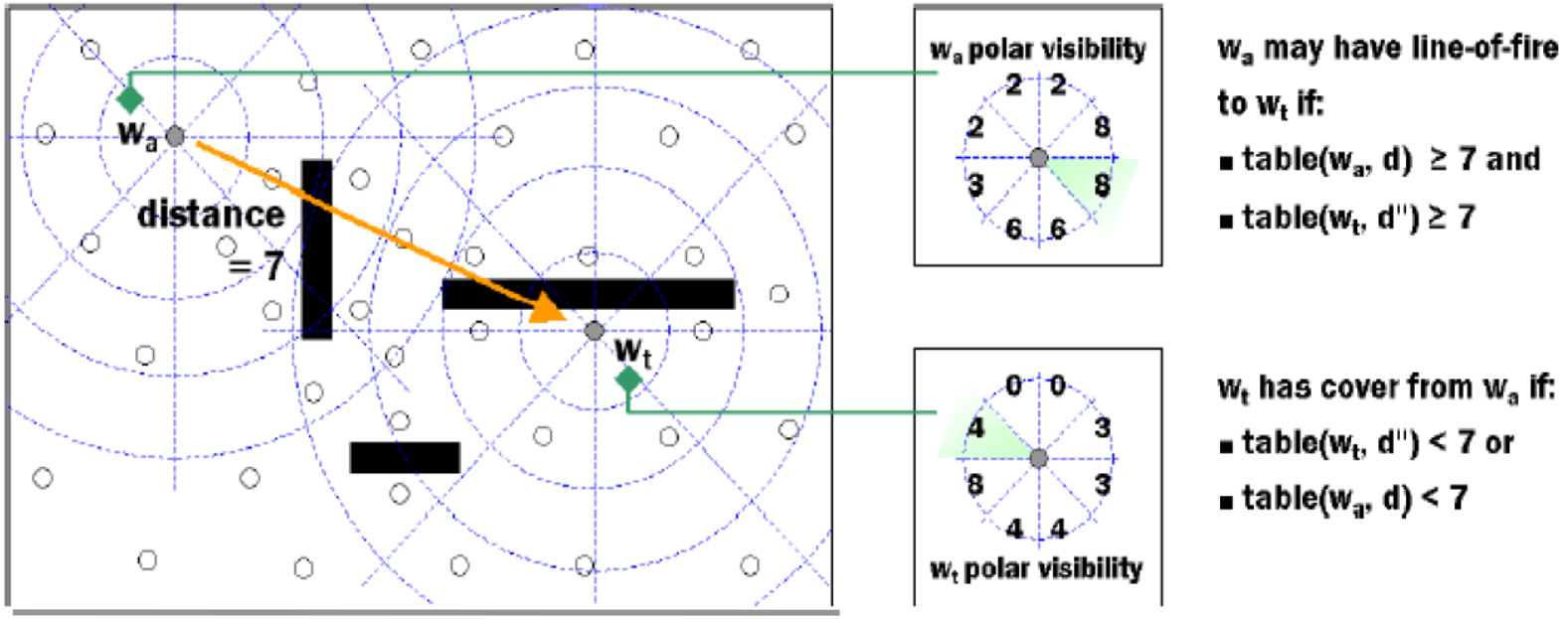
事実表現（信頼度表現）



Griesemer,J, "The Illusion of Intelligence: The Integration of AI and Level Design in Halo", 2002
http://www.bungie.net/images/Inside/publications/presentations/publicationsdes/design/gdc02_jaime_griesemer.pdf

世界表現（知識表現の一つ）

マップ全体に関わる知識表現
（ウェイポイント、ナビメッシュを基本とする表現）
World Representation（WR）
Location-based Information System ともいう。

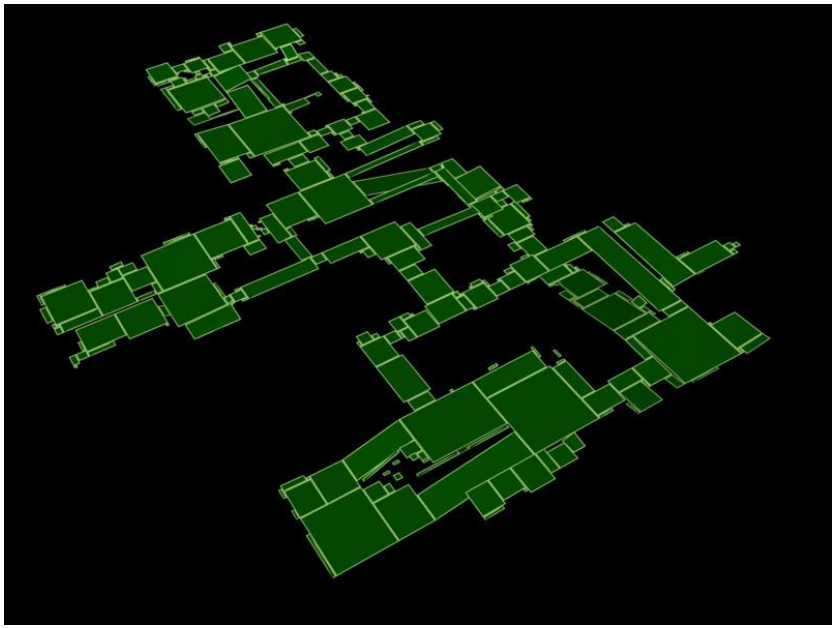


（例） 8方向の可視距離の各ポイントに埋め込まれたウェイポイント群 (Killzone)

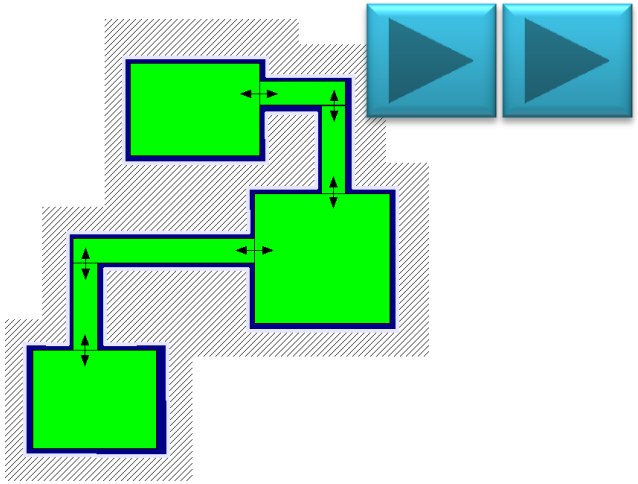
Straatman, R., Beij, A., Sterren, W.V.D., "Killzone's AI : Dynamic Procedural Combat Tactics", 2005
http://www.cgf-ai.com/docs/straatman_remco_killzone_ai.pdf

Character Navigation

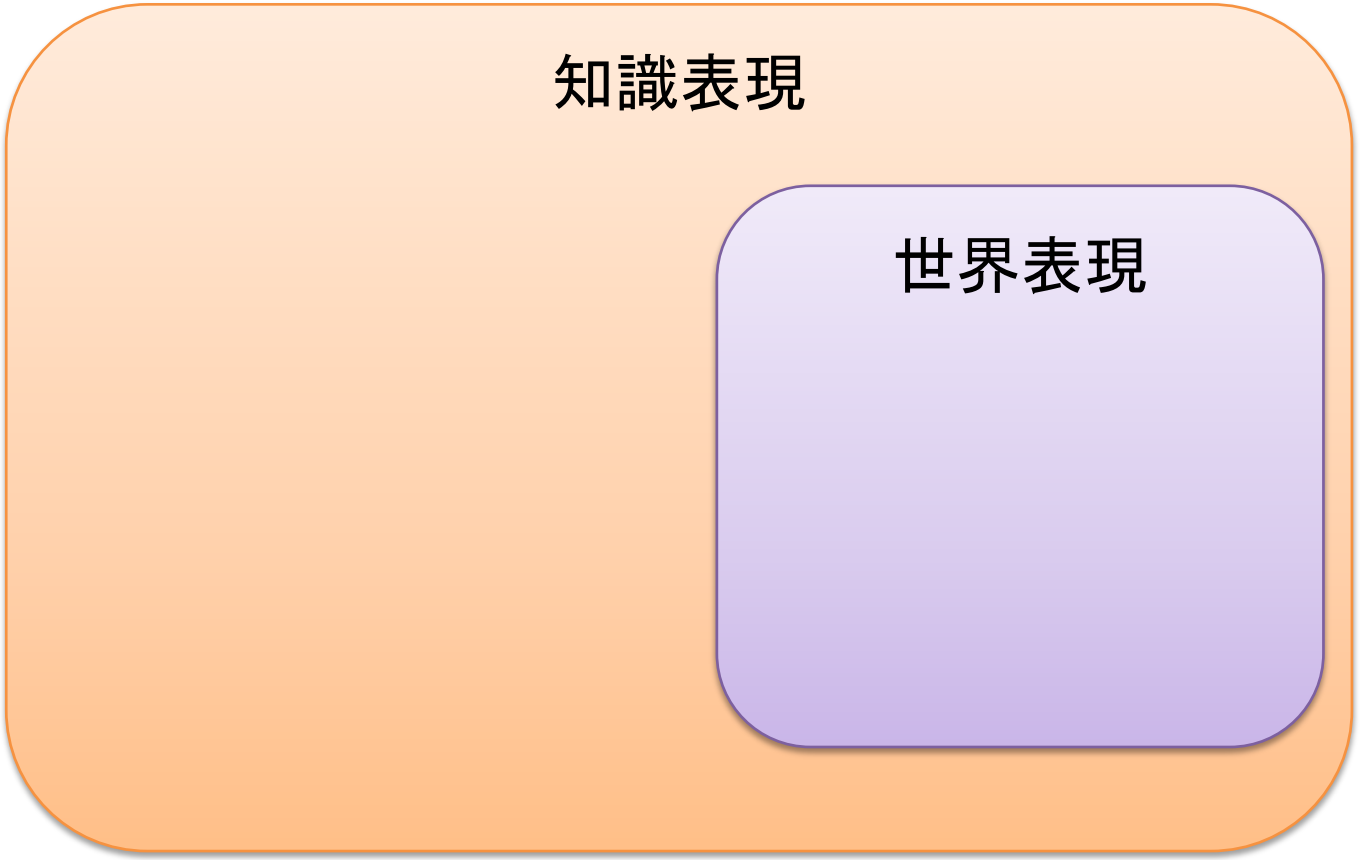
Moving Character in the Environment



- Genre : FPS
- Developer : TURTLEROCK STUDIO
- Publisher :
- Hardware: PC
- Year :



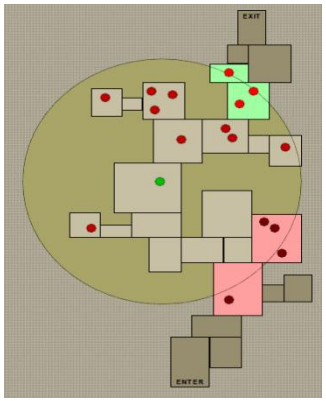
知識表現・世界表現



いろいろな世界表現

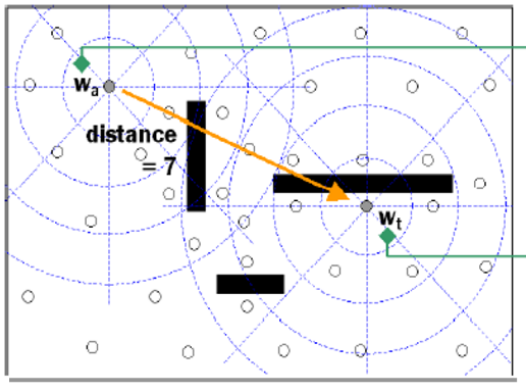
Damian Isla,"Building a Better Battle: HALO 3 AI Objectives", http://www.bungie.net/inside/publications.aspx

敵配位マップ



Left 4 Dead

LOS マップ



Killzone

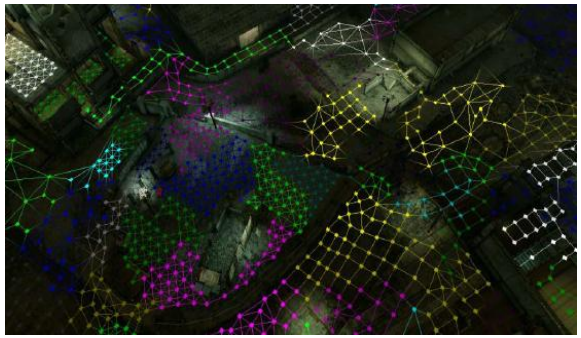
テリトリー表現

Halo2

Tactical Point System

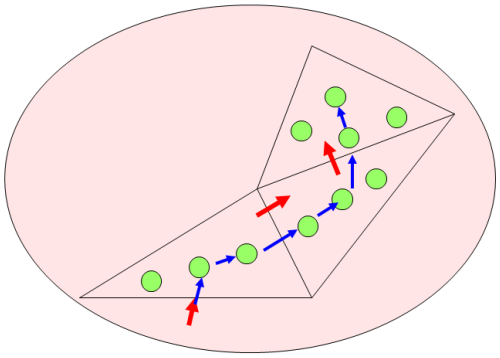
Halo

戦術マップクラスタリング



Killzone2

ナビメッシュ-ウェイポイント階層表現



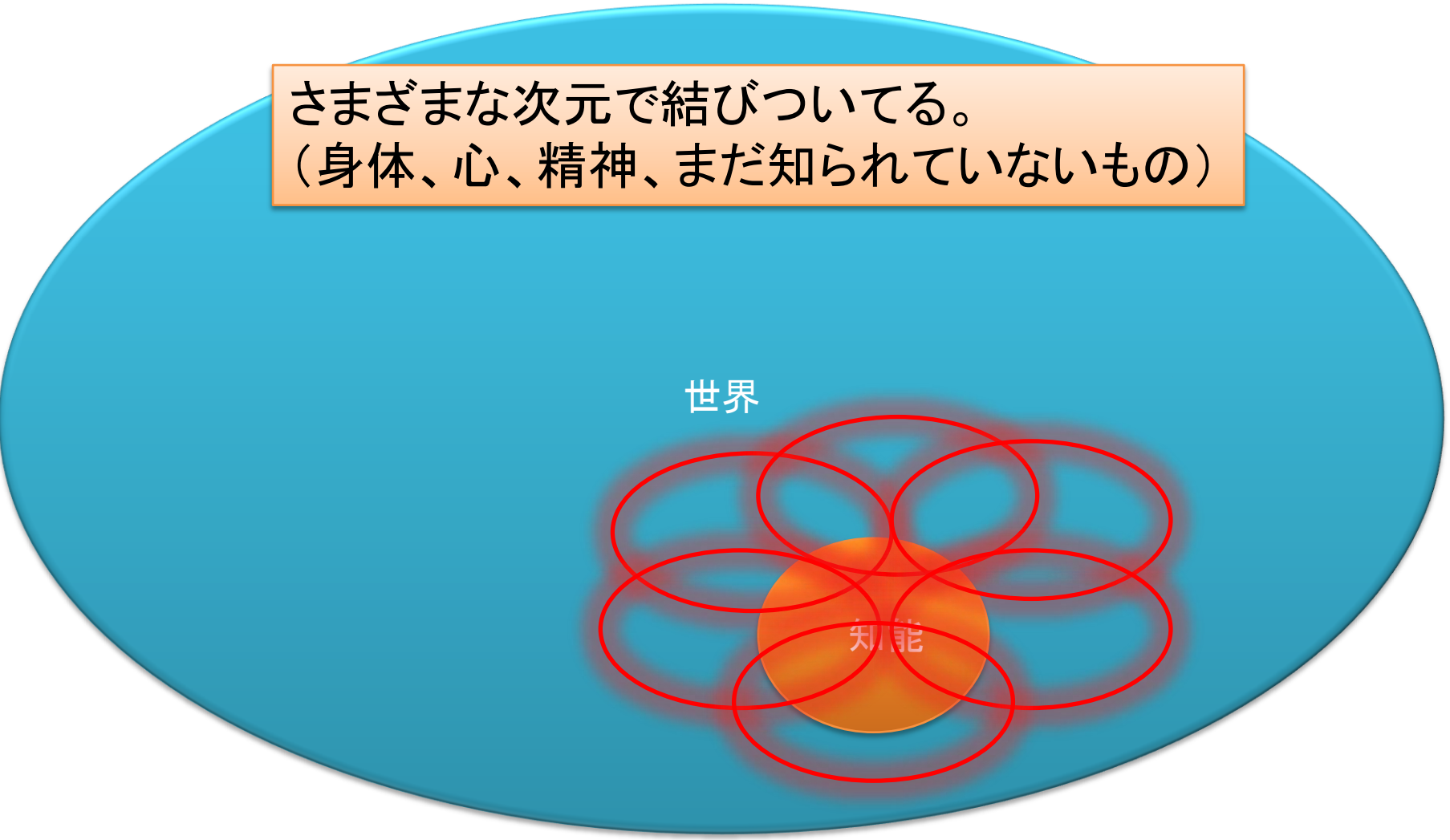
Assassin's Creed

Michael Booth, "The AI Systems of Left 4 Dead," Artificial Intelligence and Interactive Digital Entertainment Conference , http://www.valvesoftware.com/publications.html

Alex J. Champandard, Remco Straatman, Tim Verweij, "On the AI Strategy for KILLZONE 2's Bots" http://aigamedev.com/open/coverage/killzone2/

テーマ：人工知能はどのように世界と結びつくか？

さまざまな次元で結びついてる。
（身体、心、精神、まだ知られていないもの）



テーマ：人工知能はどのように世界と結びつくか？

さまざまな次元で結びついてる。
（身体、心、精神、まだ知られていないもの）

知能が世界を理解することで結びついている。

まとめ

(1) AIが世界を分離する。
(2) AIが世界を理解する。
(3) 知識表現、世界表現、アフォーダンス。

今日の内容

第一章

AIはどのように世界を結びつくか？

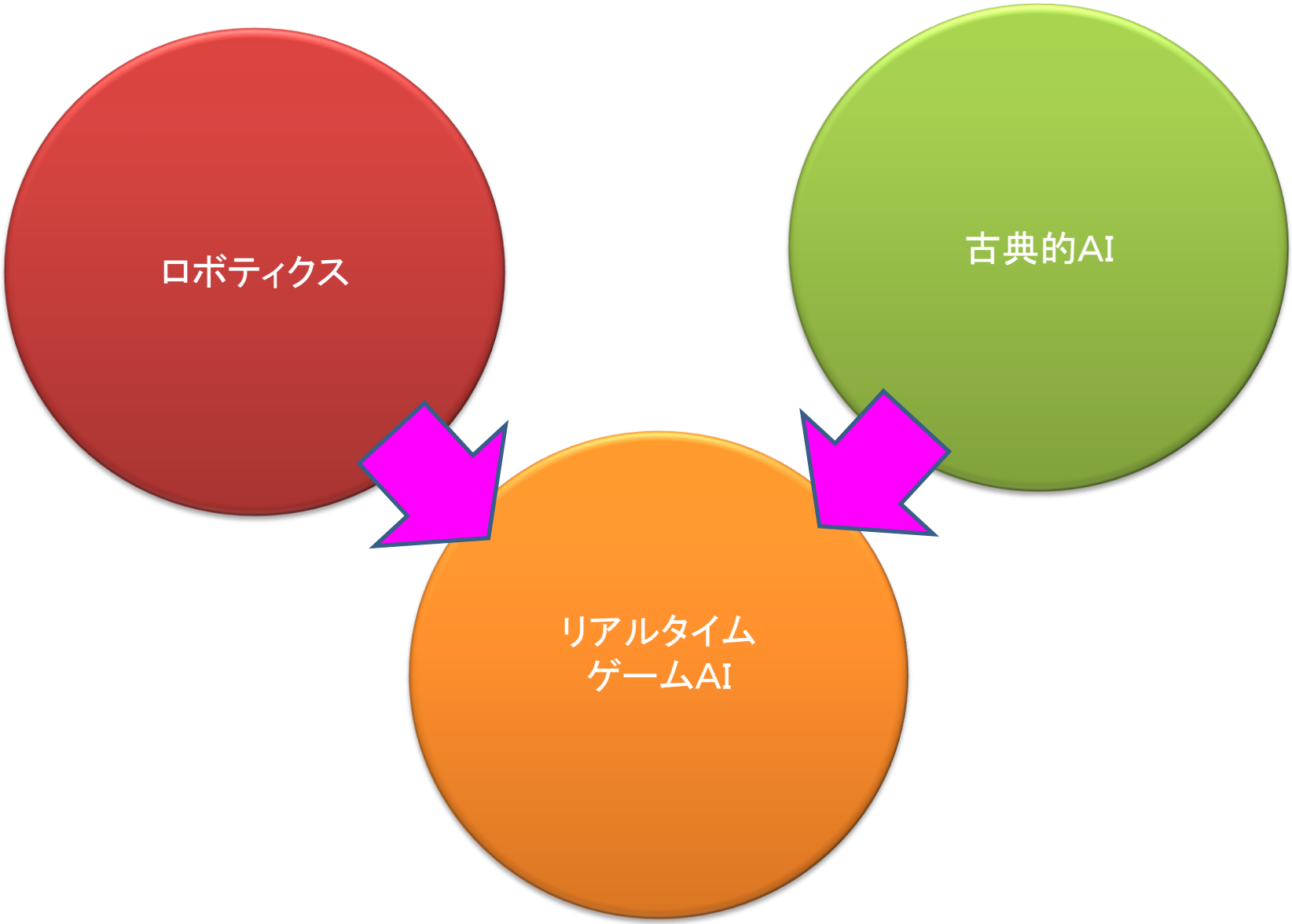
第二章

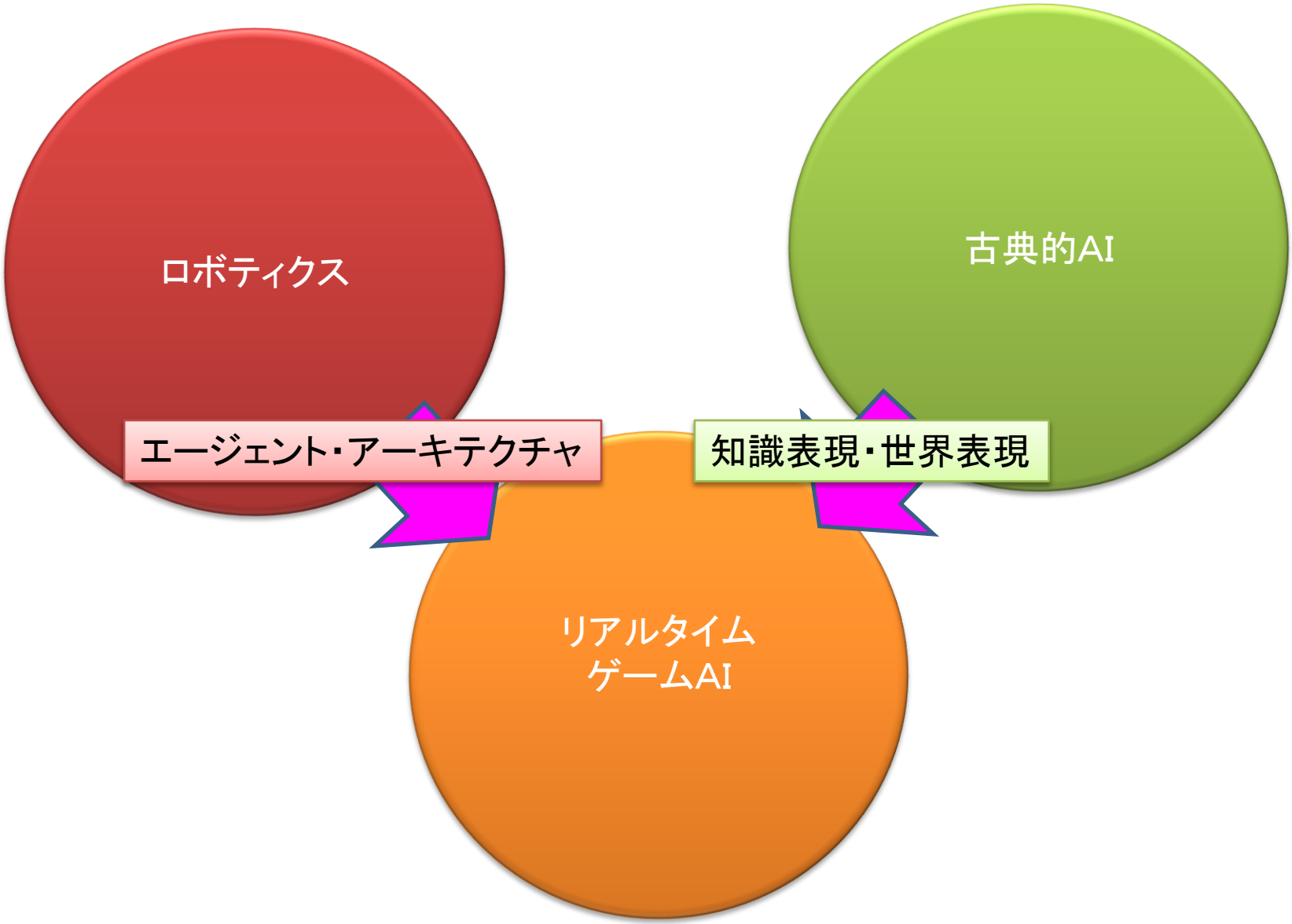
エージェントモデル

第三章

行動表現と世界表現の双対性

第2章
エージェントモデル


ロボティクス
古典的AI
リアルタイム
ゲームAI


ロボティクス
古典的AI
エージェント・アーキテクチャ
知識表現・世界表現
リアルタイム
ゲームAI

ロボティクスとゲームAIの共通点

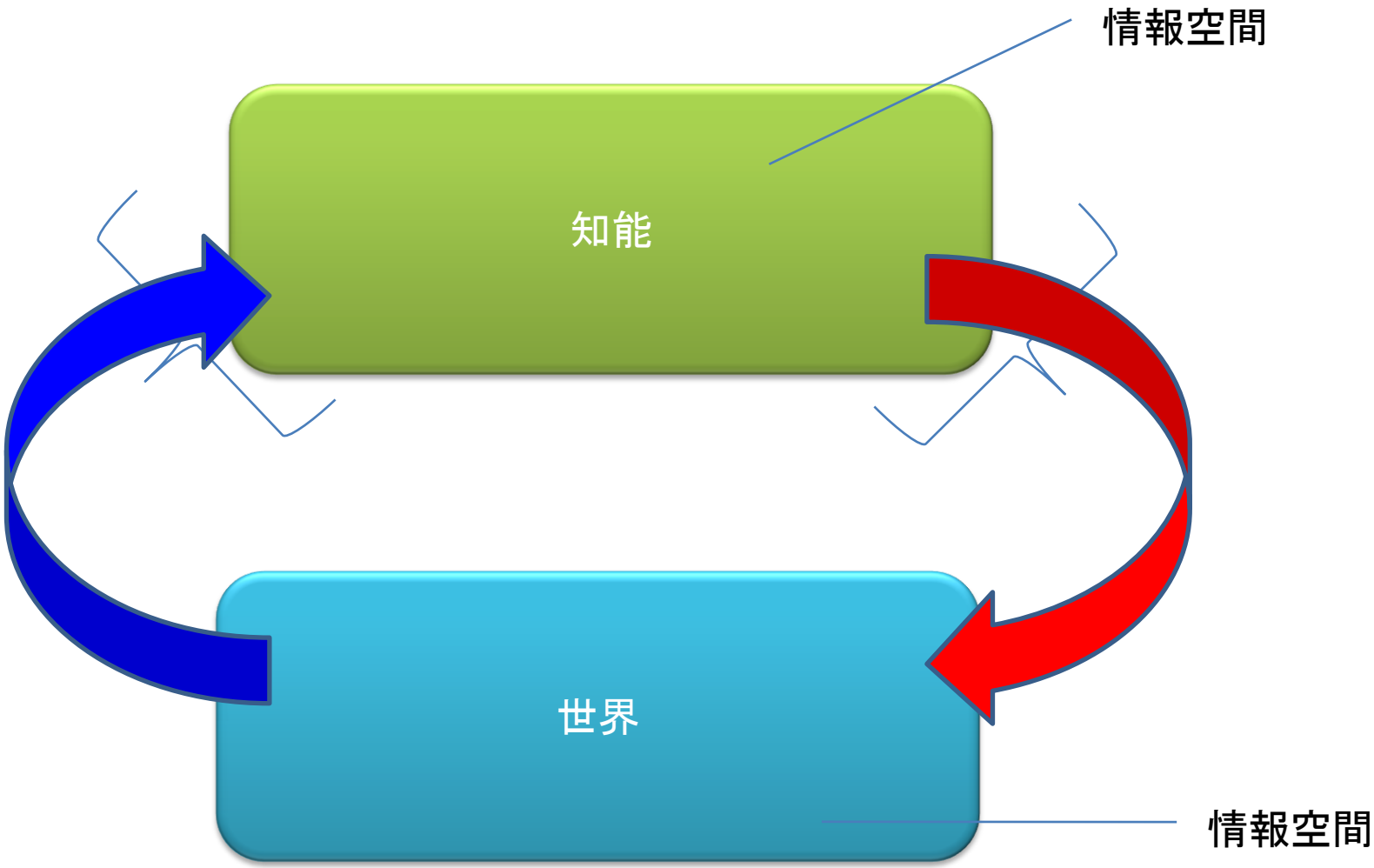
- リアルタイムAIであること。
- 外界を認識する必要があること。

実時間（リアルタイム）で動作するAIとはどんなものか。

- ゲームのリアルタイム　＝1/30, 1/60（秒）で一つの処理をまわす
- ロボット＝本当のリアルタイム（モーター処理、ステップ処理、物理系」）

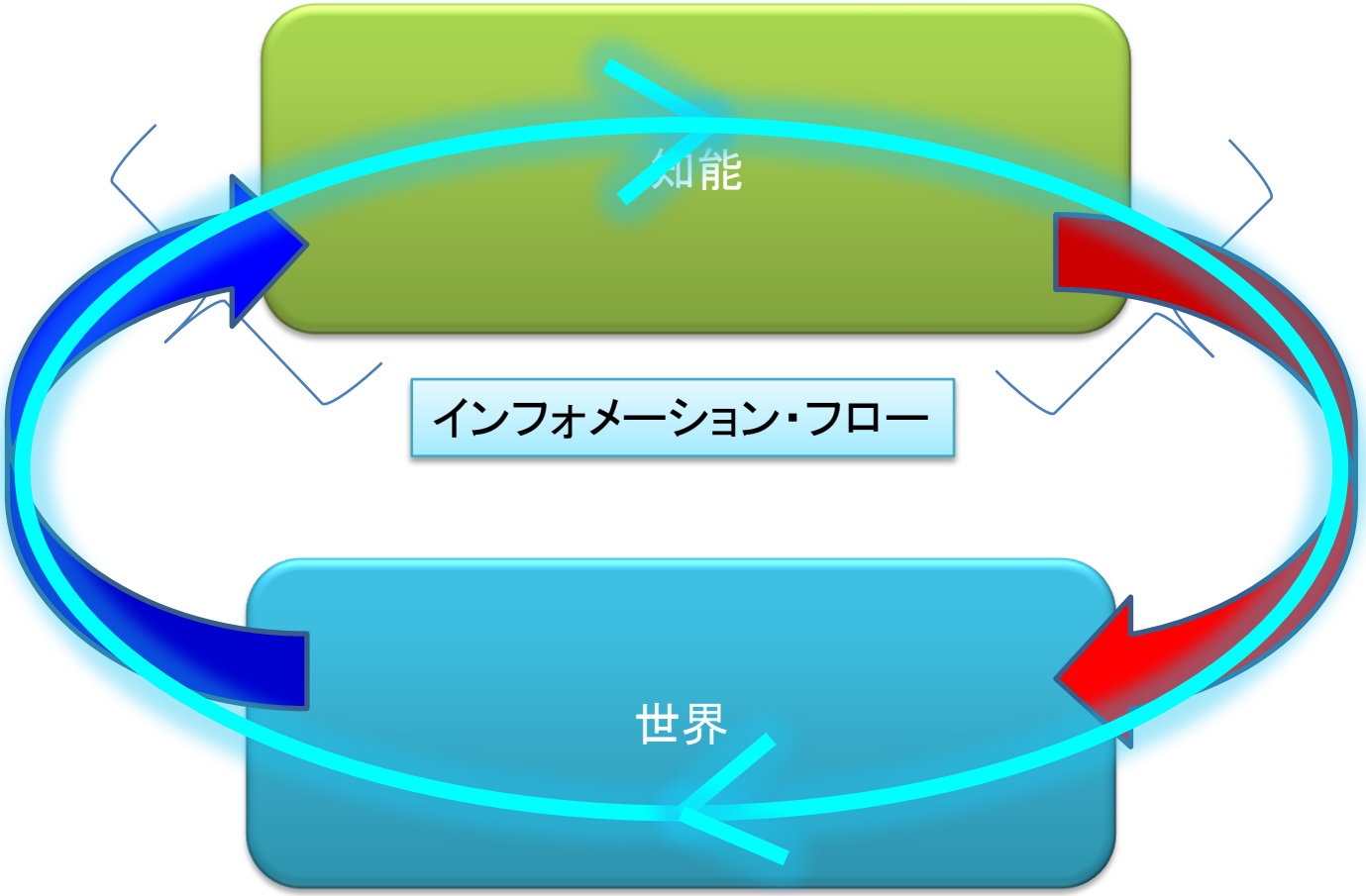
運動と知覚

- どのように運動が知覚と世界を形成していくか。



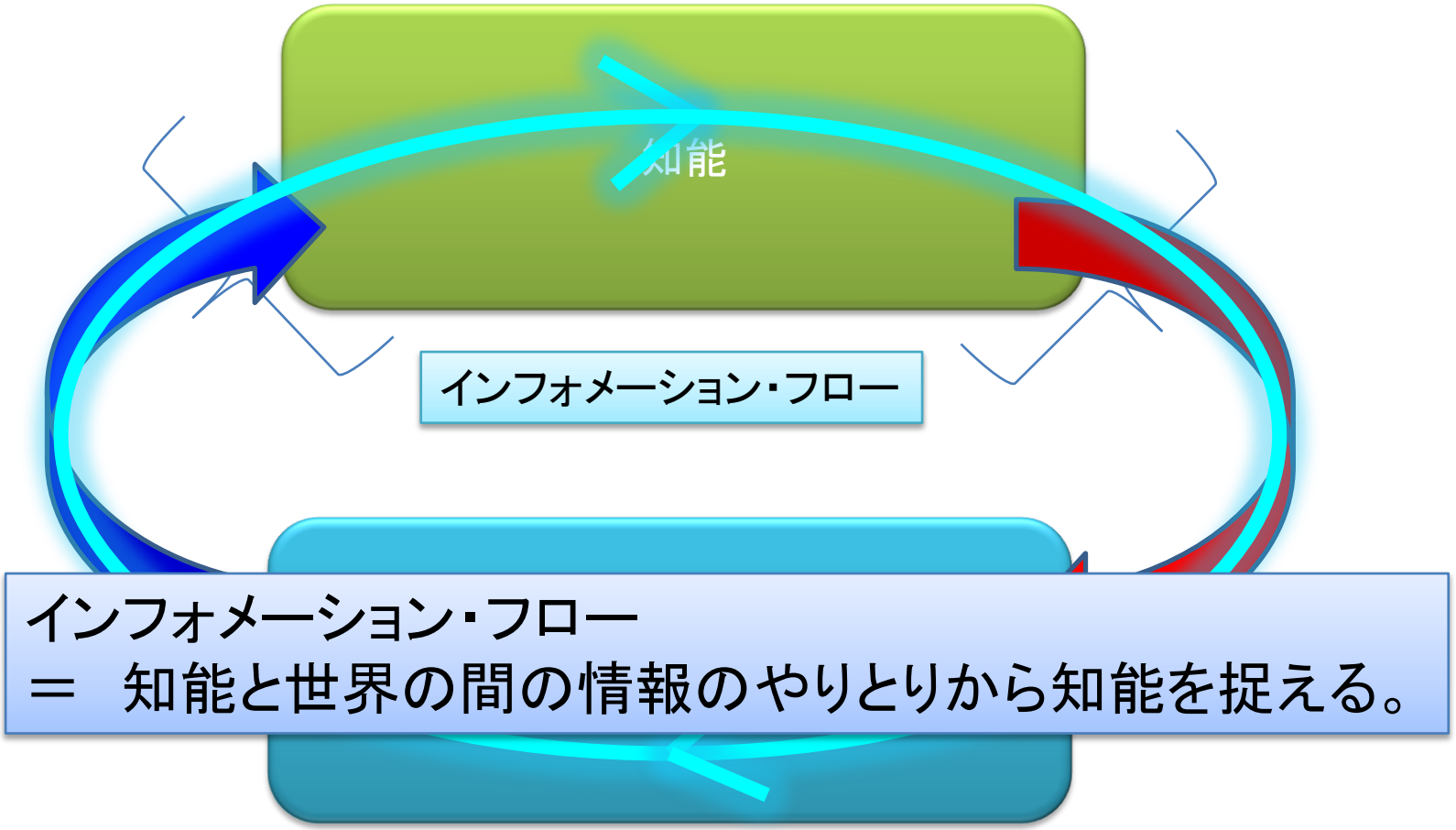
運動と知覚

- どのように運動が知覚と世界を形成していくか。



運動と知覚

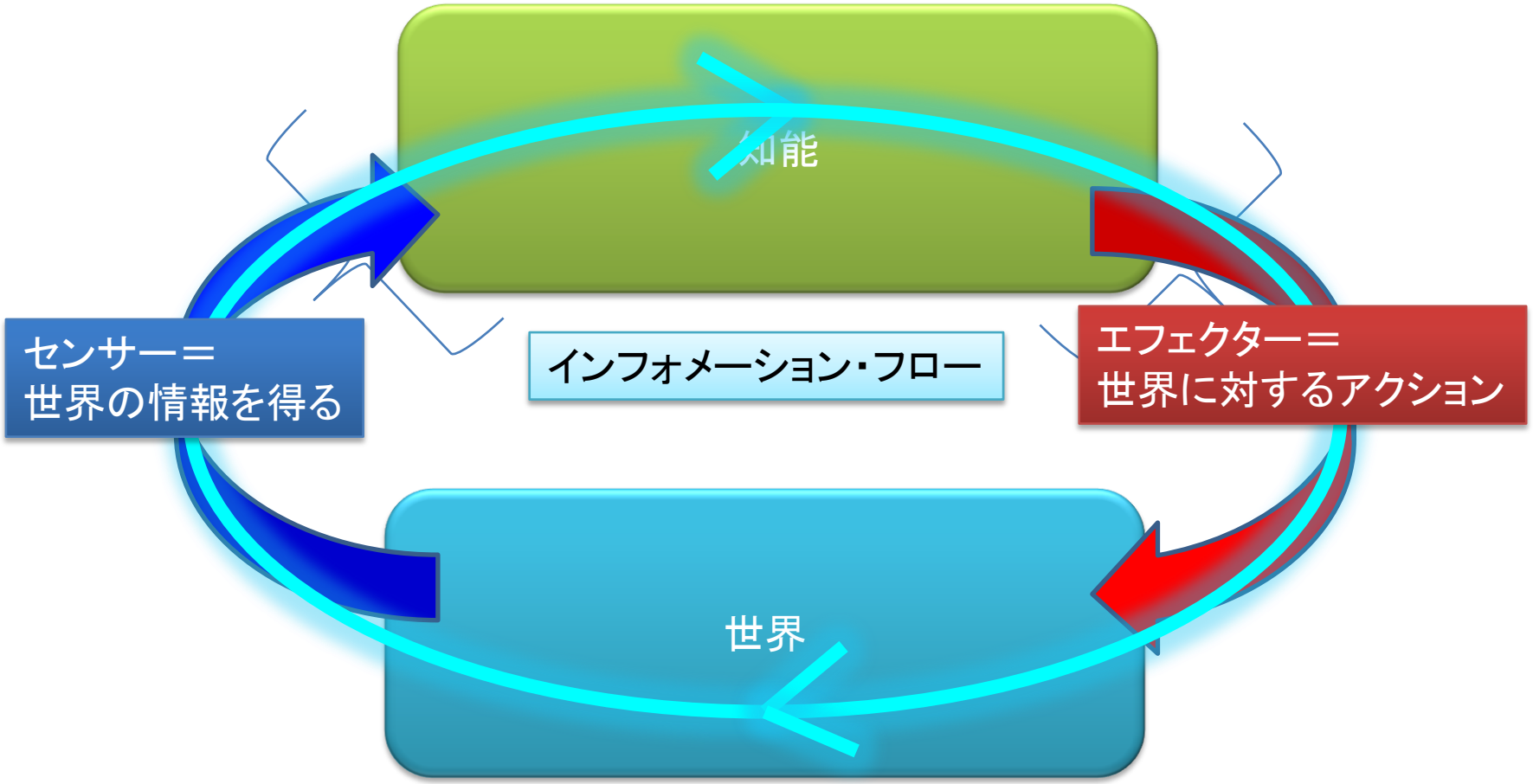
- どのように運動が知覚と世界を形成していくか。



インフォメーション・フロー
＝　知能と世界の間の情報のやりとりから知能を捉える。

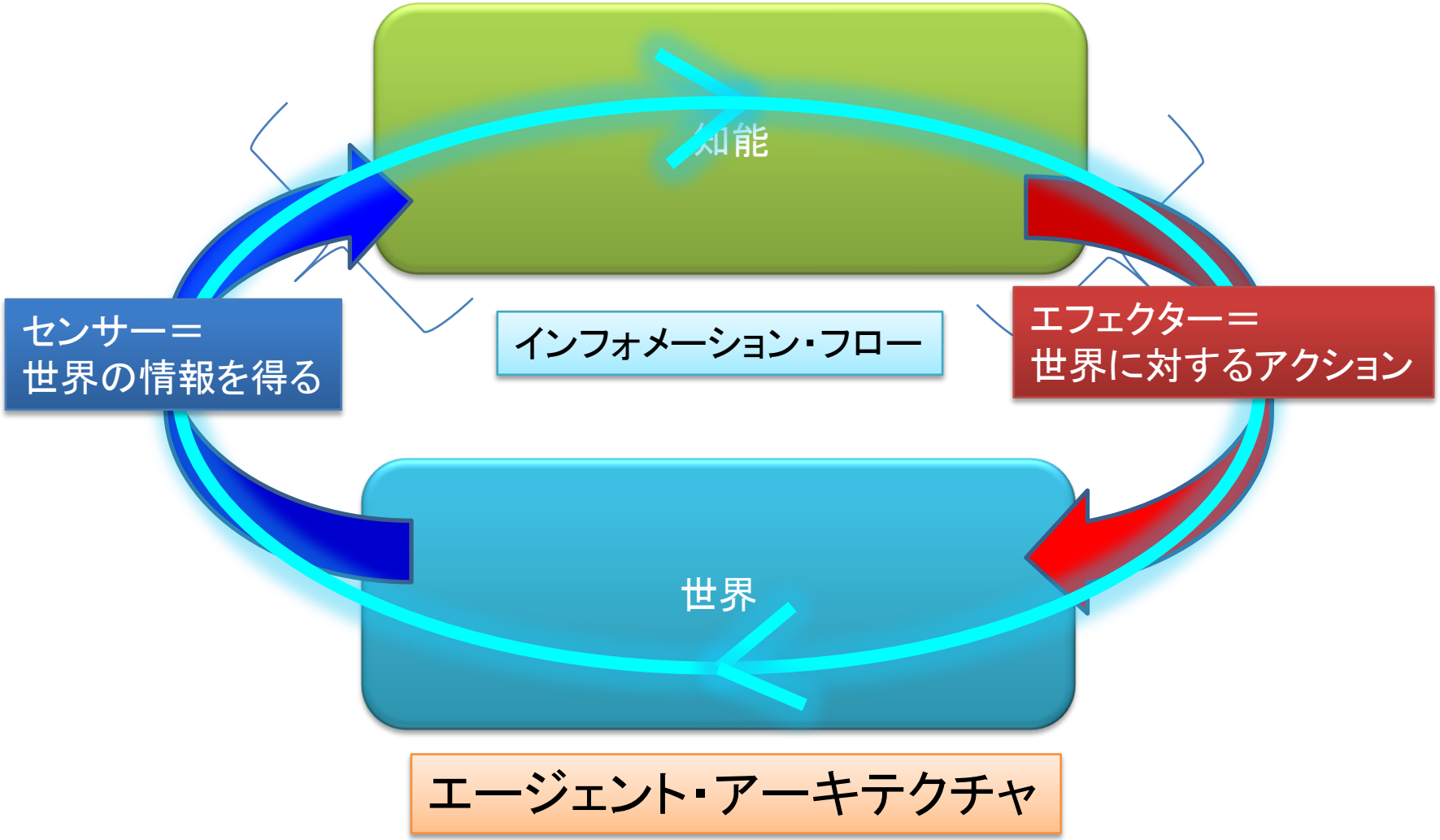
運動と知覚

- どのように運動が知覚と世界を形成していくか。



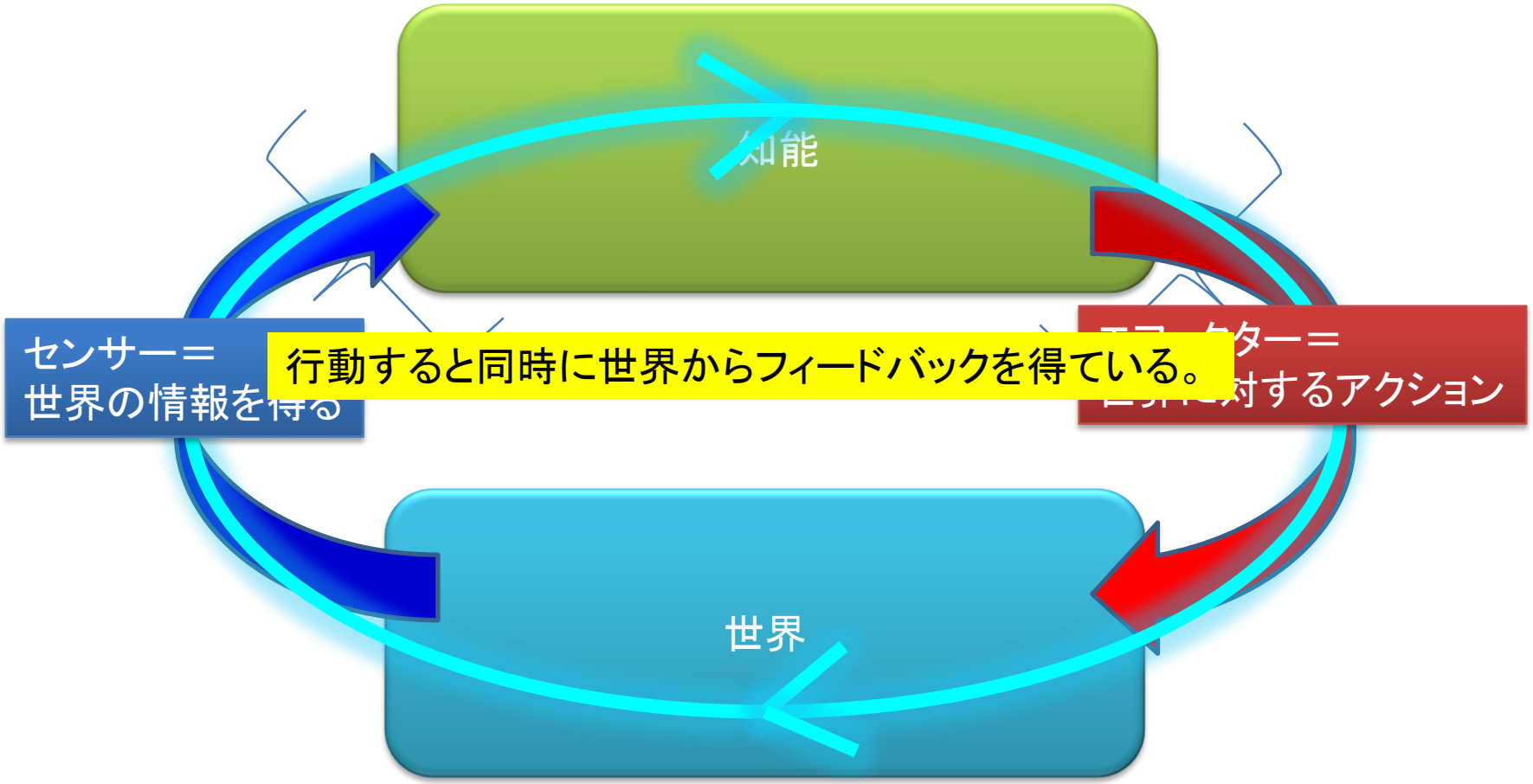
運動と知覚

- どのように運動が知覚と世界を形成していくか。



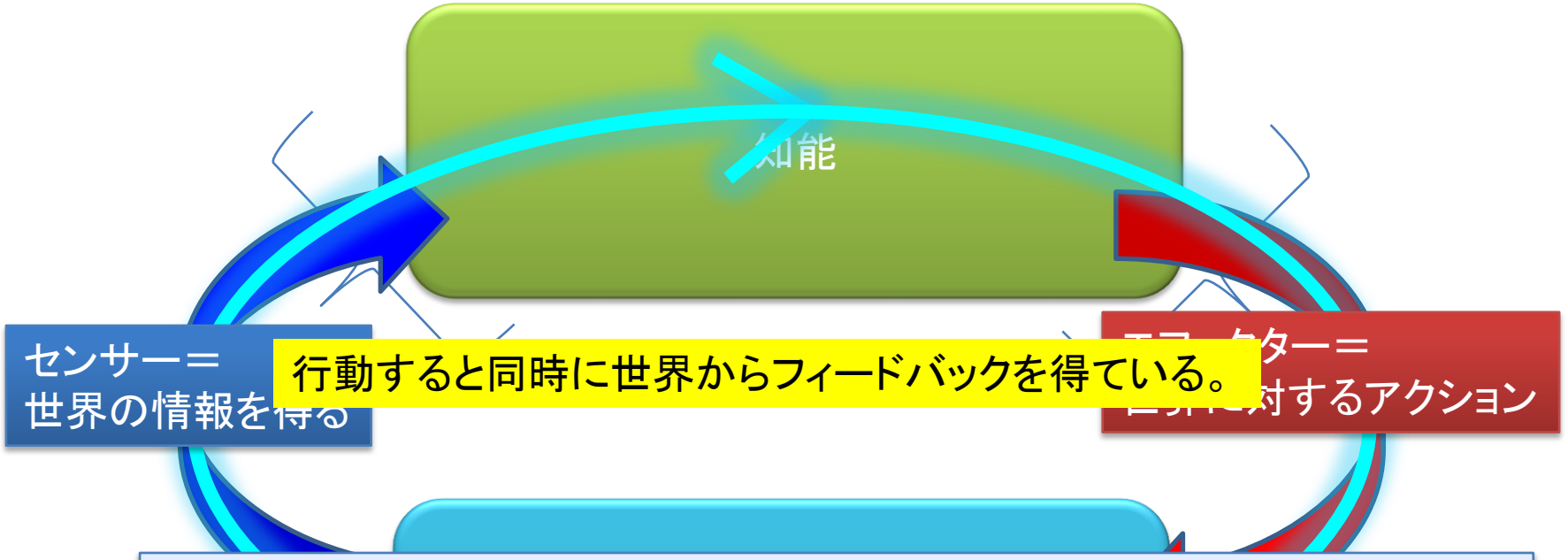
運動と知覚

- どのように運動が知覚と世界を形成していくか。



運動と知覚

- どのように運動が知覚と世界を形成していくか。



インフォメーション・フロー
＝　知能と世界の間の情報のやりとりから知能を捉える。
＝　エフェクターとセンサーはつながっている。
＝　アクションとセンシングは同時に考える。

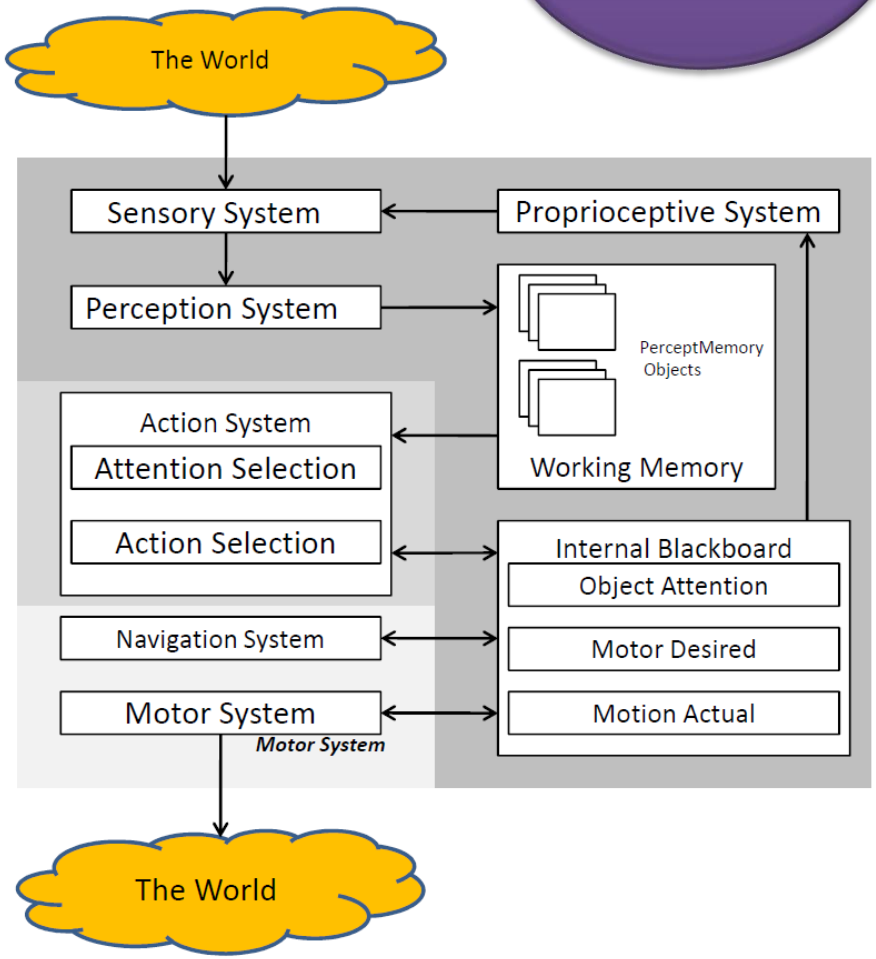
C4 Architecture

黎明期
（GDC2001）

MIT Media Lab.

Synthetic Characters Group

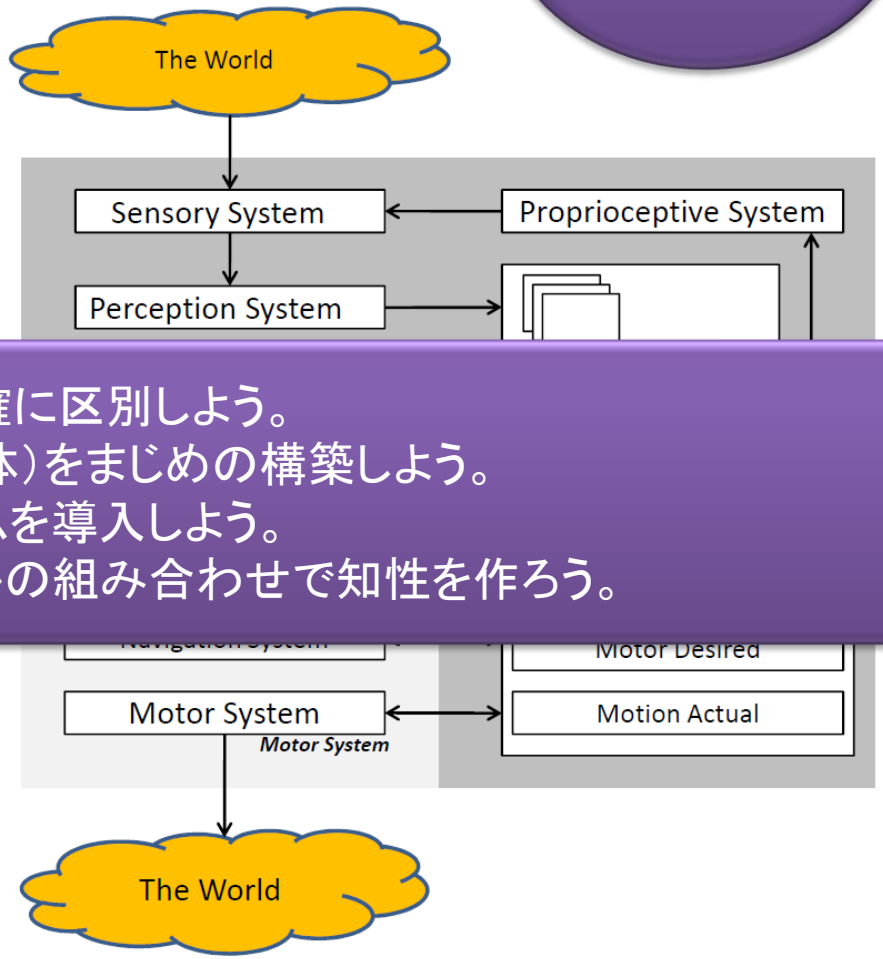
Researching Virtual Pet in Digital World.



D. Isla, R. Burke, M. Downie, B. Blumberg (2001).,
“A Layered Brain Architecture for Synthetic Creatures”, http://characters.media.mit.edu/Papers/ijcai01.pdf

C4 Architecture

黎明期
（GDC2001）



世界と知性を明確に区別しよう。
センサーとエフェクター（身体）をまじめの構築しよう。
記憶のメカニズムを導入しよう。
作り変えやすいように、モジュールの組み合わせで知性を作ろう。

MIT Media Lab.

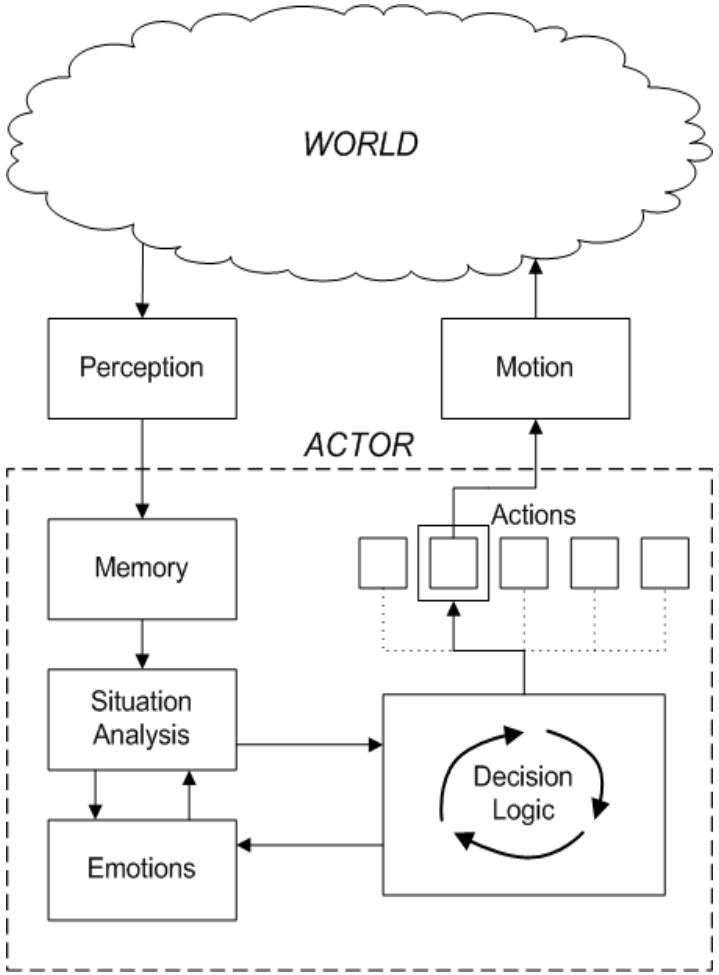
Synthetic Characters Group

Researching Virtual Pet in Digital World.

D. Isla, R. Burke, M. Downie, B. Blumberg (2001).,
“A Layered Brain Architecture for Synthetic Creatures”, http://characters.media.mit.edu/Papers/ijcai01.pdf

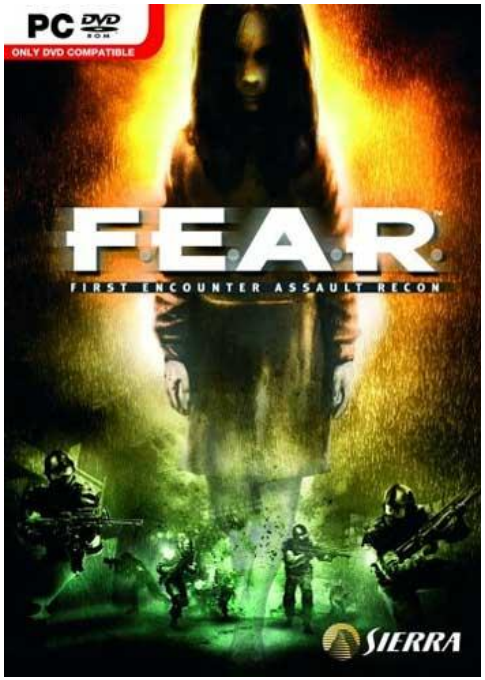
Halo Agent Architecture

- Genre : SciFi-FPS
- Developer : BUNGIE Studio
- Publisher : Microsoft
- Hardware: Xbox, Windows, Mac
- Year : 2002

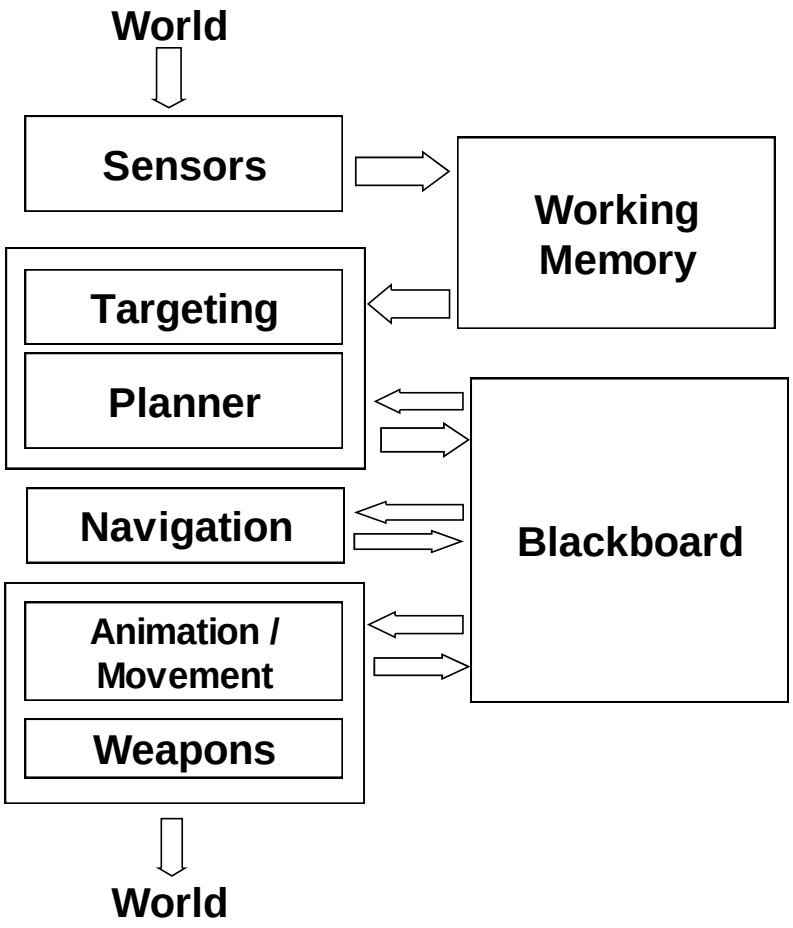


Jaime Griesemer(GDC 2002),The Illusion of Intelligence: The Integration of AI and Level Design in Halo
http://www.bungie.net/Inside/publications.aspx

F.E.A.R Agent Architecture



- Genre : Horror FPS
- Developer : Monolith Production
- Publisher : SIERRA
- Hardware: Windows
- Year : 2004



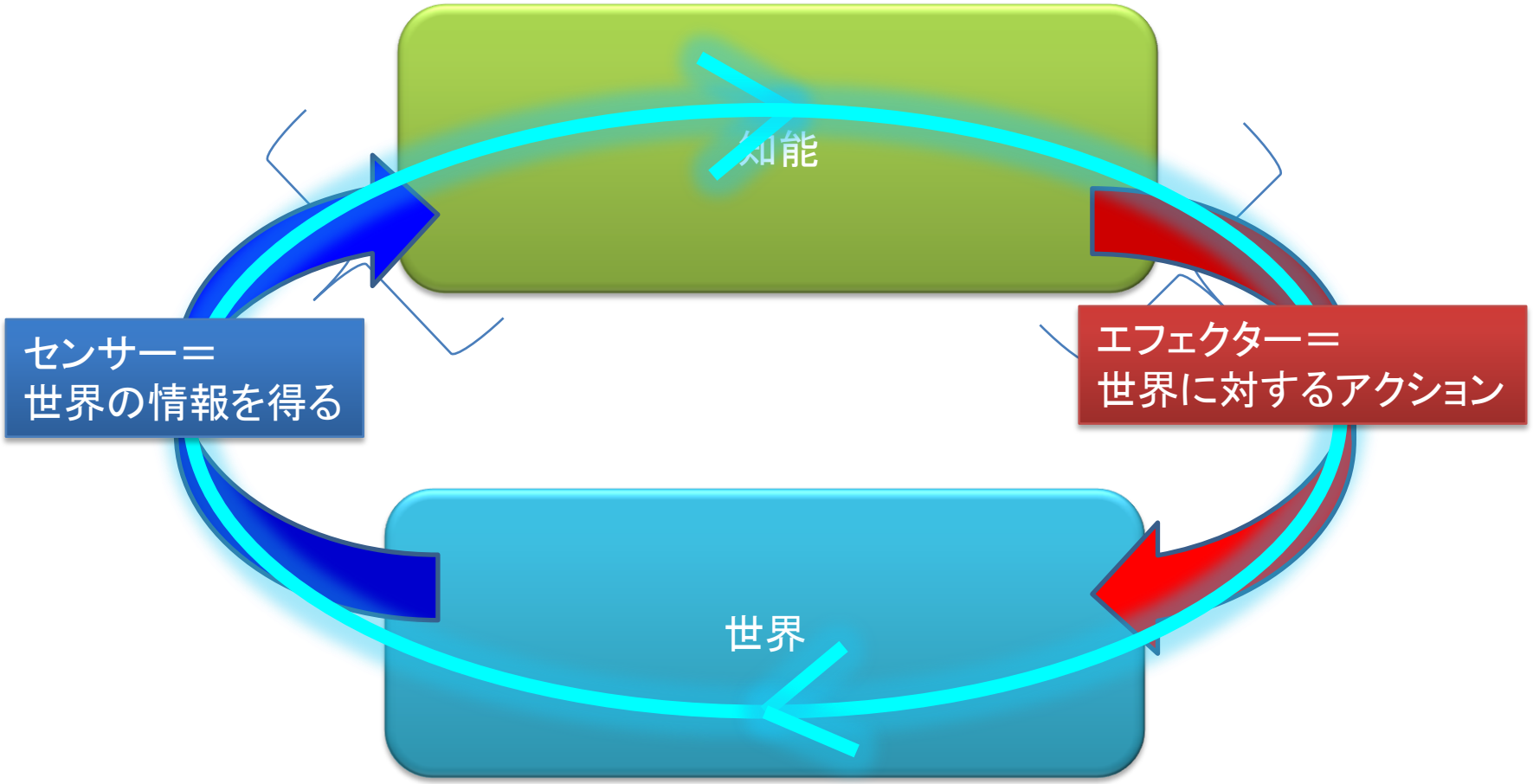
Agent Architecture Considerations for Real-Time Planning in Games (AIIDE 2005)
http://web.media.mit.edu/~jorkin/AIIDE05_Orkin_Planning.ppt

人工知能の世界とのインターフェース

- センシング

運動と知覚

- どのように運動が知覚と世界を形成していくか。

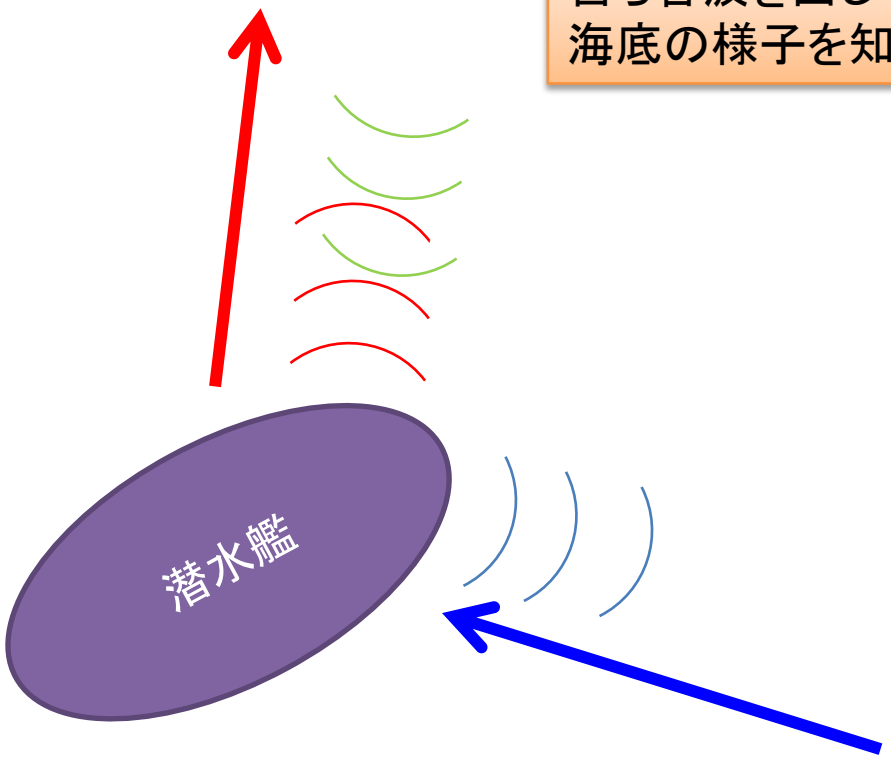


人工知能の世界とのインターフェース

- センシングとアクションの組み合わせ

アクティブ・ソナー、パッシブ・ソナー

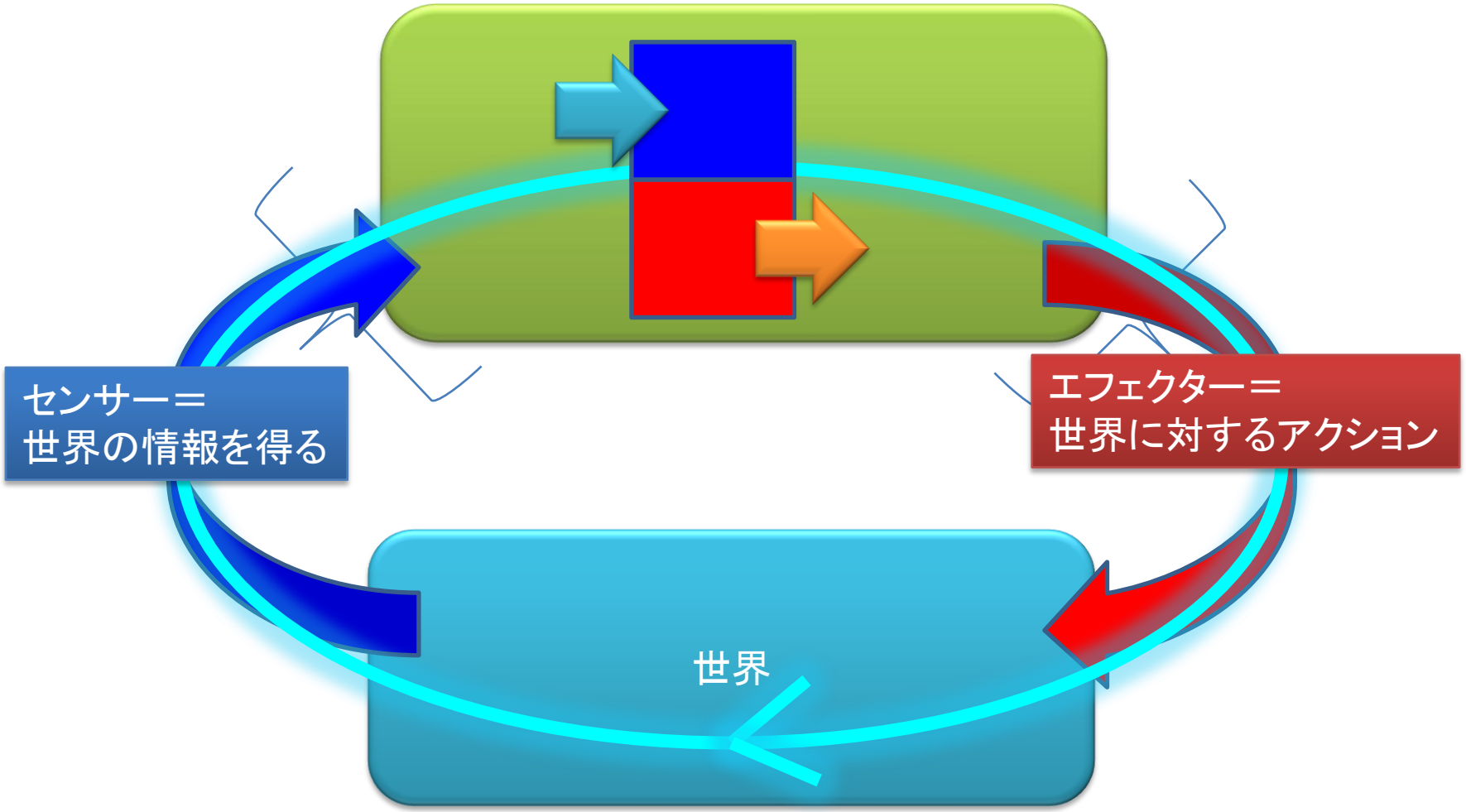
自ら音波を出して、その反射から海底の様子を知る。



センサーを最大感度にして、潜水艦に来る音波に耳を澄ます。

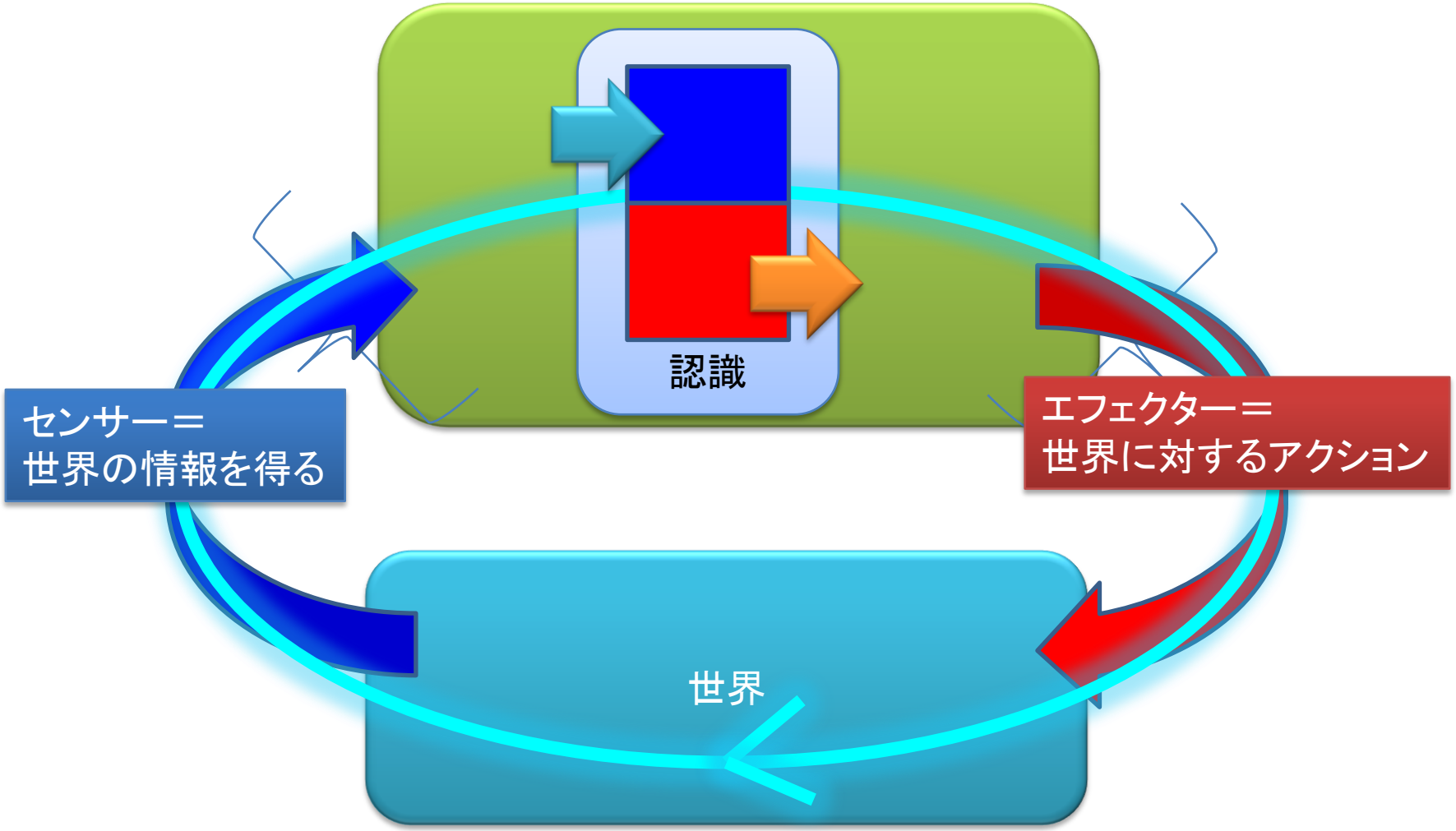
運動と知覚

- どのように運動が知覚と世界を形成していくか。



運動と知覚

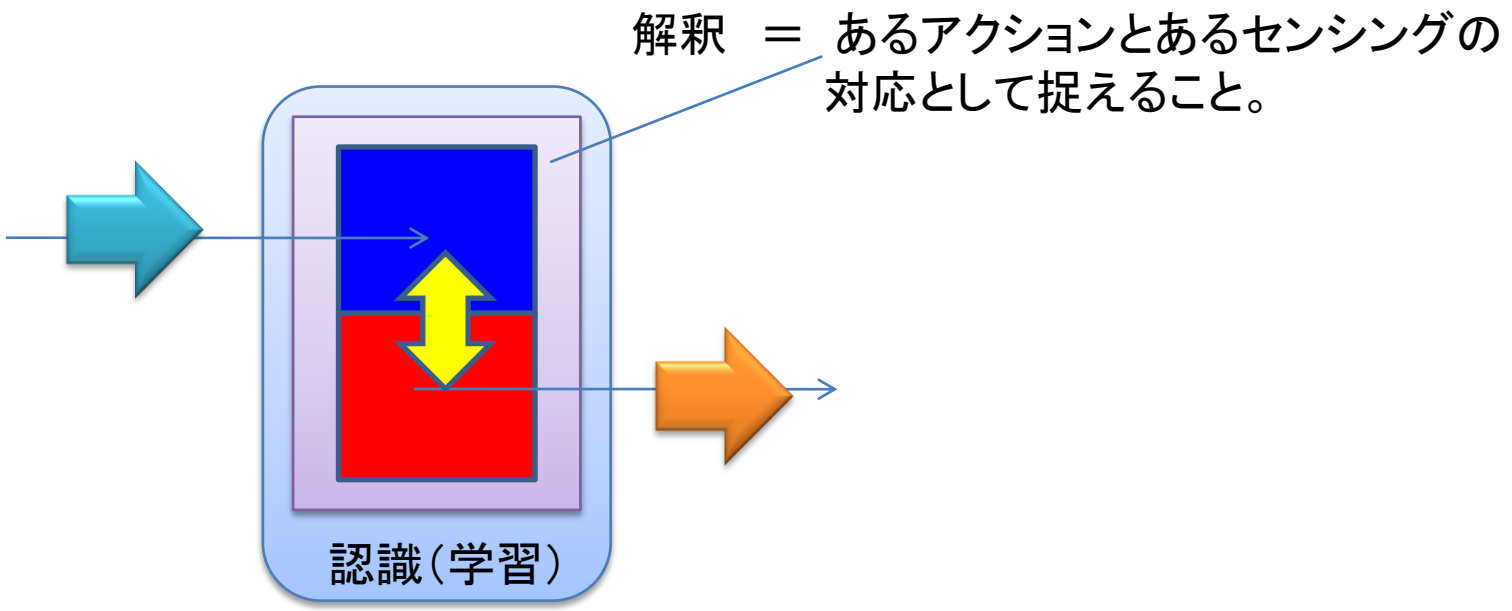
- どのように運動が知覚と世界を形成していくか。



運動と知覚

運動すると同時に知覚している。

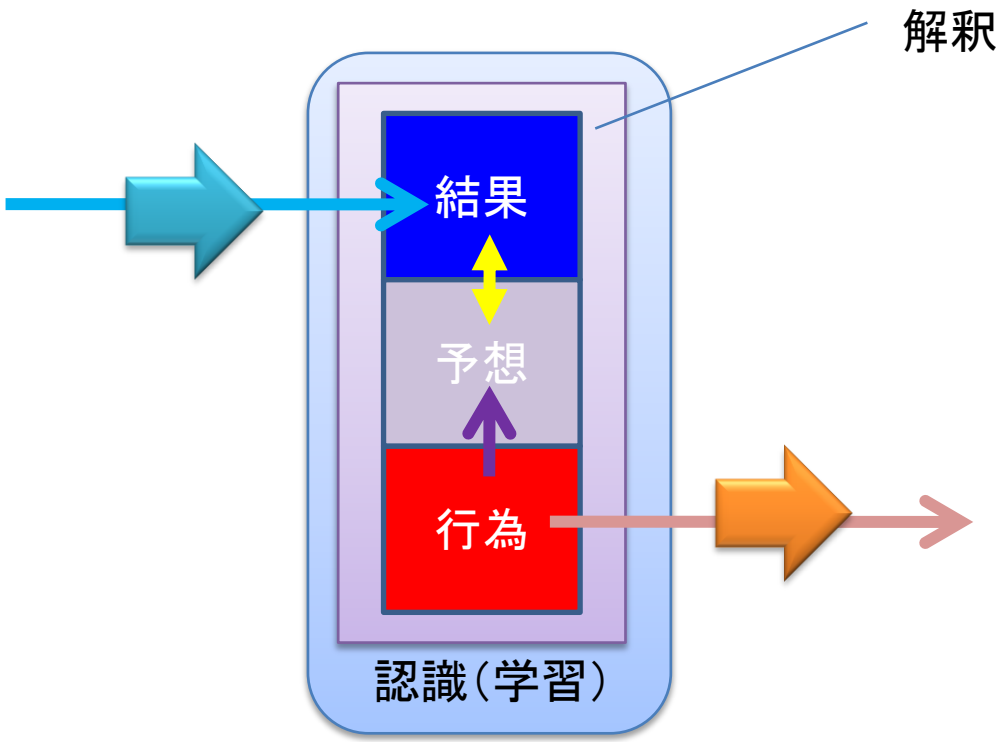
それがインフォメーションフローの本質。



運動と知覚

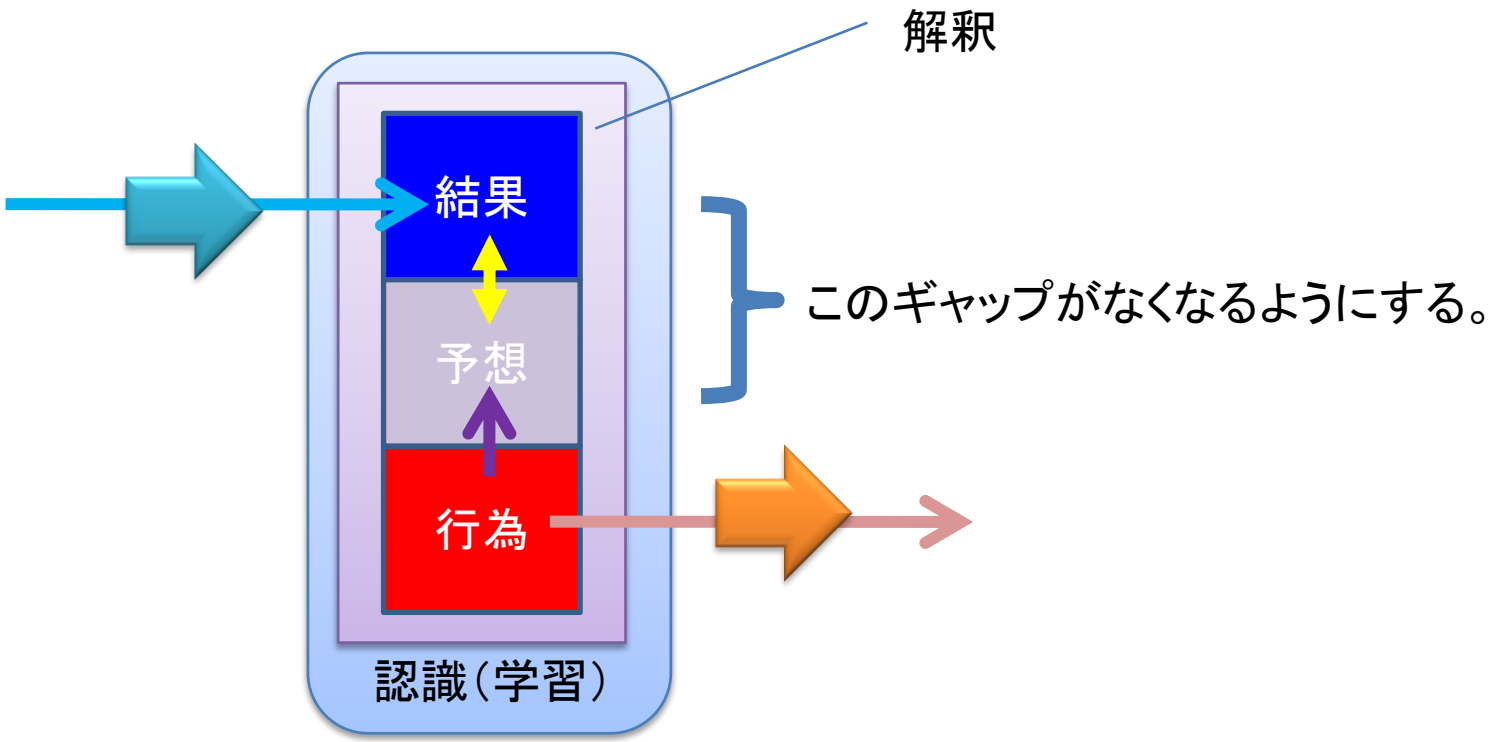
運動すると同時に知覚している。

それがインフォメーションフローの本質。



運動と知覚

運動すると同時に知覚している。
それがインフォメーションフローの本質。

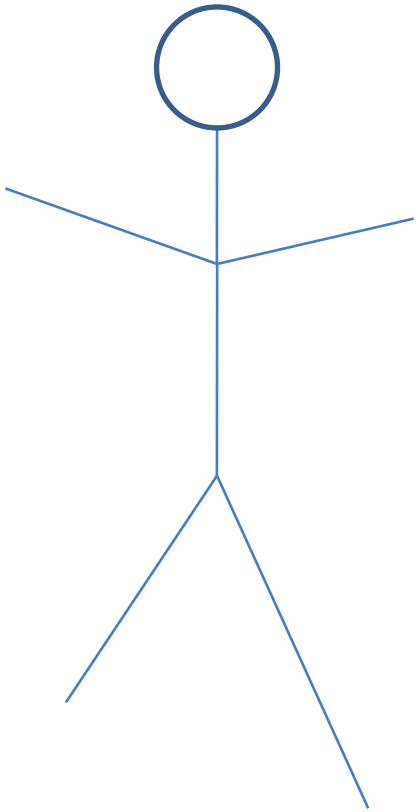


行為と同時にセンサーが身構えている。

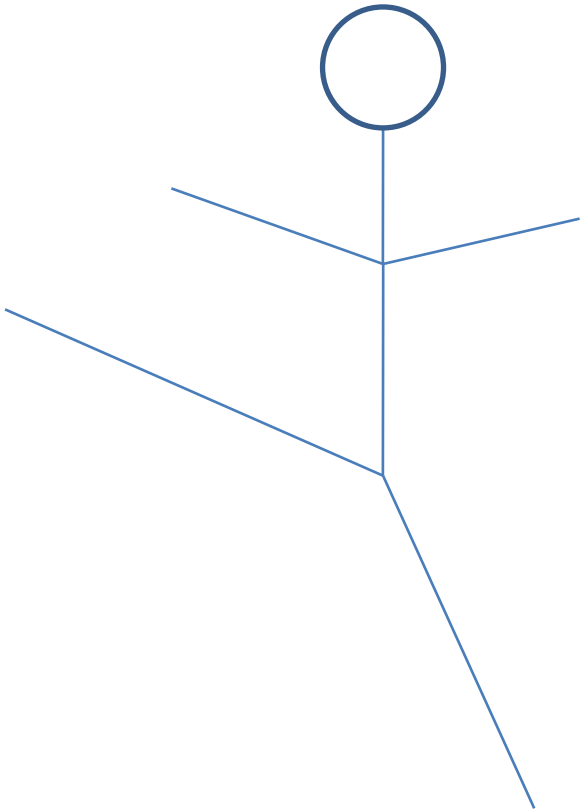
センシングとアクション

- インフォメーション・フローの上ですべては繋がっている。
- 行動すること＝同時に情報取得。
- 行動しようとすることは、得るべき情報セットに感度を張ることでもある。
- センシングとアクションの間に思考がある。

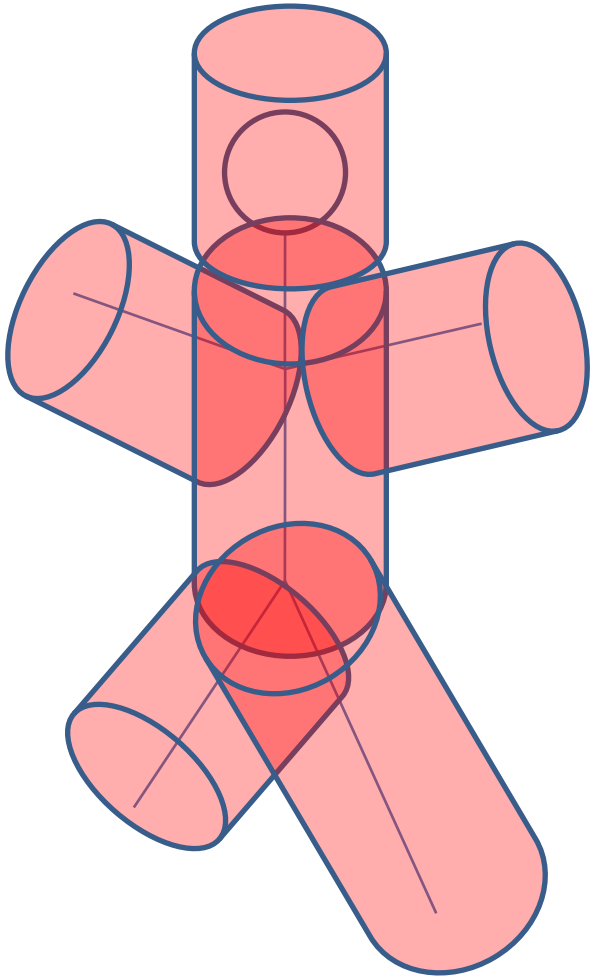
[例] キャラクターの運動



[例]キャラクターの運動

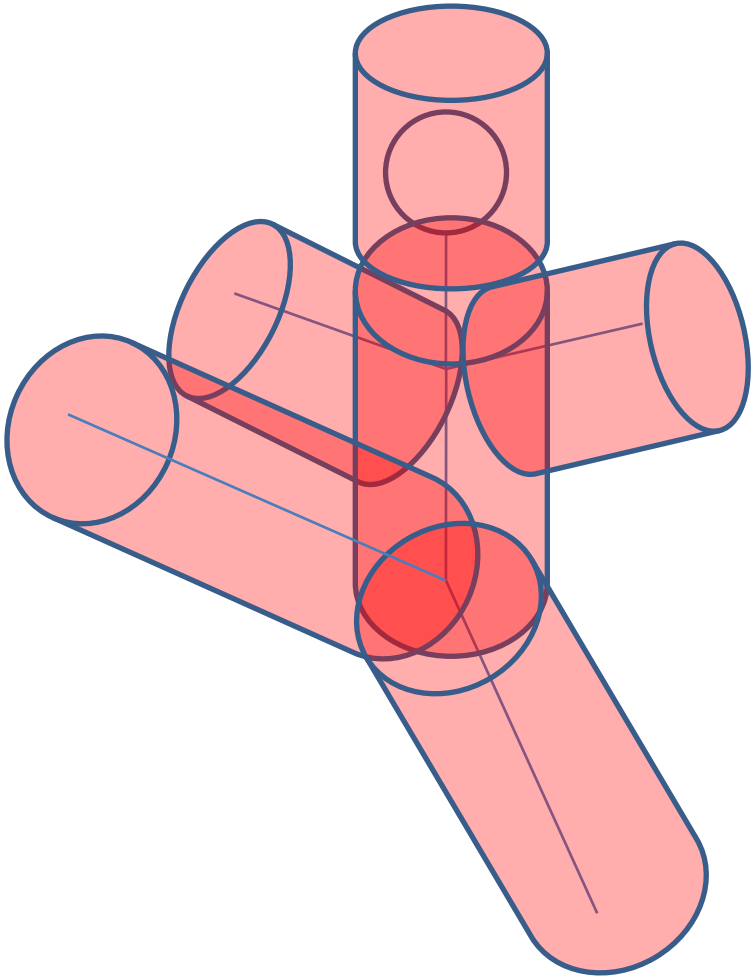


[例] キャラクターの当たりモデル



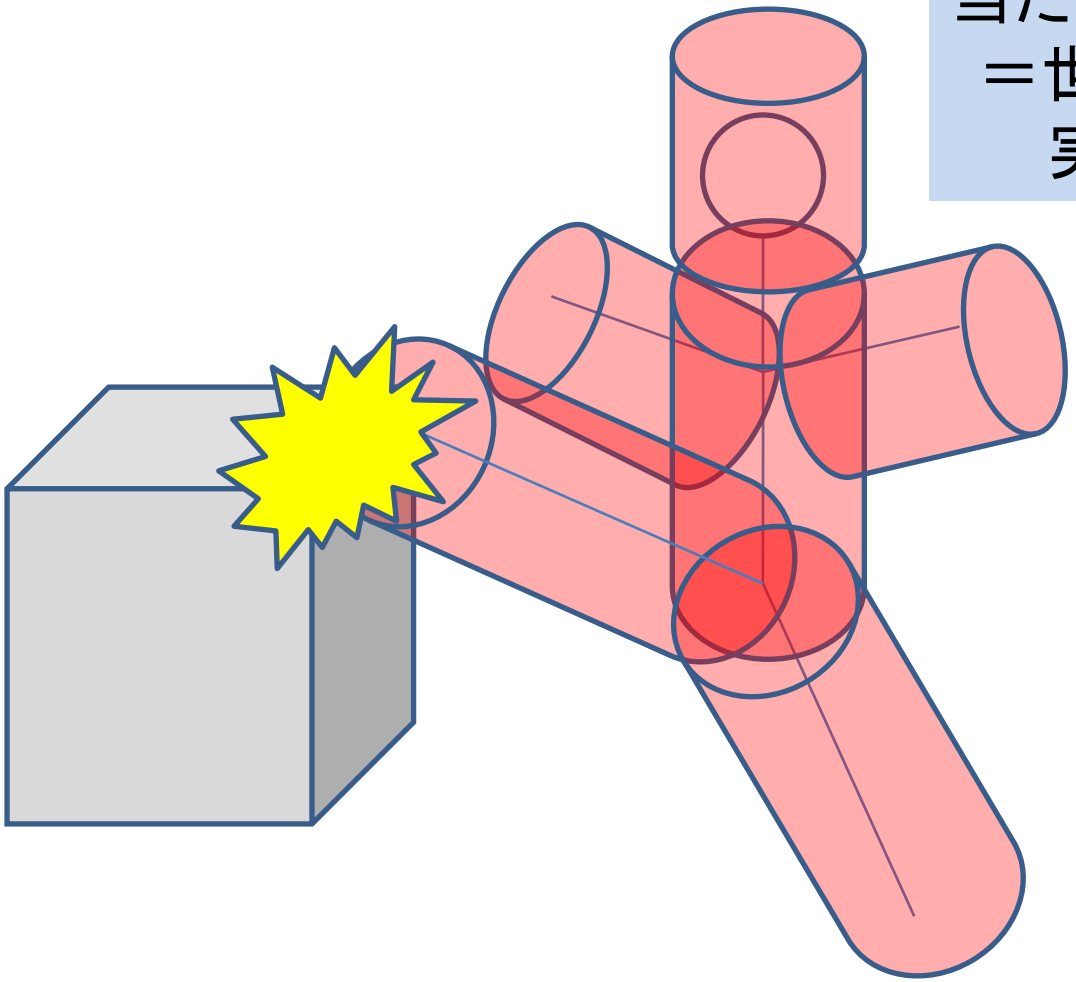
当たりモデル

[例] キャラクターの当たりモデル

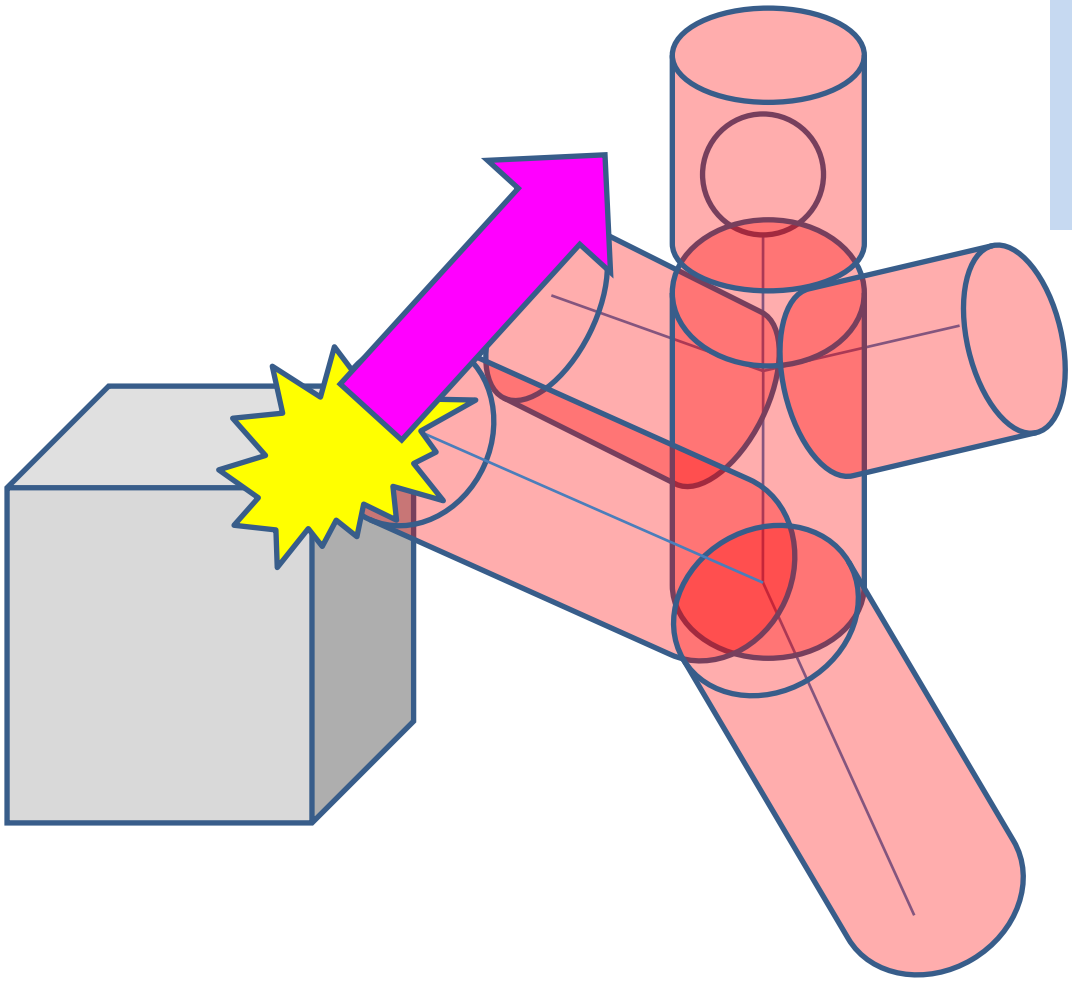


[例] キャラクターの当たりモデル

当たりモデル
＝世界とのインタラクションを
実現する簡易モデル



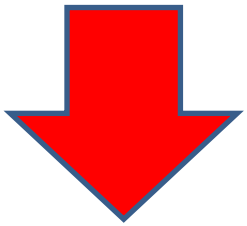
[例] キャラクターの当たりモデル



当たりモデル
＝世界とのインタラクションを実現する簡易モデル
＝ 世界からのセンシングを得る仕組みでもある。

主観的知を形成する

- 自分の運動可能性から環境に対する認識を形成する。



行動を環境と身体の関係性の上で記述する。

運動

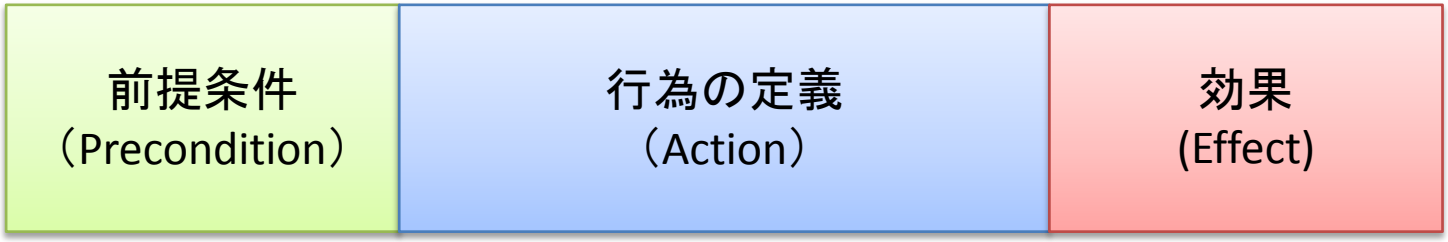
（例） 飛んでいるときと、歩いているとき、休んでいる時と、可能なアクションの種類は違う。

（運動可能性）

（例） 一つのアクションでも、周囲の状況によって結果が違う。

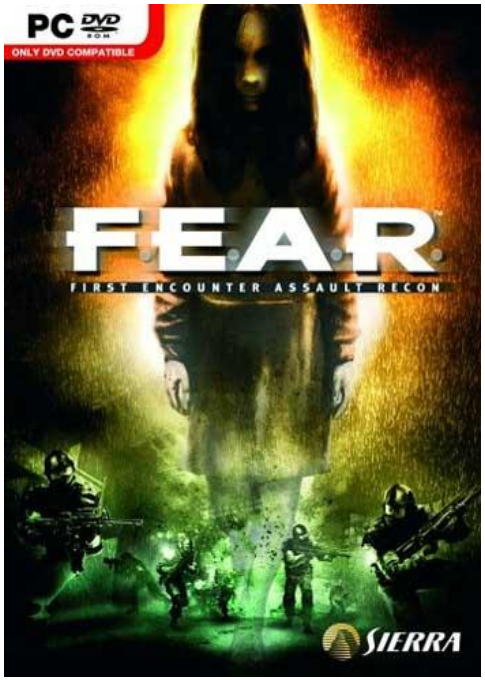
（効果の可能性）

運動の形式

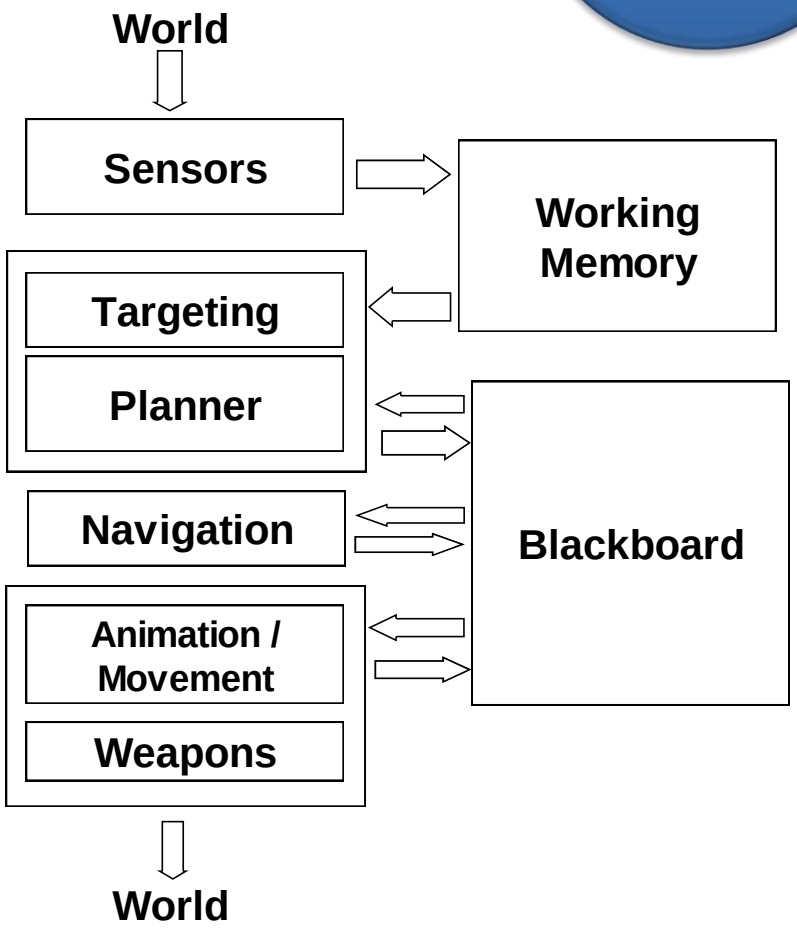


F.E.A.R Agent Architecture

発展期
(GDC2004)



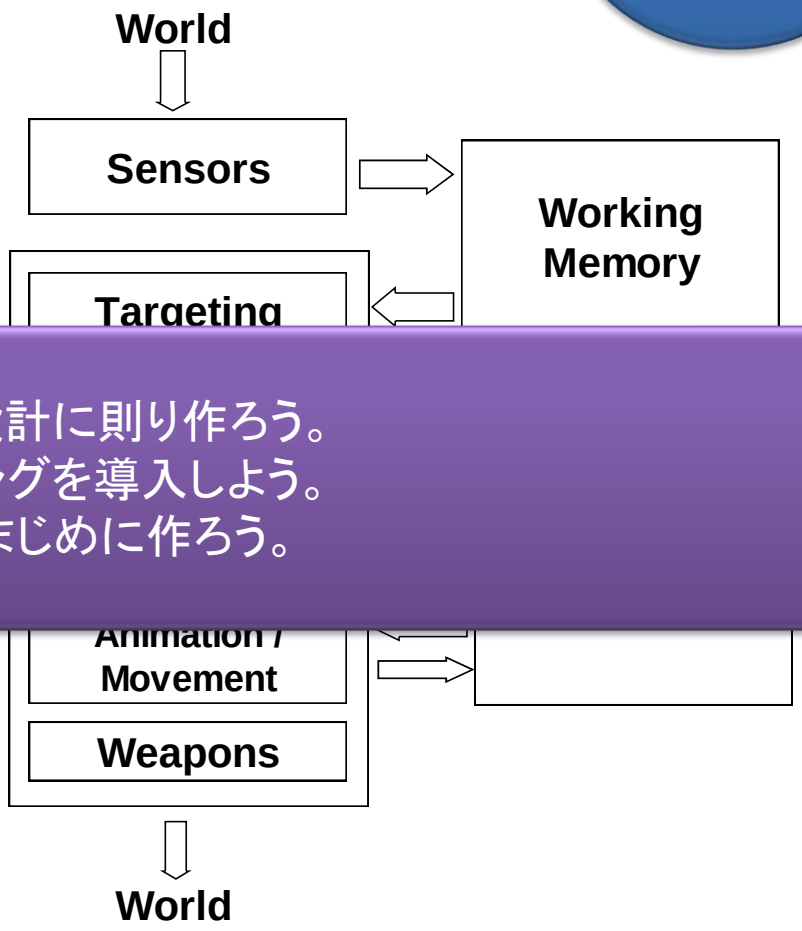
- Genre : Horror FPS
- Developer : Monolith Production
- Publisher : SIERRA
- Hardware: Windows
- Year : 2004



Agent Architecture Considerations for Real-Time Planning in Games (AIIDE 2005)
http://web.media.mit.edu/~jorkin/AIIDE05_Orkin_Planning.ppt

F.E.A.R Agent Architecture

発展期
(GDC2004)



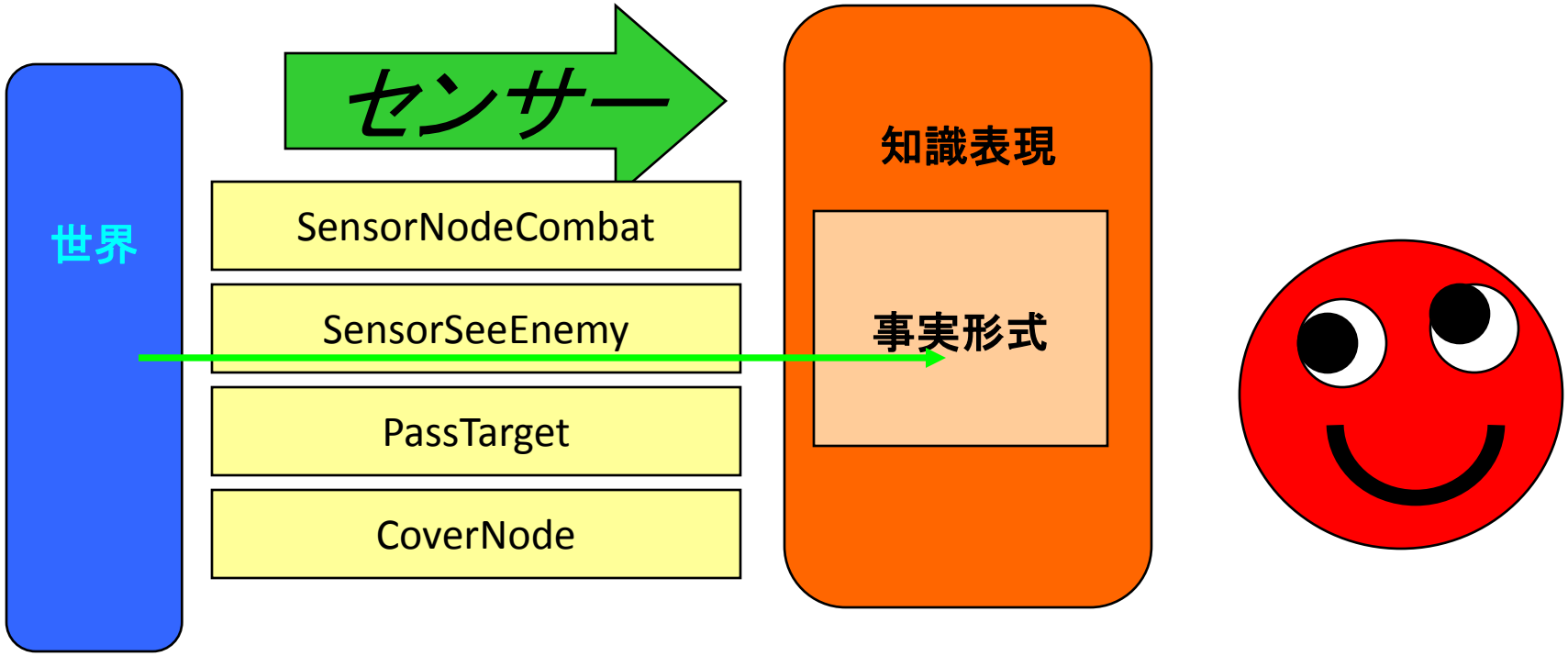
C4アーキテクチャの設計に則り作ろう。
意志決定にプランニングを導入しよう。
ワーキングメモリをまじめに作ろう。

- Genre：Horror FPS
- Developer： Monolith Production
- Publisher ：SIERRA
- Hardware: Windows
- Year： 2004

Agent Architecture Considerations for Real-Time Planning in Games (AIIDE 2005)
http://web.media.mit.edu/~jorkin/AIIDE05_Orkin_Planning.ppt

F.E.A.R のCOMの感覚（センサー）

センサー ＝ 五感からの情報、及び、五感から得られるはずの情報を模擬する機能



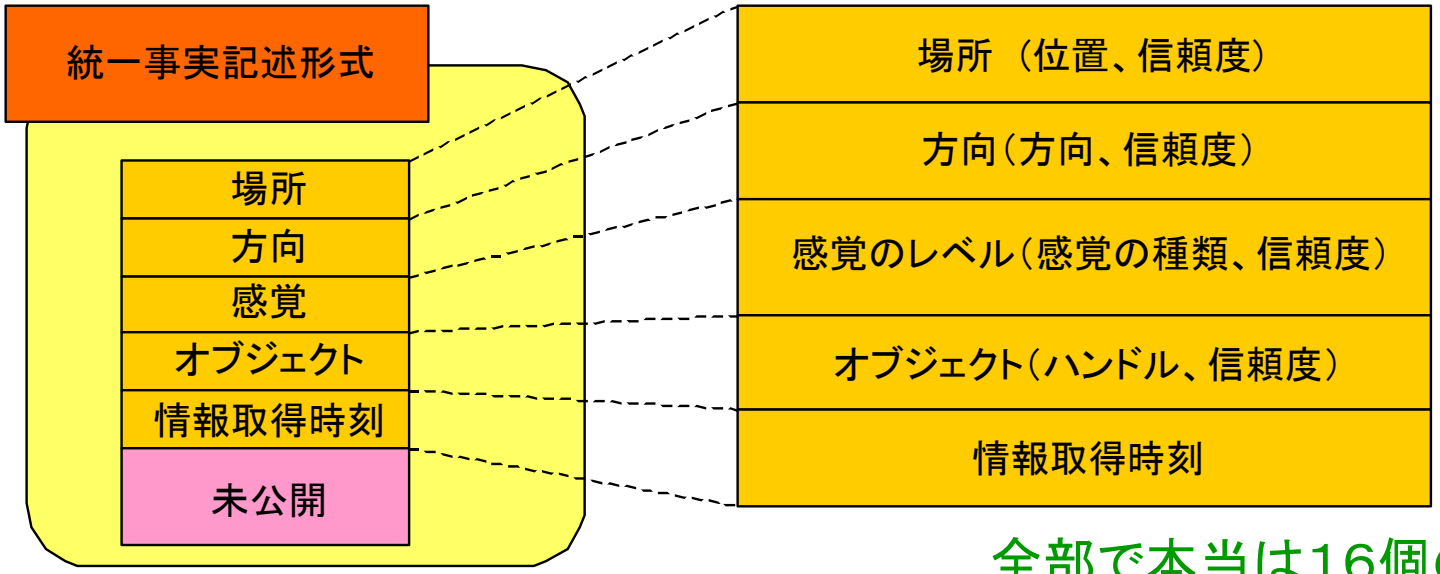
レイキャスト（視線チェック）、パス検索など重たい処理からなる関数

SensorNodeCombat　隠れる、或いは、隠れながら攻撃できる場所を探す。
SensorSeeEnemy　敵が見えるかチェック
PassTarget　戦術ポイントまでのパスを見つけ、かつそのパスが敵から安全であることをチェックする。
CoverNode　隠れることができるノード

統一事実記述形式

キャラクター	オブジェクト	任務	事件	パス	欲求	ノード

全て以下の形式（フォーマット）で記述する。

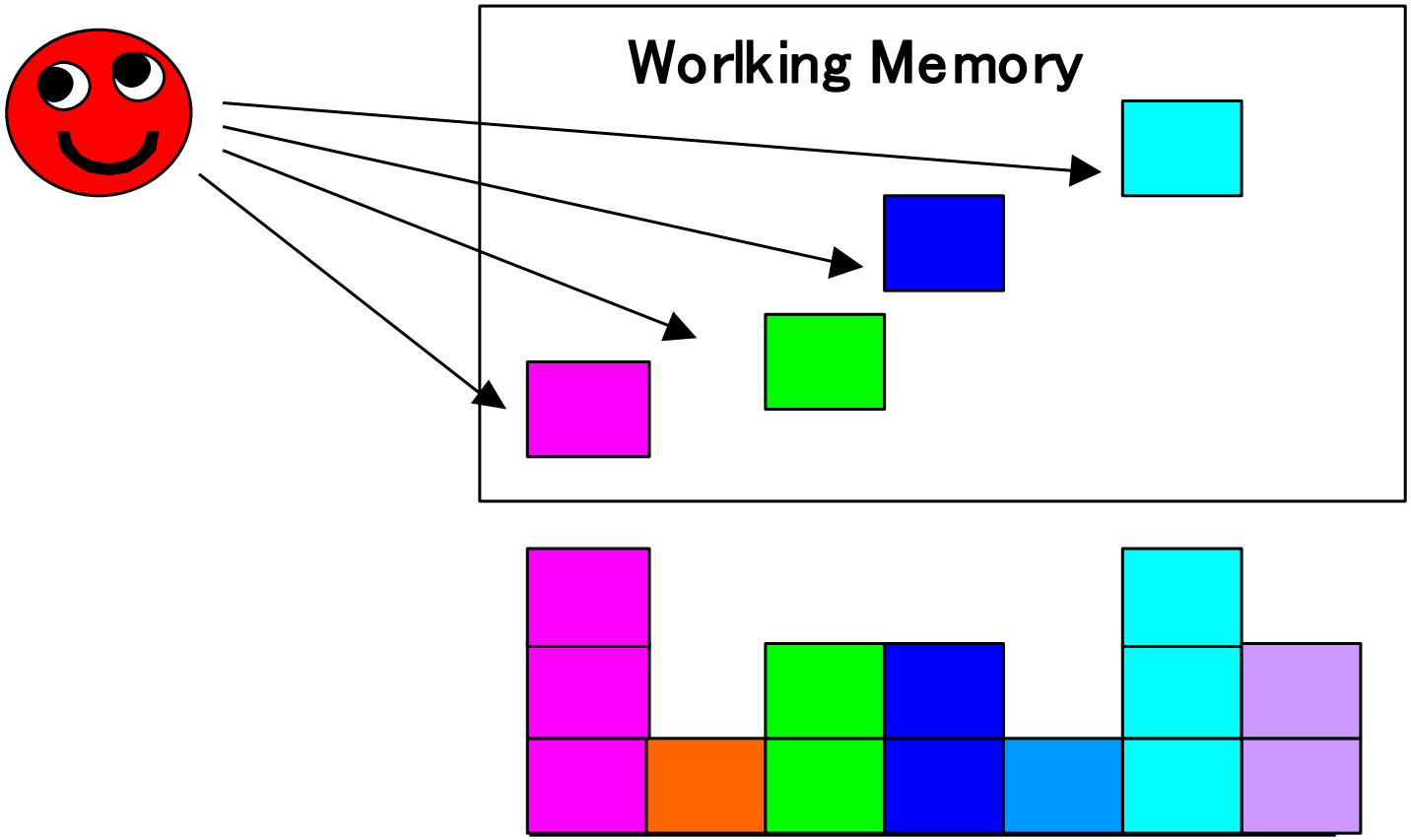


全部で本当は16個の属性がある

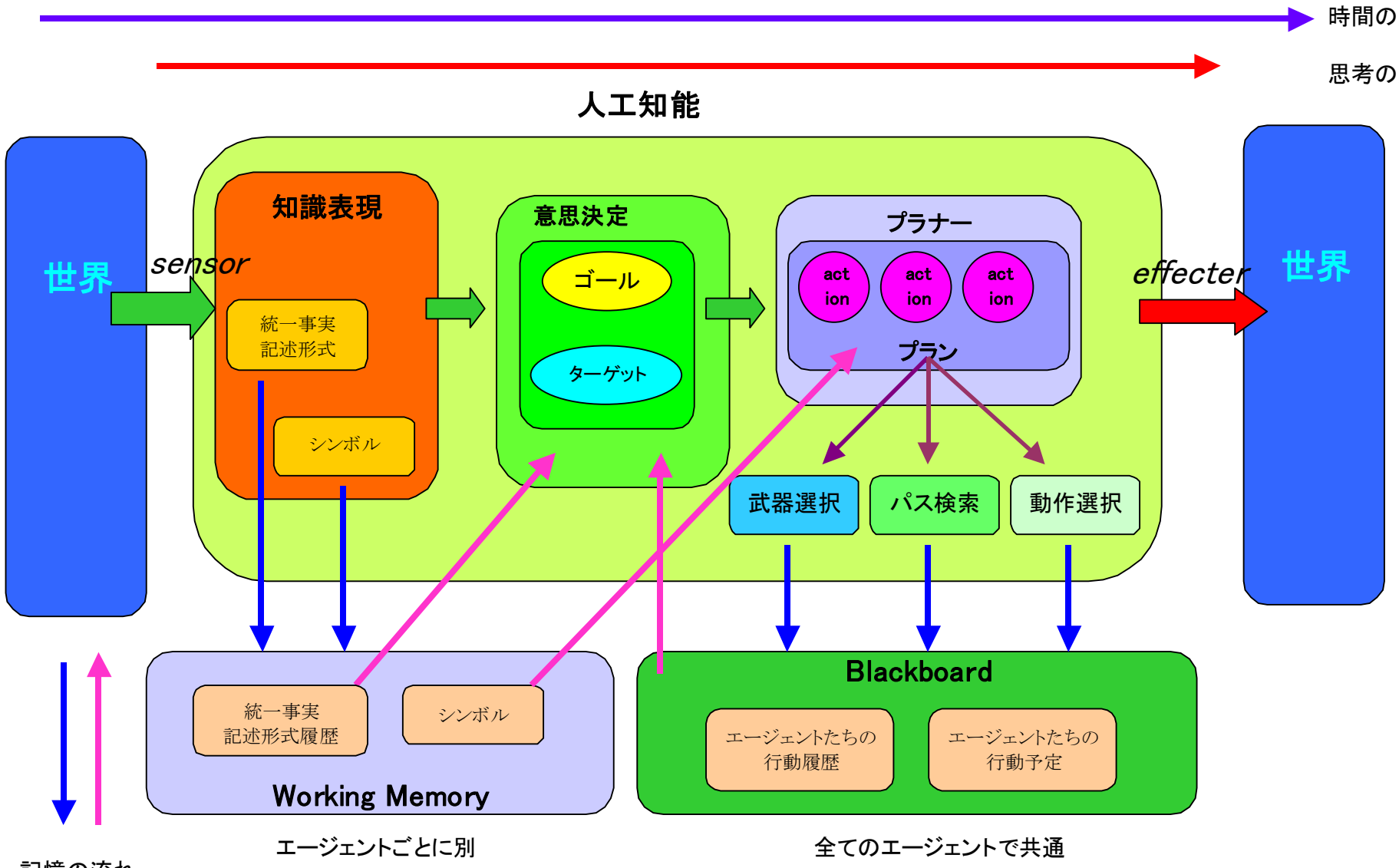
```
WorkingMemoryFact
{
   Attribute<Vector3D>      Position
   Attribute<Vector3D>      Direction
   Attribute<StimulusType>  Stimulus
   Attribute<Handle>        Object
   Attribute<float>         Desire
   ...
   float                fUpdateTime
}
```

```
Attribute<Type>
{
   Type  Value
   float fConfidence
}
```

自分の記憶領域に認識した事実を蓄積する



F.E.A.R AI 全体設計図



F.E.A.R AI 設計図

プランニングのための知識

エージェントの認識する世界をもっとシンプルに表現したい

20個のシンボルで世界を集約して表現する

シンボル

- kSymbol_AtNode
  どのノードにいるか
- kSymbol_
  WeaponArmed
  武装しているか
- kSymbol_Target
  IsSuppressed
  威嚇されているか
- kSymbol_TargetIs
  AimingAtMe
  どのノードにいるか
- kSymbol_AtNodeType
  どんなタイプのノードにいるか
- kSymbol_
  UsingObject
  オブジェクトを
  使っているか？
- kSymbol_
  WeaponLoaded
  装填されているか
- kSymbol_
  RidingVehicle
  乗り物に乗っているか
- kSymbol_
  TargetIsDead
  死んでいるか

各エージェントについて（Agent-centric）

上記のシンボルは、対象とするエージェントについての情報。

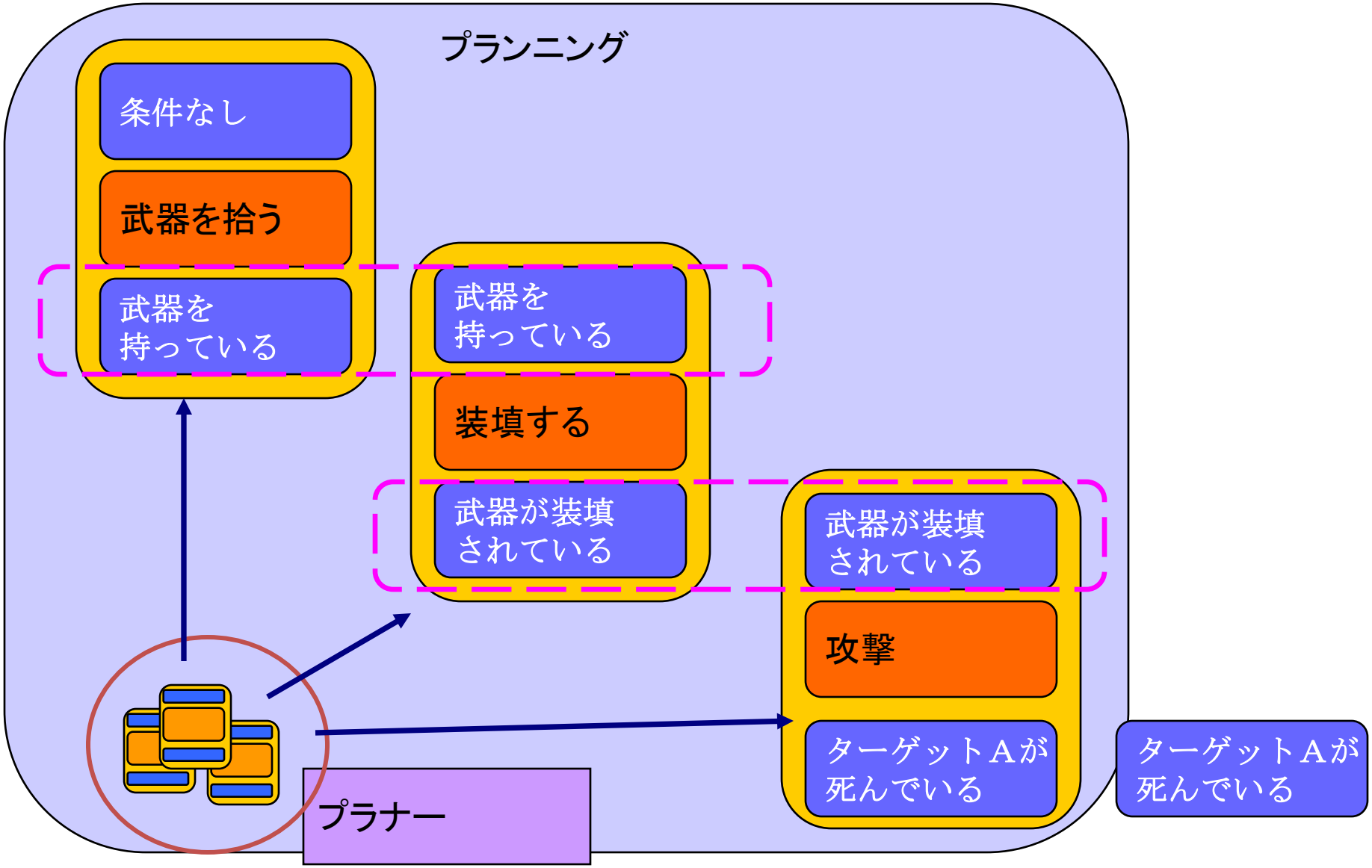
シンボル

kTargetIsDead = ture
この兵士Aは死んだ

kTargetAtme = ture
この兵士Bは自分を狙っている

kWeaponIsLoaded = false
私Cの武器は装填済みでない

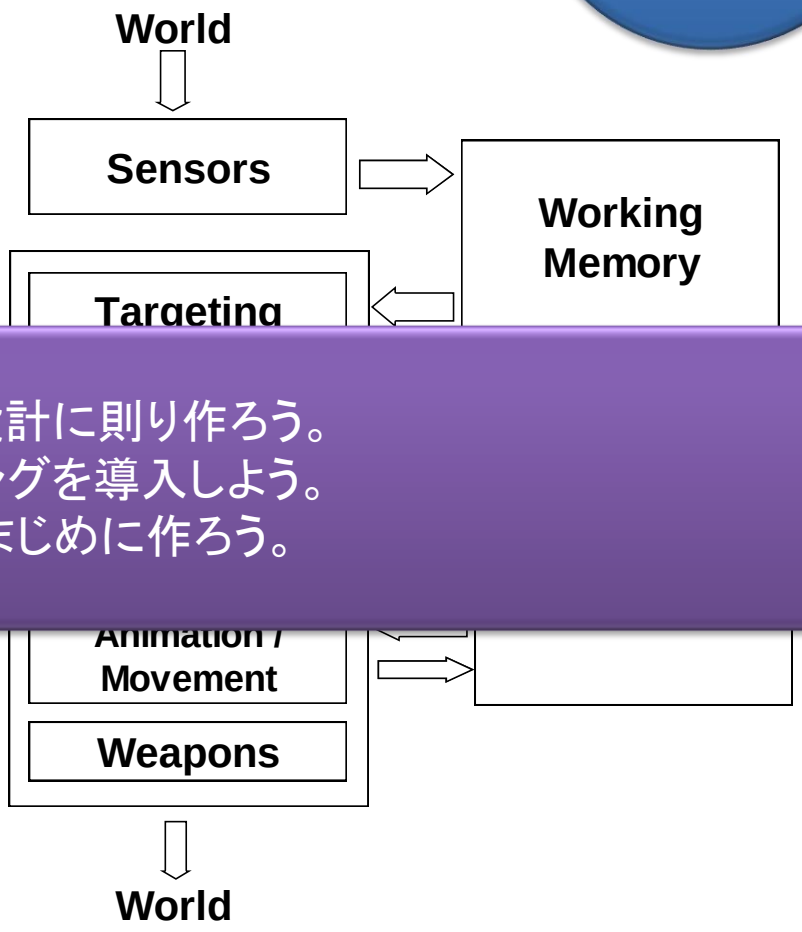
F.E.A.R.のプランニング②
シンボルによる連鎖プランニング



プランニング

F.E.A.R Agent Architecture

発展期



C4アーキテクチャの設計に則り作ろう。
意志決定にプランニングを導入しよう。
ワーキングメモリをまじめに作ろう。

- Genre：Horror FPS
- Developer：　Monolith Production
- Publisher　：SIERRA
- Hardware: Windows
- Year：　2004

Agent Architecture Considerations for Real-Time Planning in Games (AIIDE 2005)
http://web.media.mit.edu/~jorkin/AIIDE05_Orkin_Planning.ppt

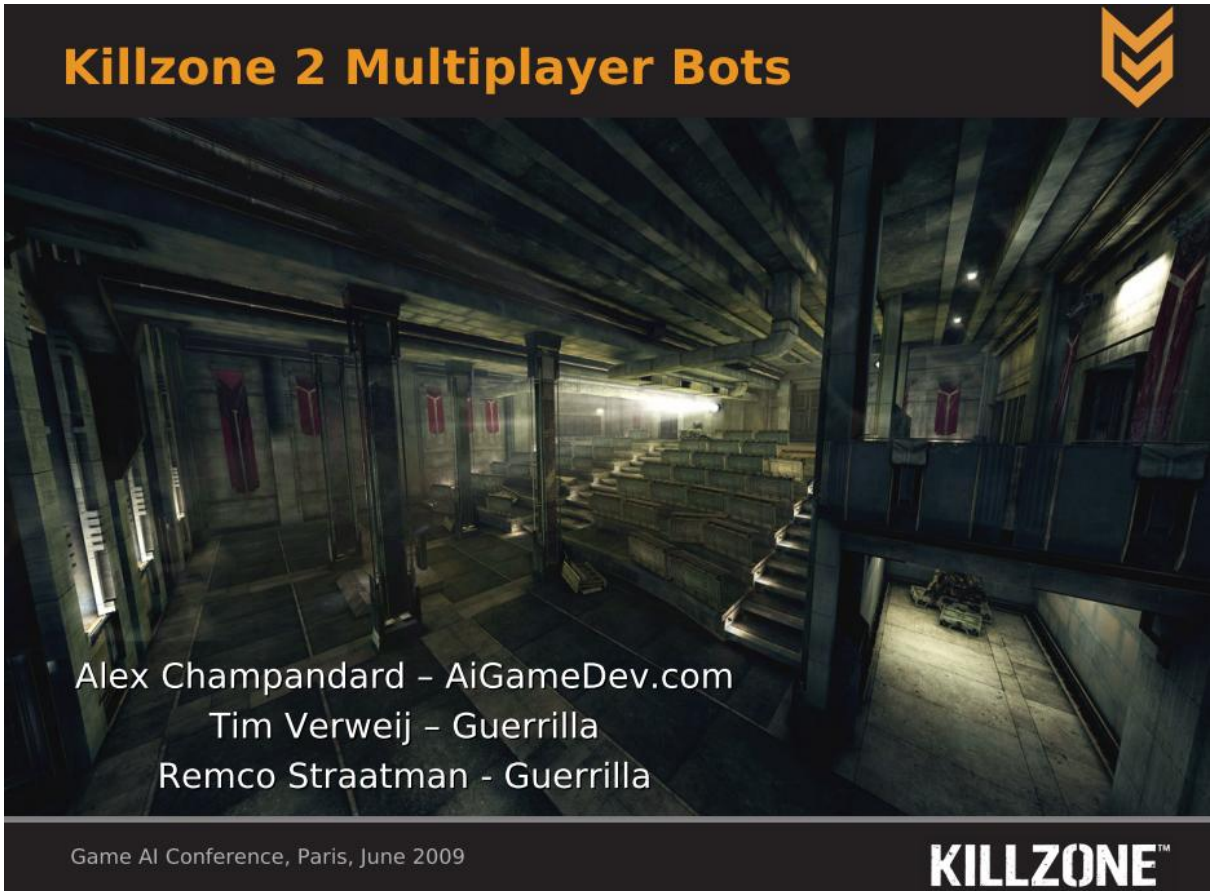
Jeff Orkin と僕 @グラナダ

Killzone 2 AI (マルチプレイヤーモード)

(1) Killzone 2 のAI思考

(2) Killzone 2 のマップ自動解析

On the AI Strategy for KILLZONE 2's Multiplayer Bots
http://aigamedev.com/open/coverage/killzone2/



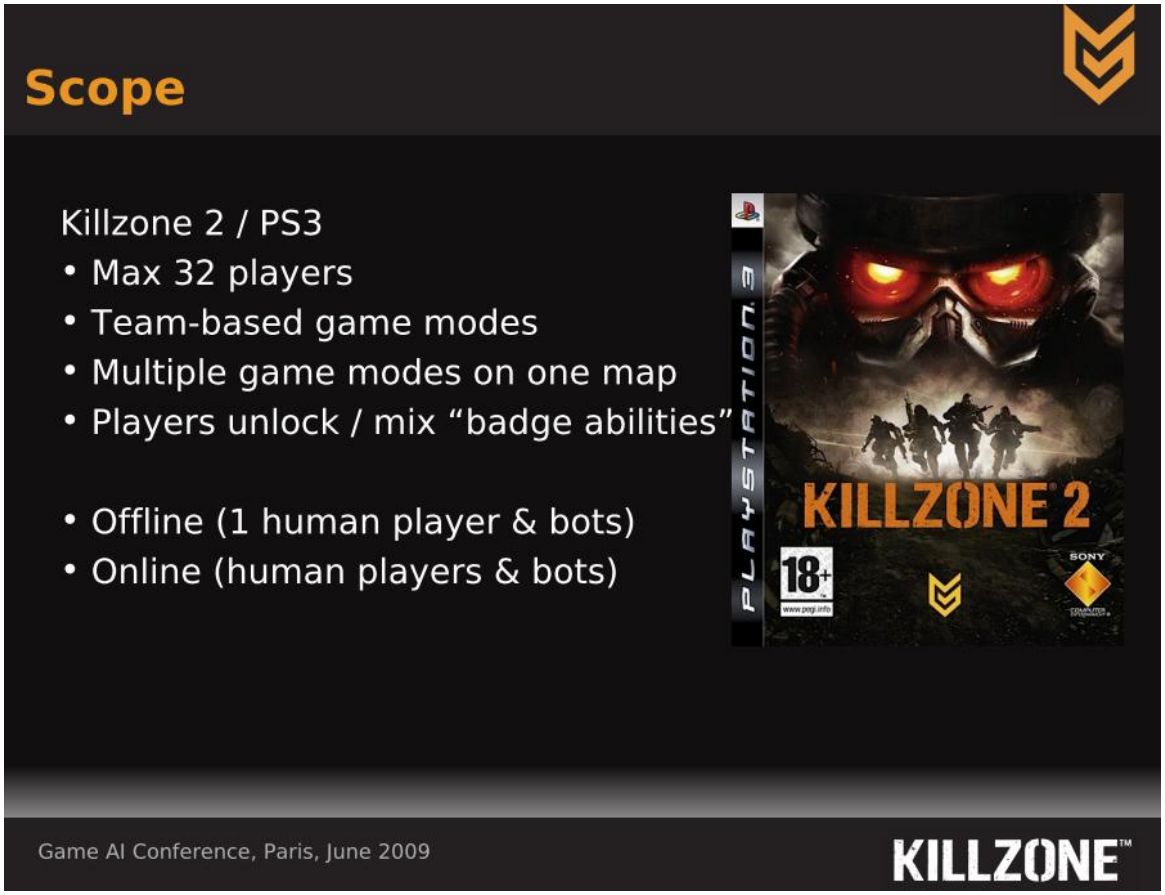
Killzone 2 Screen

On the AI Strategy for KILLZONE 2's Multiplayer Bots
http://aigamedev.com/open/coverage/killzone2/

Killzone 2 AI (マルチプレイヤーモード)

(1) Killzone 2 のAI思考

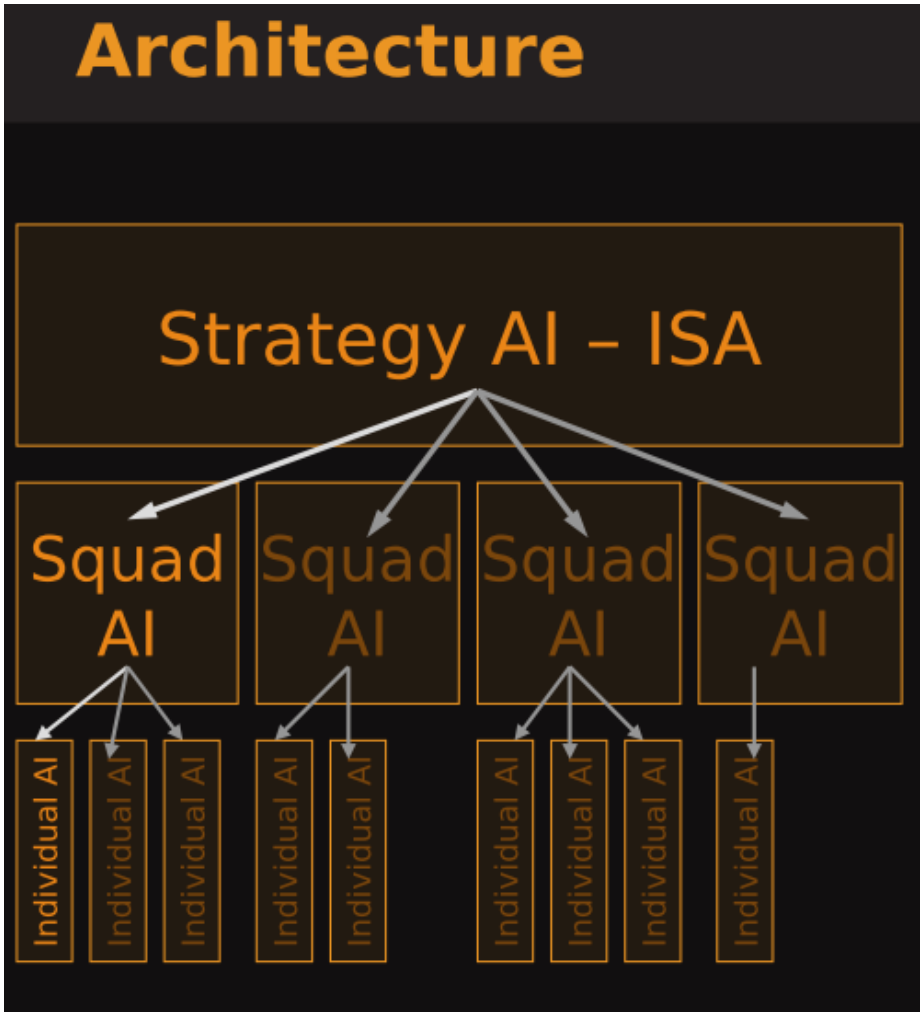
(2) Killzone 2 のマップ自動解析



On the AI Strategy for KILLZONE 2’s Multiplayer Bots
http://aigamedev.com/open/coverage/killzone2/

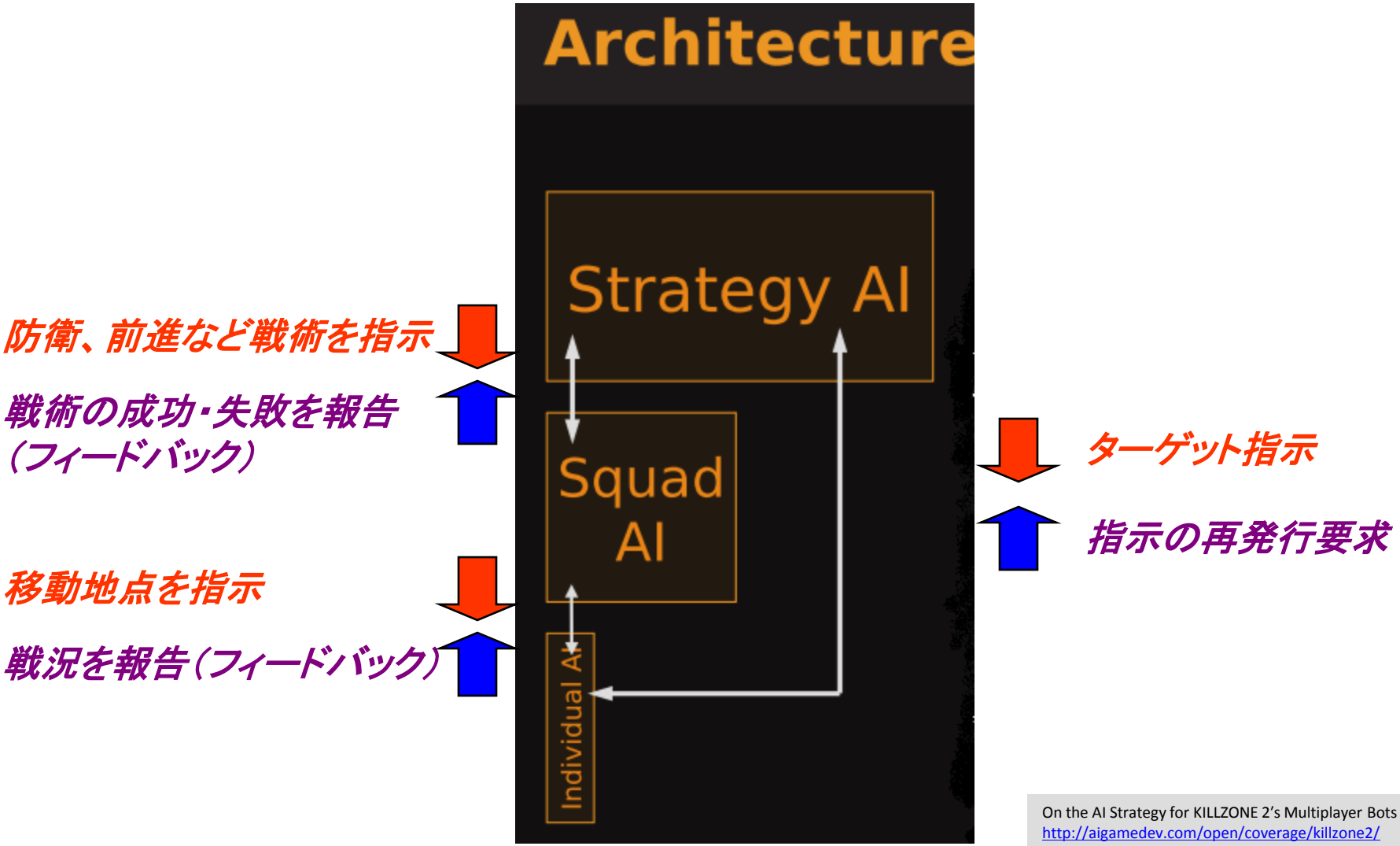
Killzone2 AI

集団のAIのコントロールの仕方を考える



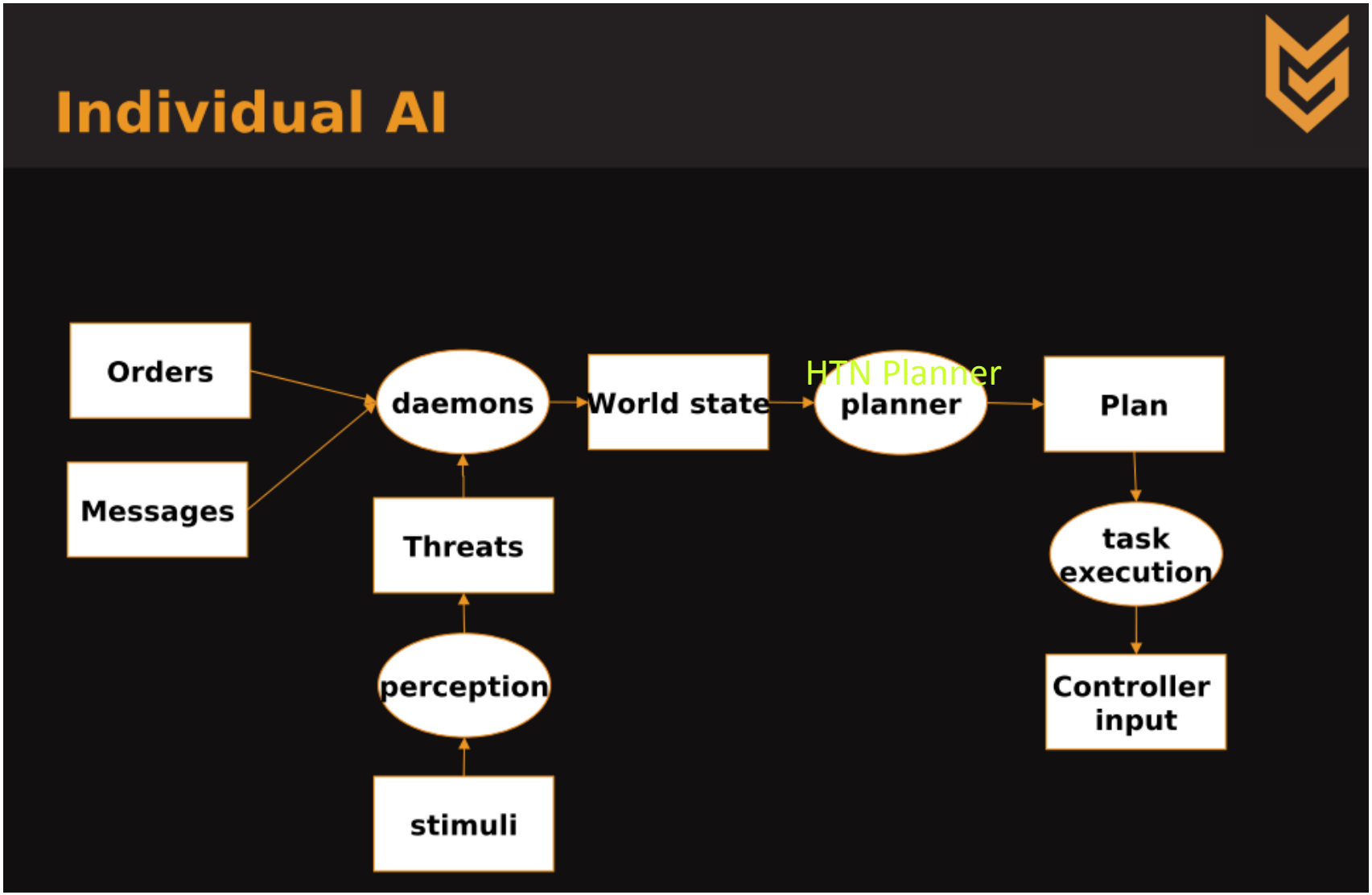
On the AI Strategy for KILLZONE 2's Multiplayer Bots
http://aigamedev.com/open/coverage/killzone2/

Killzone2 AI



On the AI Strategy for KILLZONE 2's Multiplayer Bots
http://aigamedev.com/open/coverage/killzone2/

各AIのアーキテクチャ



On the AI Strategy for KILLZONE 2's Multiplayer Bots
http://aigamedev.com/open/coverage/killzone2/

HTN (Hierarchical Task Network)

Individual AI : HTN Planner

HTN Planning: Complexity and Expressivity.
K. Erol, J. Hendler and D. Nau.
In Proc. AAAI-94.

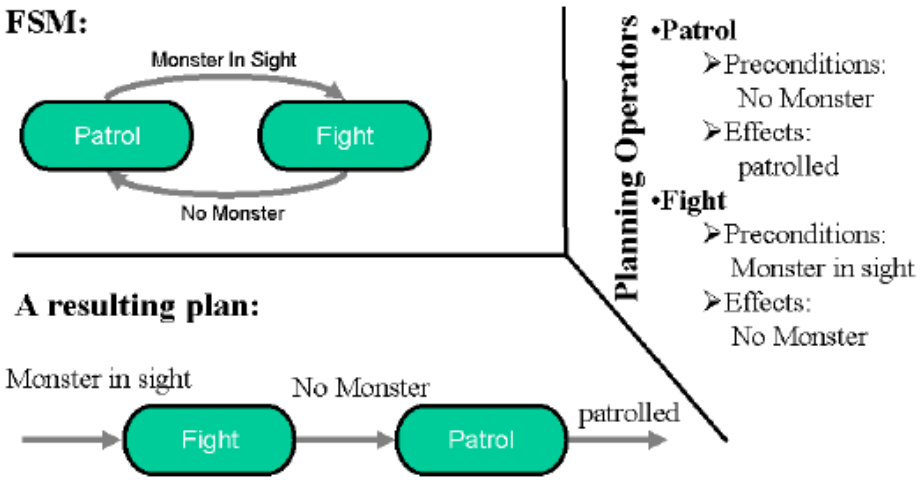


Figure 1. Contrasting FSMs and GOAP

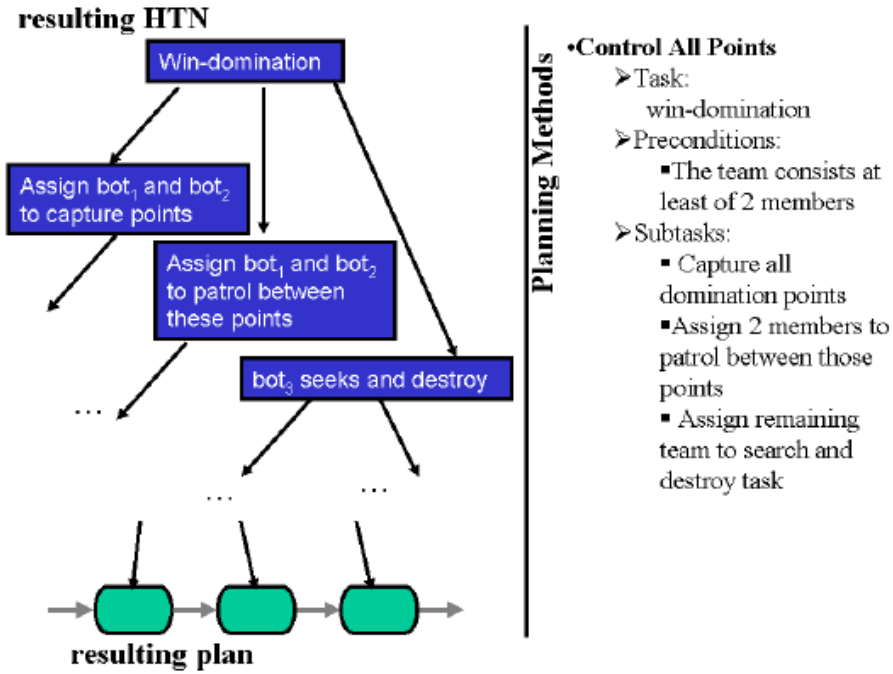
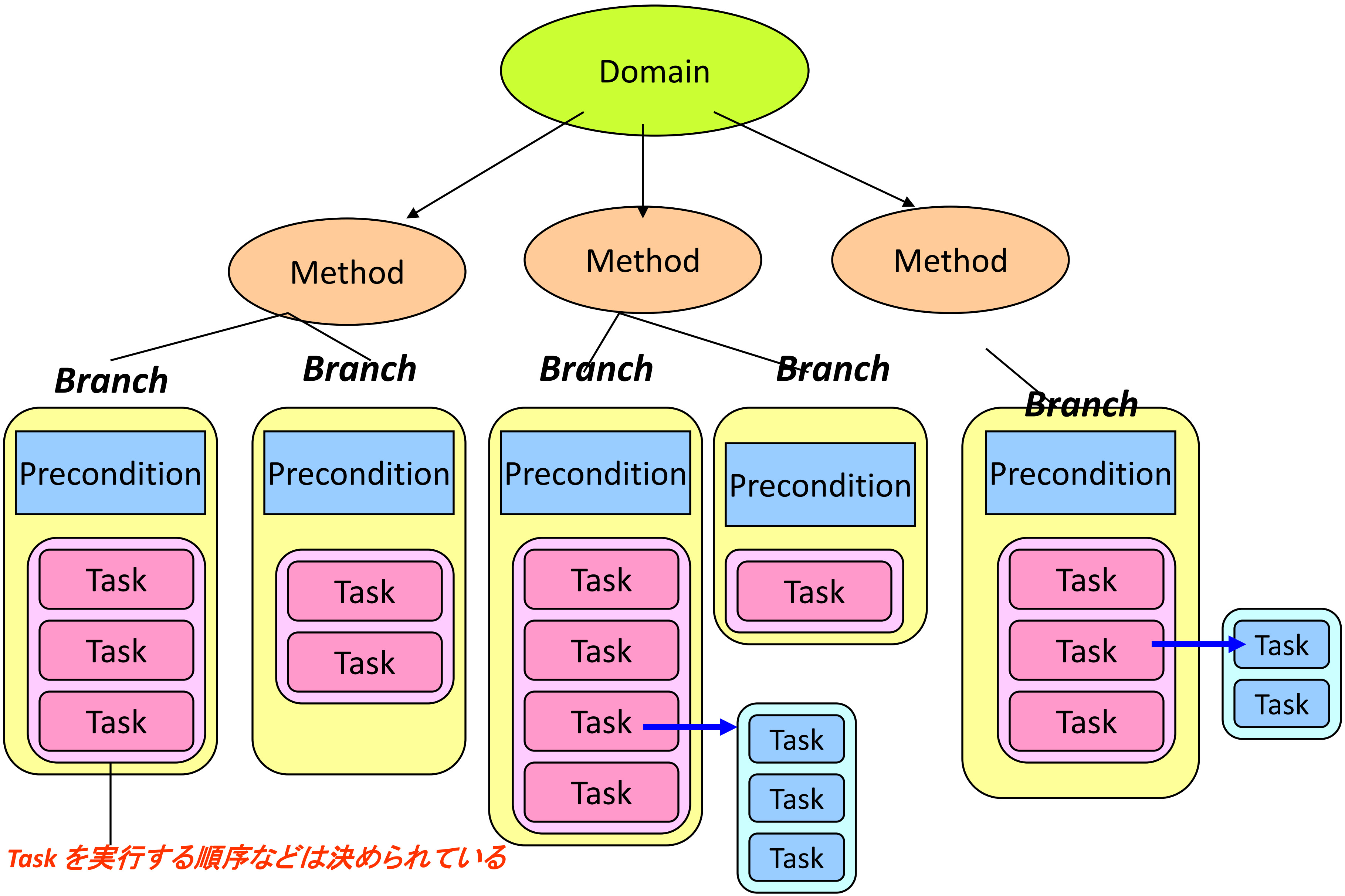


Figure 2. A Method and an HTN for UT bots

On the AI Strategy for KILLZONE 2's Multiplayer Bots
http://aigamedev.com/open/coverage/killzone2/

HTN (Hierarchical Task Network)



HTN (Hierarchical Task Network)

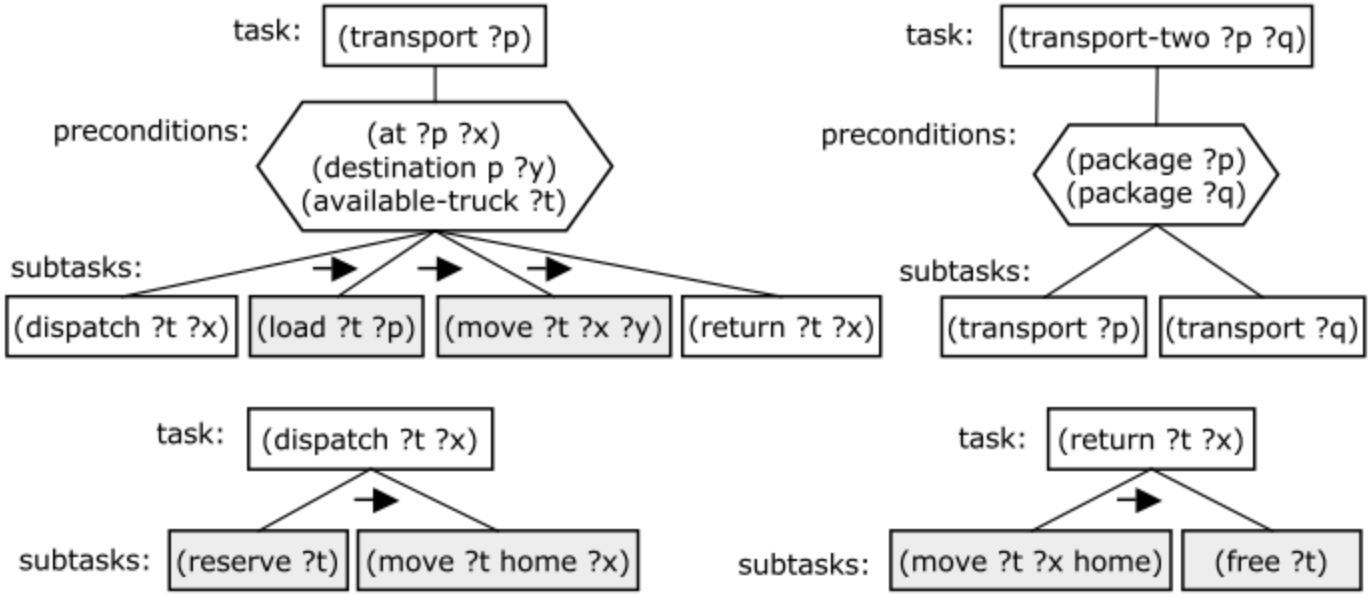
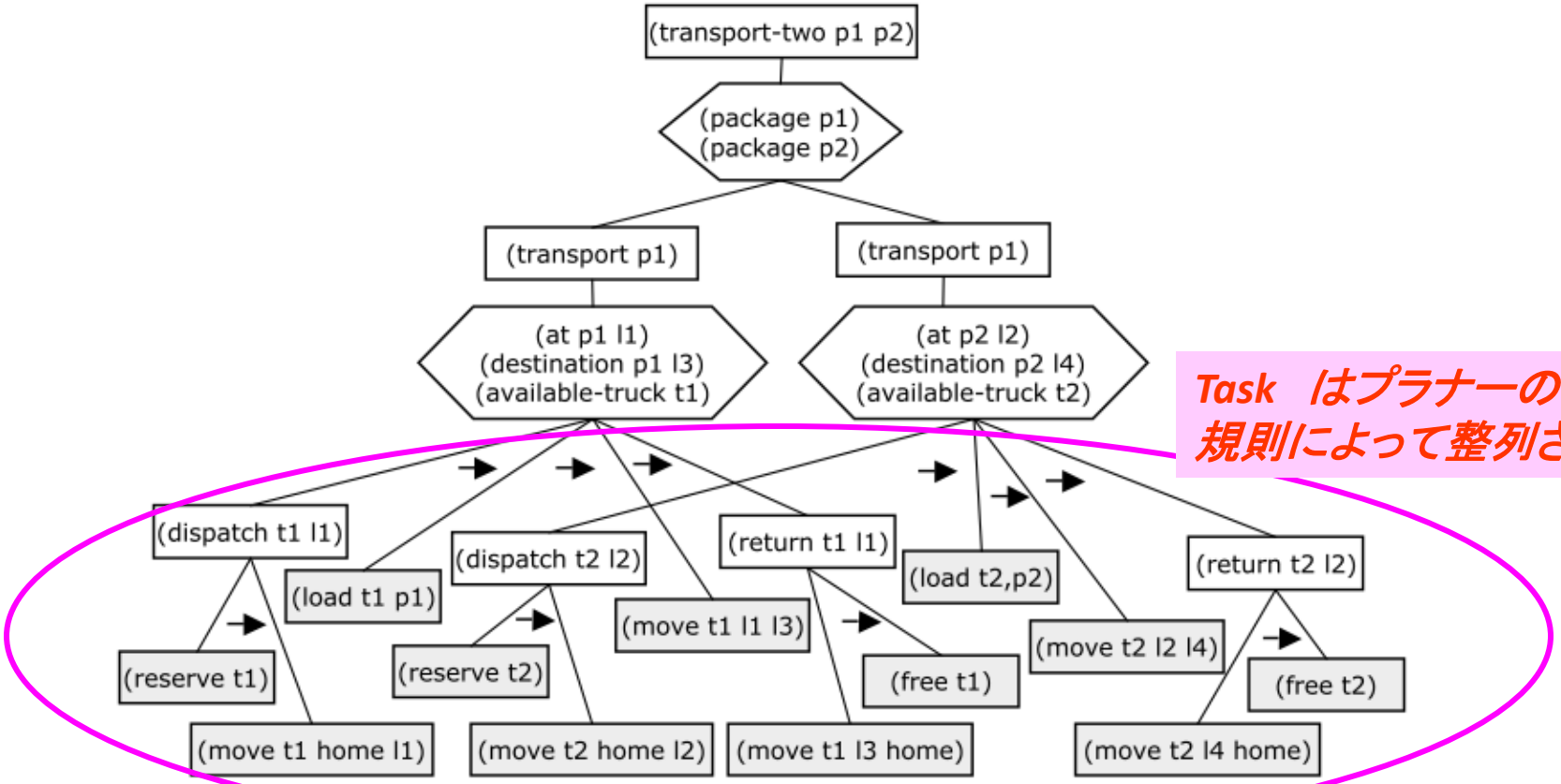


Figure 1: Methods for transporting a package ?p, transporting two packages ?p and ?q, dispatching a truck ?t, and returning the truck. Arrows are ordering constraints. The shaded subtasks are *primitive* tasks that are accomplished by the following planning operators: (load ?t ?p) loads ?p onto ?t; (move ?t ?x ?y) moves ?t from ?x to ?y; (reserve ?t) deletes (available-truck ?t) to signal that the truck is in use; (free ?t) adds (available-truck ?t) to signal that the truck is no longer in use.

Dana Nau et al., "SHOP2: An HTN Planning System",
Journal of Artificial Intelligence Research 20 (2003) 379-404

HTN (Hierarchical Task Network)



Task はプラナーの規則によって整列される

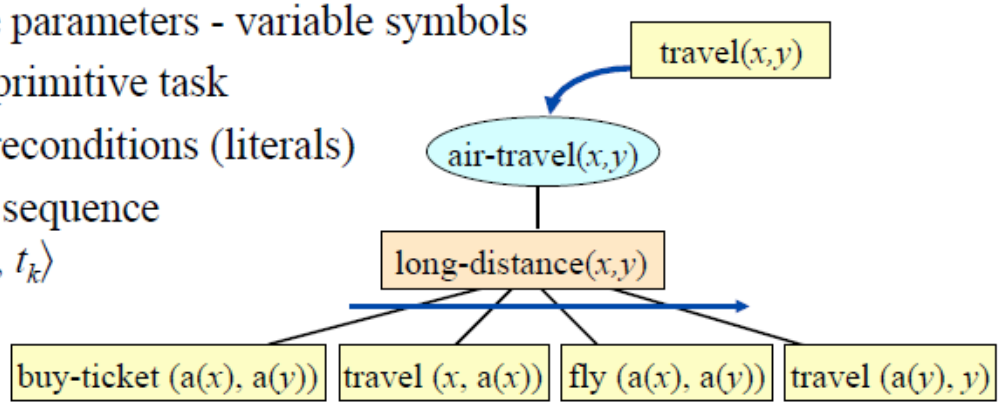
Figure 2: A plan for accomplishing (transport-two p1 p2) from the following initial state: {(package p1), (at p1 l1), (destination p1 l3), (available-truck t1), (at t1 home), (package p2), (at p2 l2), (destination p2 l4), (available-truck t2), (at t2 home)}.

Dana Nau et al., "SHOP2: An HTN Planning System", Journal of Artificial Intelligence Research 20 (2003) 379-404

HTN (Hierarchical Task Network)

Methods

- Totally ordered method: a 4-tuple
  m = (name(m), task(m), precond(m), subtasks(m))
  - name(m): an expression of the form $n(x_1,\ldots,x_n)$
    - » $x_1,\ldots,x_n$ are parameters - variable symbols
  - task(m): a nonprimitive task
  - precond(m): preconditions (literals)
  - subtasks(m): a sequence of tasks $\langle t_1, \ldots, t_k \rangle$



air-travel(x,y)

task: travel(x,y)

precond: long-distance(x,y)

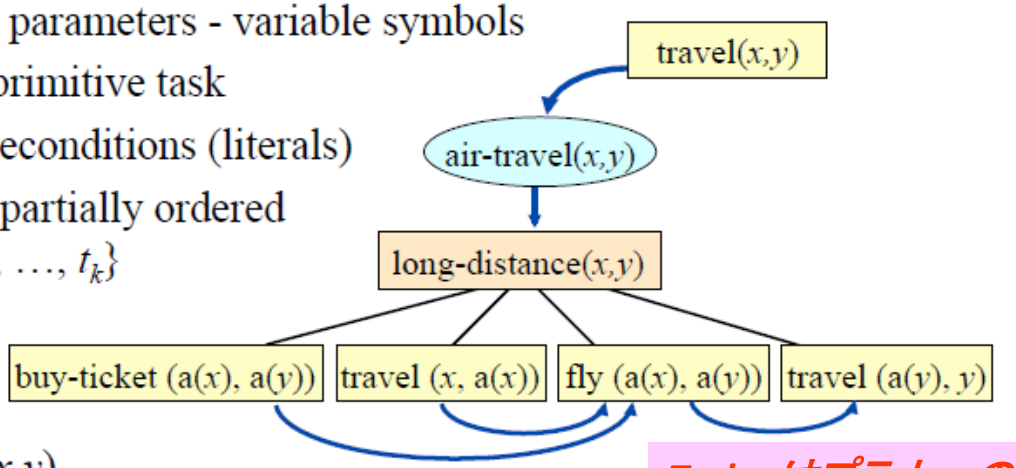
subtasks: ⟨buy-ticket($a(x)$, $a(y)$), travel(x,$a(x)$), fly($a(x)$, $a(y)$), travel($a(y)$,y)⟩

Dana S. Nau,"Hierarchical Task Network Planning", Lecture slides for Automated Planning: Theory and Practice http://www.cs.umd.edu/~nau/cmsc722/notes/chapter11.pdf

HTN (Hierarchical Task Network)

Methods (Continued)

- Partially ordered method: a 4-tuple
  m = (name(m), task(m), precond(m), subtasks(m))
  - name(m): an expression of the form $n(x_1,\ldots,x_n)$
    - » $x_1,\ldots,x_n$ are parameters - variable symbols
  - task(m): a nonprimitive task
  - precond(m): preconditions (literals)
  - subtasks(m): a partially ordered set of tasks $\{t_1, \ldots, t_k\}$



air-travel(x,y)

task: travel(x,y)

precond: long-distance(x,y)

network: u_1=buy-ticket($a(x)$,$a(y)$), u_2= travel(x,$a(x)$), u_3= fly($a(x)$, $a(y)$)
u_4= travel($a(y)$,y), $\{(u_1,u_3), (u_2,u_3), (u_3,u_4)\}$

Task はプラナーの規則によって整列される

Dana S. Nau,"Hierarchical Task Network Planning", Lecture slides for Automated Planning: Theory and Practice http://www.cs.umd.edu/~nau/cmsc722/notes/chapter11.pdf

今日の内容

第一章

AIはどのように世界を結びつくか？

第二章

エージェントモデル

第三章

行動表現と世界表現の双対性

第三章
行動表現と世界表現の双対性

世界の表現の双対性

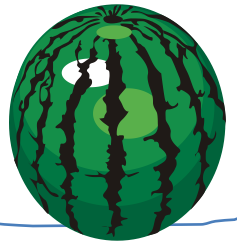
- アフォーダンス ＝ 環境（物）の側に情報を埋め込む。
- 行動表現　　＝ 行動の側に情報を埋め込む。

エージェント

知識表現

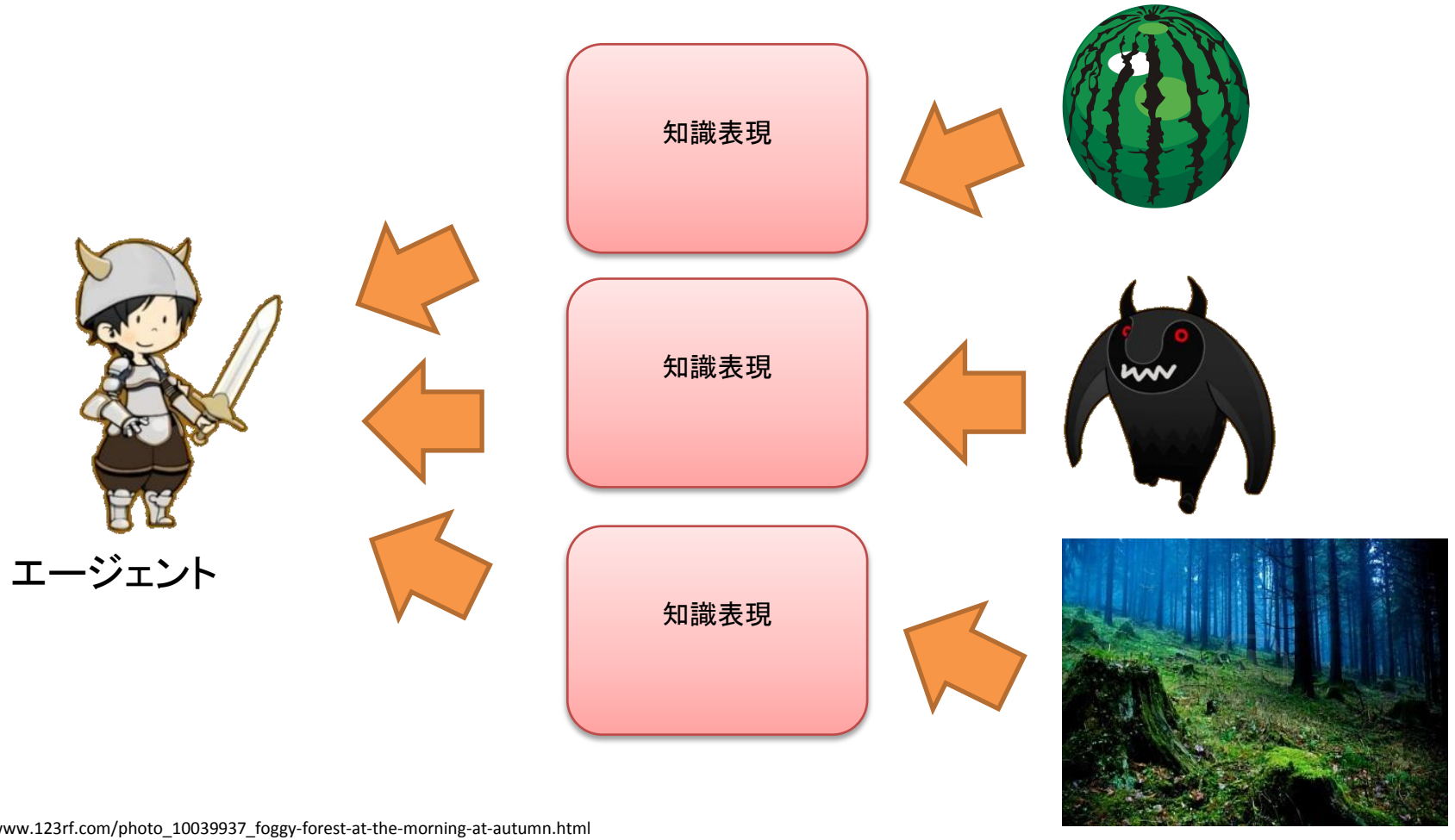
静的属性
-緑と黒のしましま。
- 丸い。
- つやつや。
- おいしそう。

アフォーダンス
-転がすことができる。
- 投げることができる。
-食べることができる。



世界の表現の双対性

- アフォーダンス ＝ 環境（物）の側に情報を埋め込む。
- 行動表現　　＝ 行動の側に情報を埋め込む。



http://www.123rf.com/photo_10039937_foggy-forest-at-the-morning-at-autumn.html

世界の表現の双対性

- アフォーダンス ＝ 環境（物）の側に情報を埋め込む。
- 行動表現　　＝ 行動の側に情報を埋め込む。

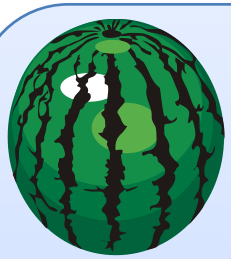
エージェント

知識表現

AIから見えている世界

知識表現

知識表現



世界そのもの

http://www.123rf.com/photo_10039937_foggy-forest-at-the-morning-at-autumn.html

世界の表現の双対性

- アフォーダンス ＝ 環境（物）の側に情報を埋め込む。
- 行動表現　　＝ 行動の側に情報を埋め込む。

エージェント

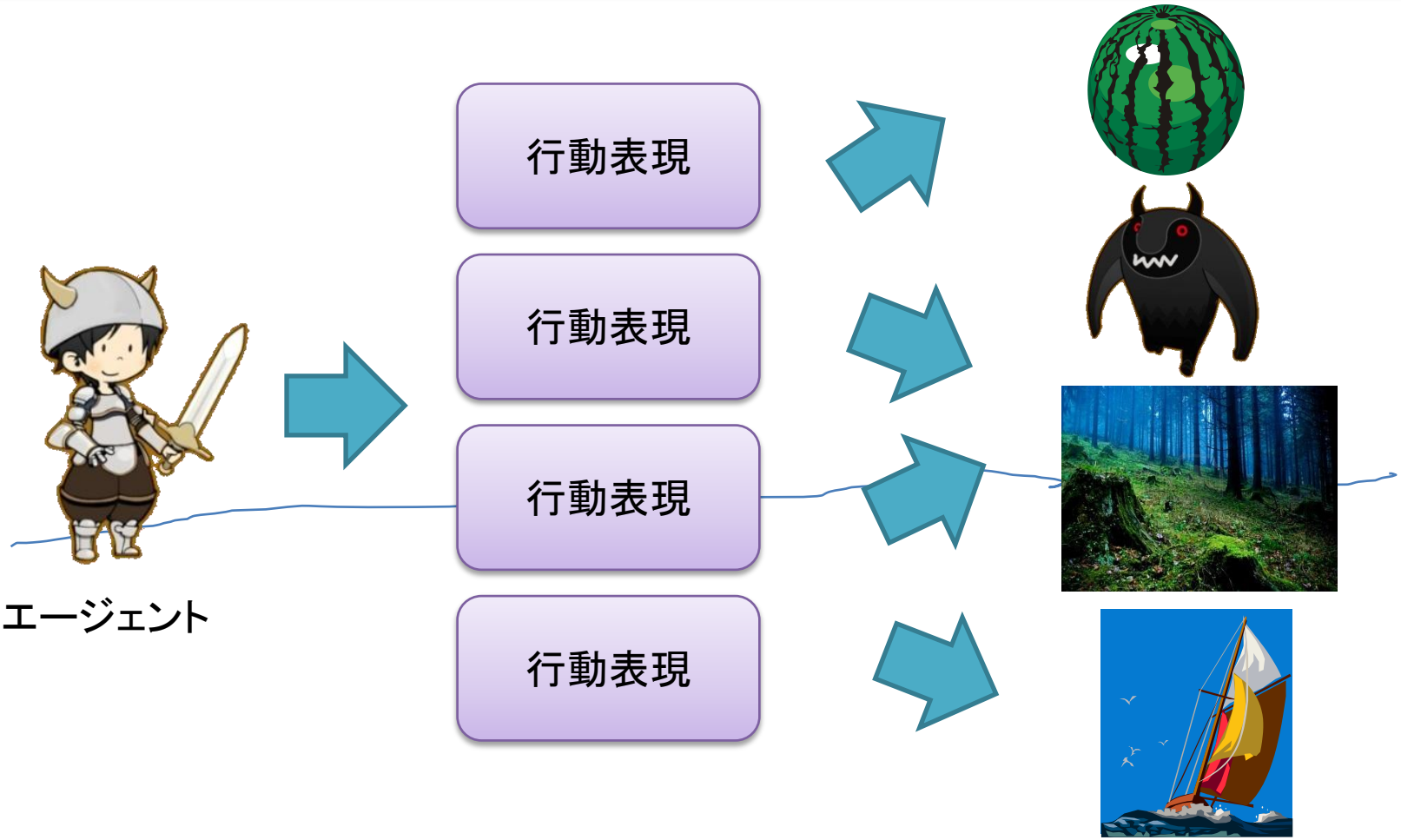
行動表現

行動条件
- 行動開始条件
- 性質
- 時間
- アニメーション

行動表現
- 投げる。

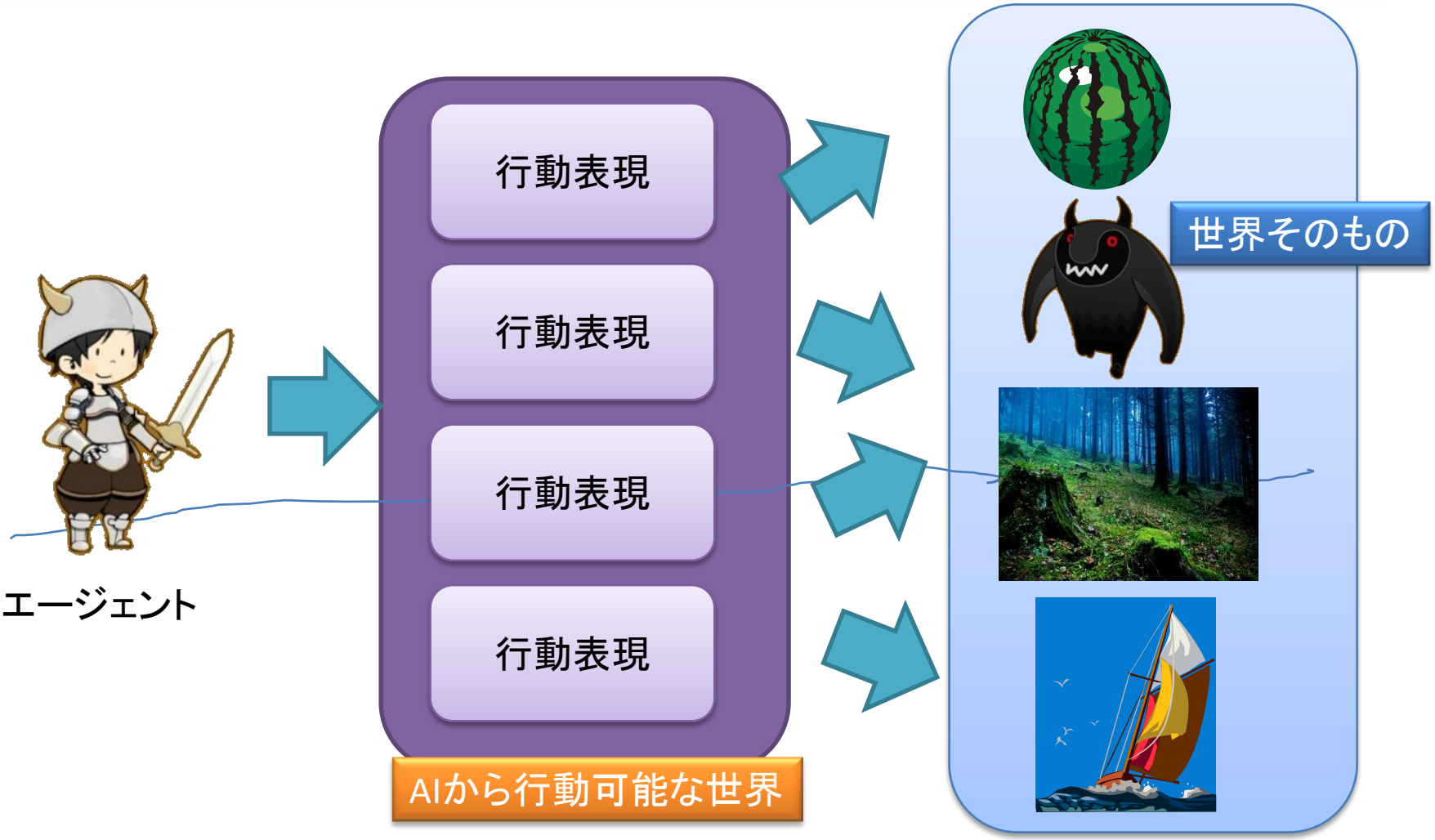
世界の表現の双対性

- アフォーダンス ＝ 環境（物）の側に情報を埋め込む。
- 行動表現　　＝ 行動の側に情報を埋め込む。



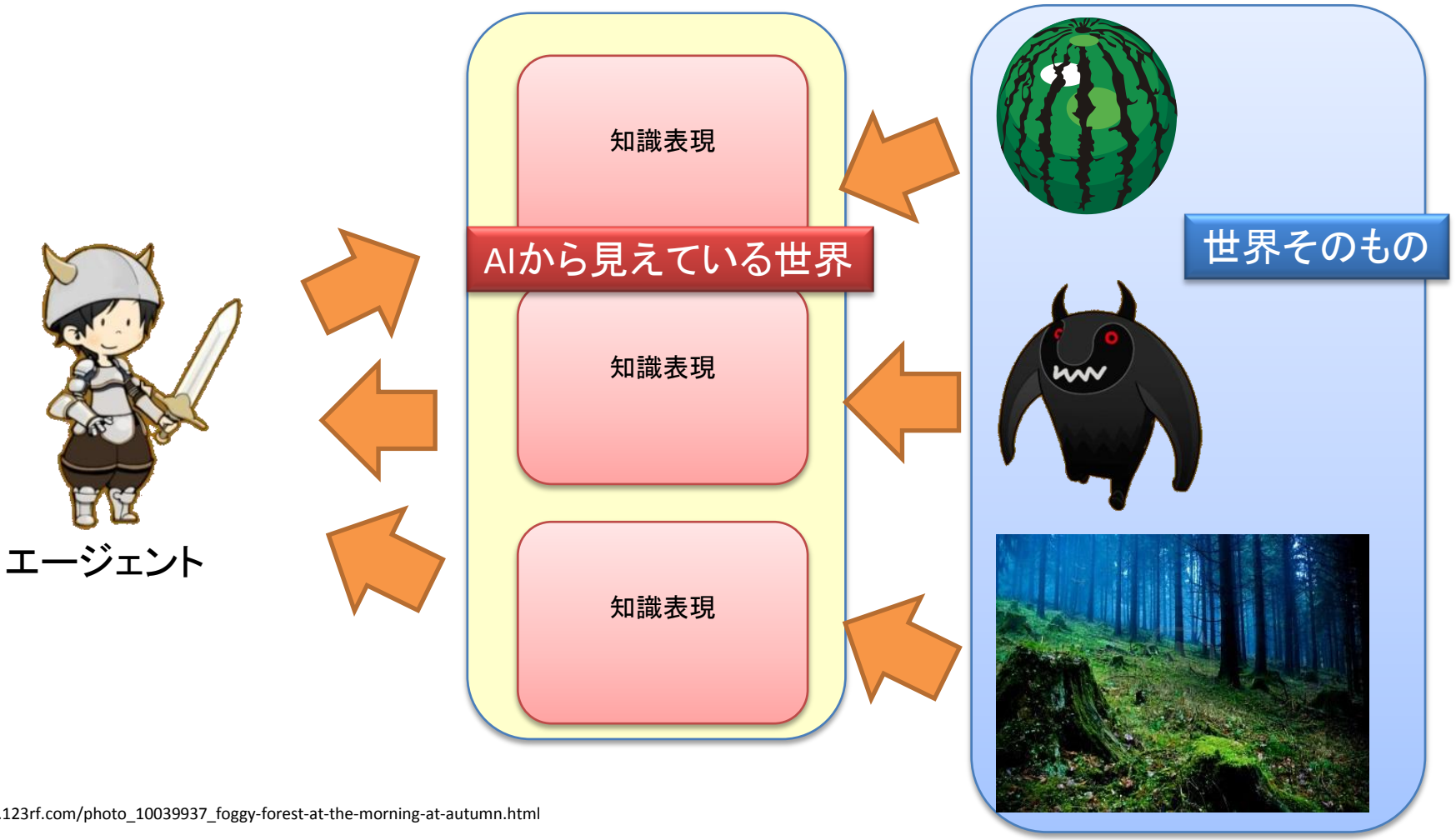
世界の表現の双対性

- アフォーダンス ＝ 環境（物）の側に情報を埋め込む。
- 行動表現　　＝ 行動の側に情報を埋め込む。



世界の表現の双対性

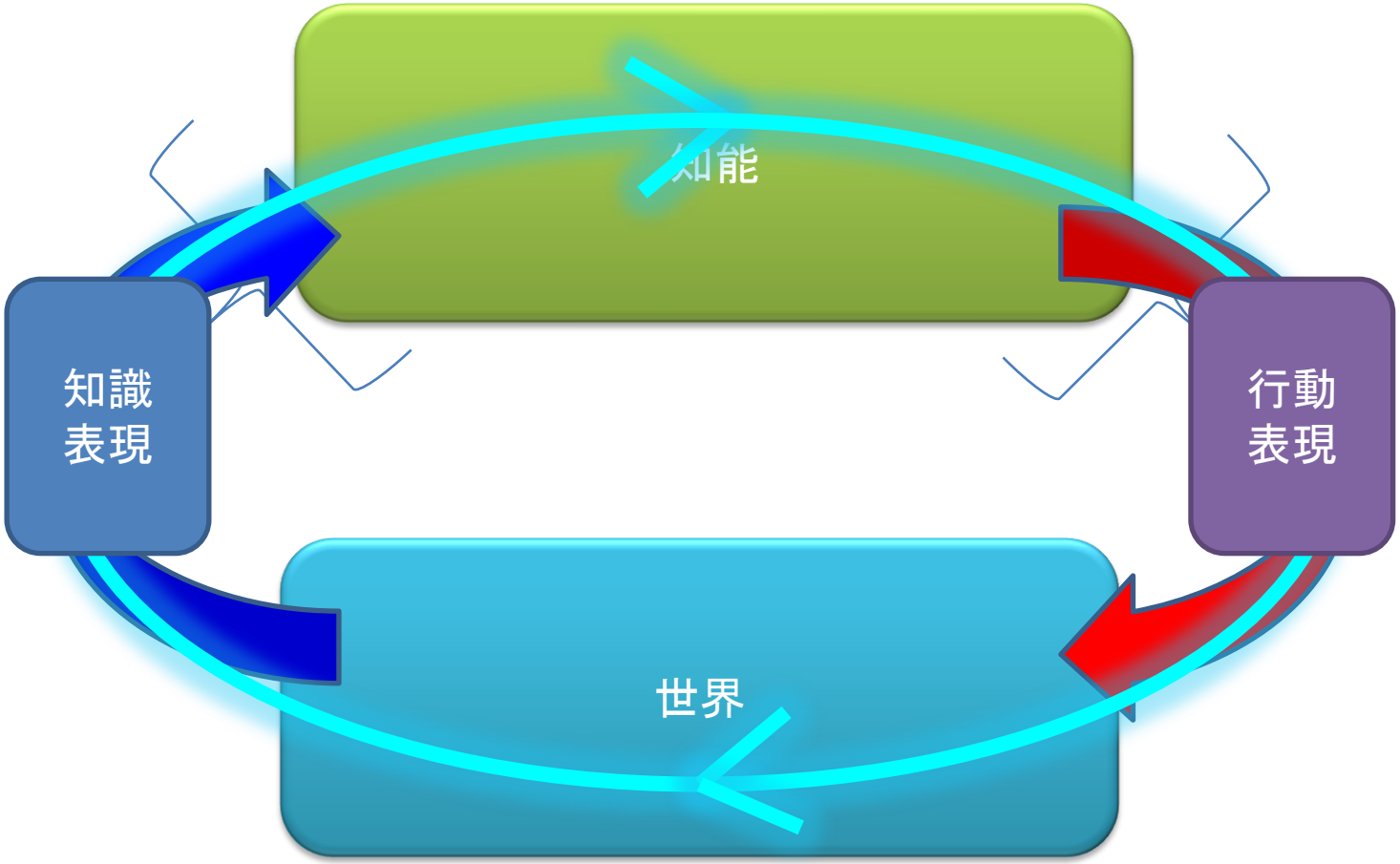
- アフォーダンス ＝ 環境（物）の側に情報を埋め込む。
- 行動表現　　＝ 行動の側に情報を埋め込む。



http://www.123rf.com/photo_10039937_foggy-forest-at-the-morning-at-autumn.html

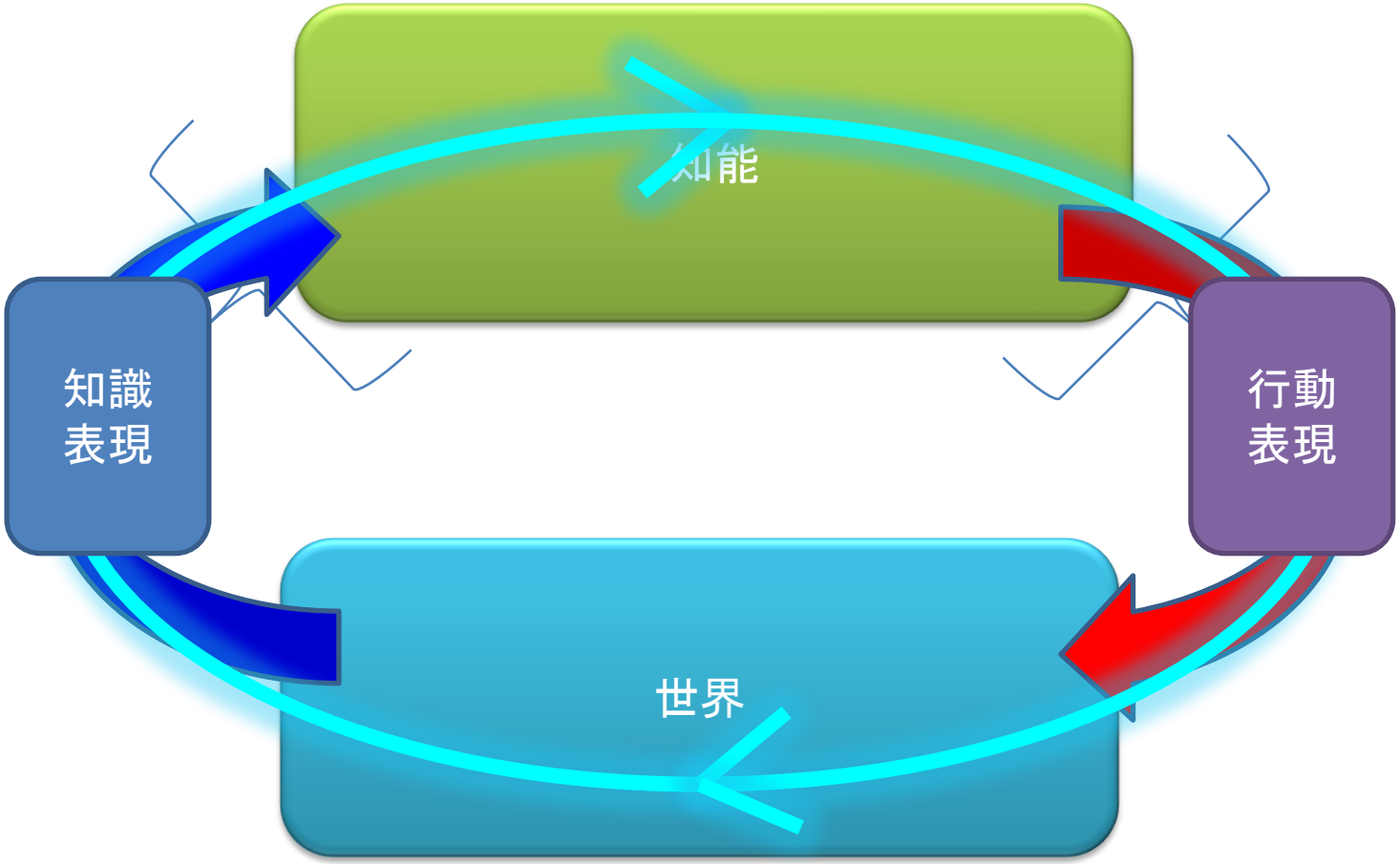
運動と知覚

- どのように運動が知覚と世界を形成していくか。



運動と知覚

知識表現は環境・オブジェクトの中に行動の情報を持ち、
行動表現はアクションの中に世界の情報を持つ。



環境に対立するものとしての知性

- 知性は独立した自律系
- 環境を制覇するためにある。

環境の中の知性という考え方

•生物は環境の中から生まれて来た、ということは、生物は環境の一部として運動している。
•大局的な複雑系（＝環境）の中の局所的適応系（＝知性）として捉える。
•アージ理論

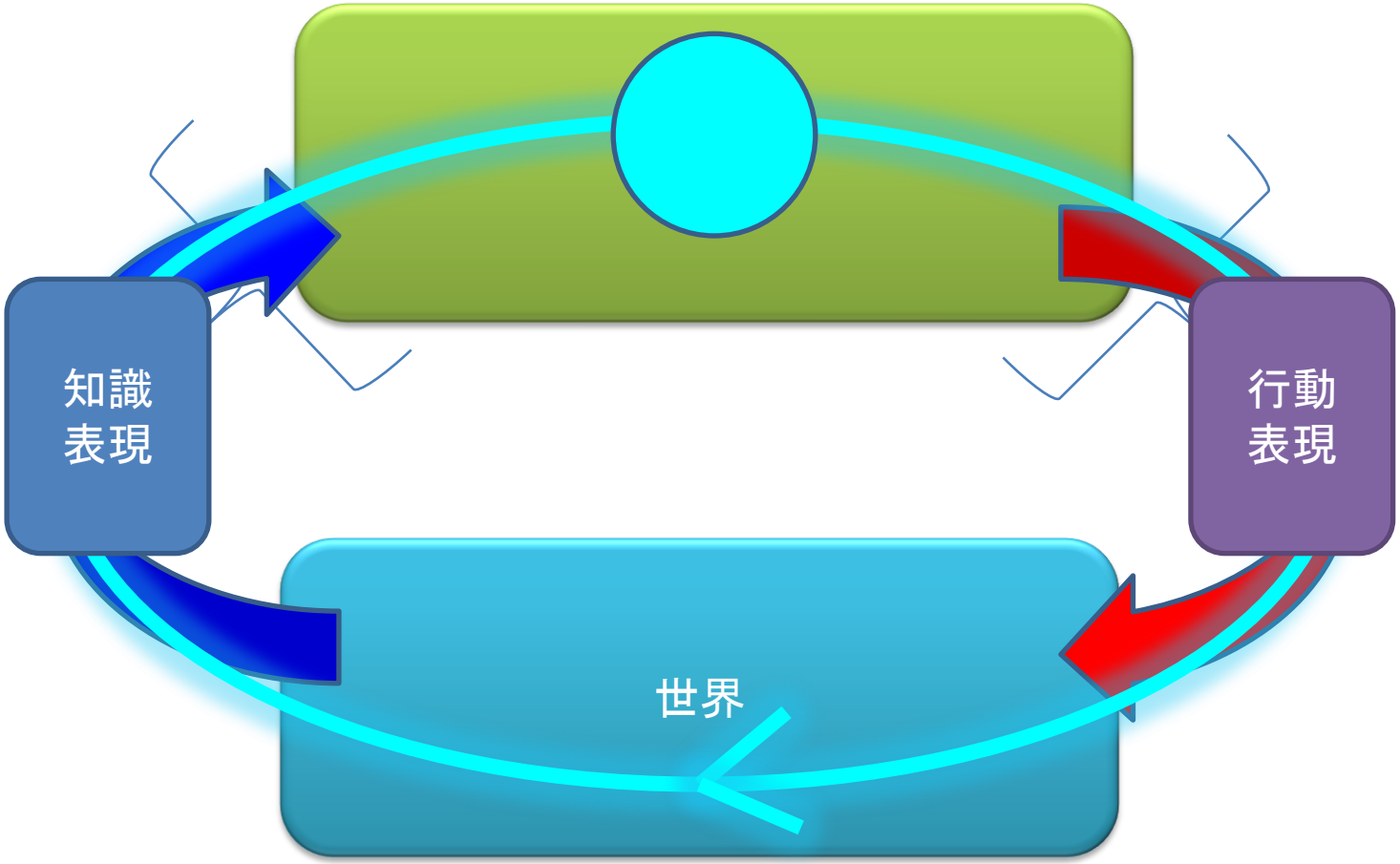
二つの情報の起源

行為　：　知性 → 世界 → 知性
（行きっぱなしの行為はない）

世界　：　世界 → 知性 → 世界
（世界に還元されない判断はない）

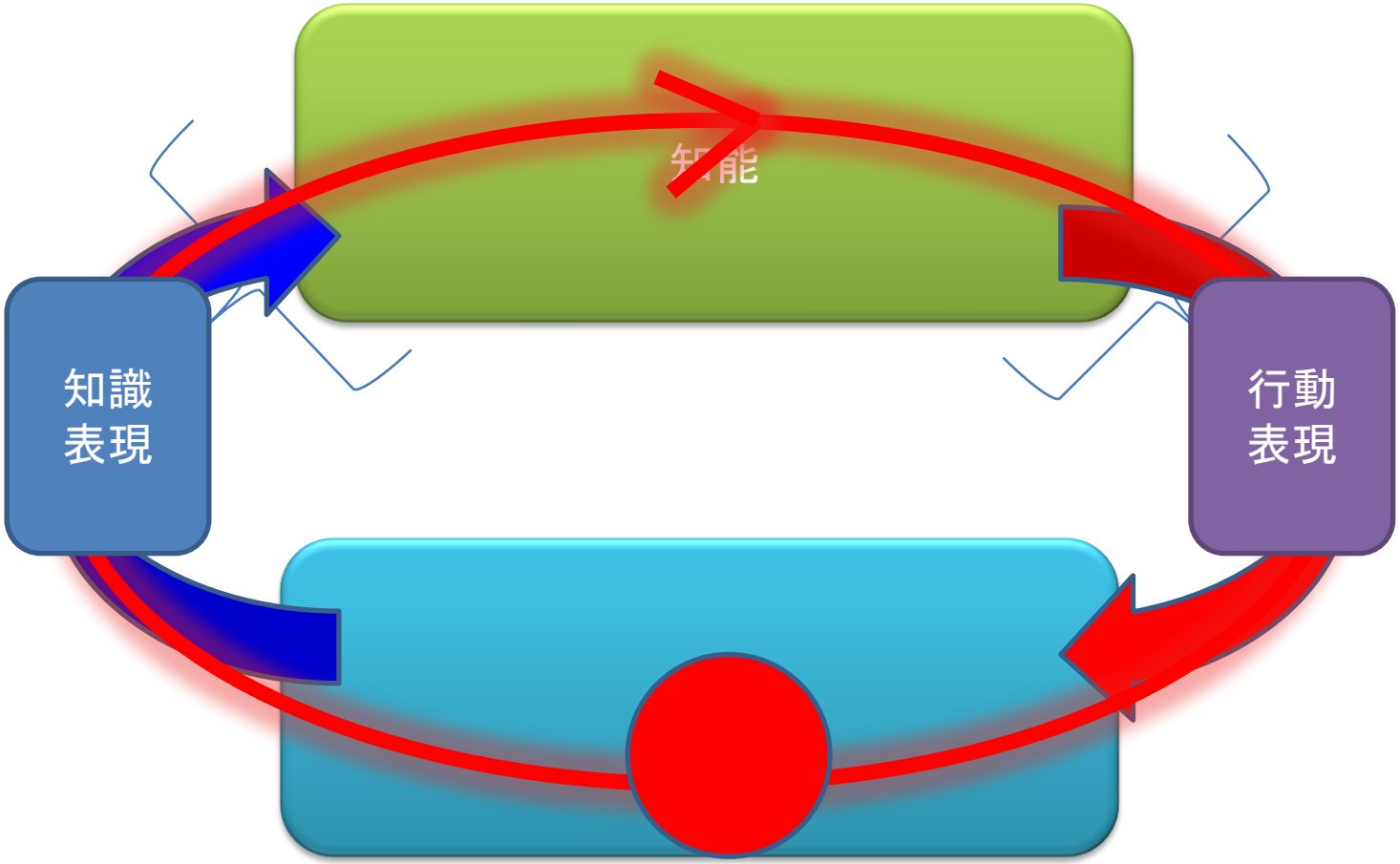
知能を起点に知性を作る

知識表現は環境・オブジェクトの中に行動の情報を持ち、
行動表現はアクションの中に世界の情報を持つ。



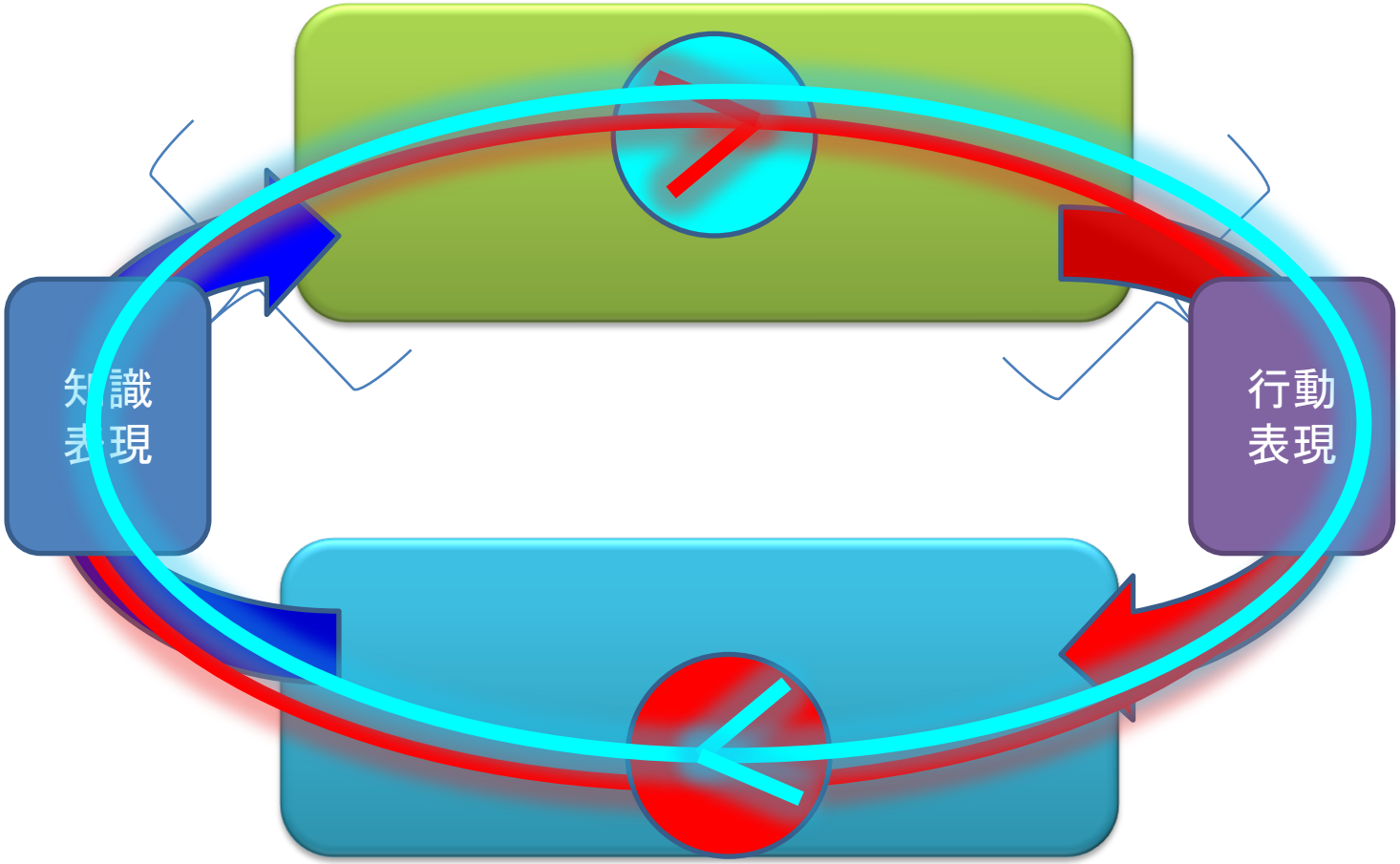
世界を起点に知能を作る

知識表現は環境・オブジェクトの中に行動の情報を持ち、
行動表現はアクションの中に世界の情報を持つ。



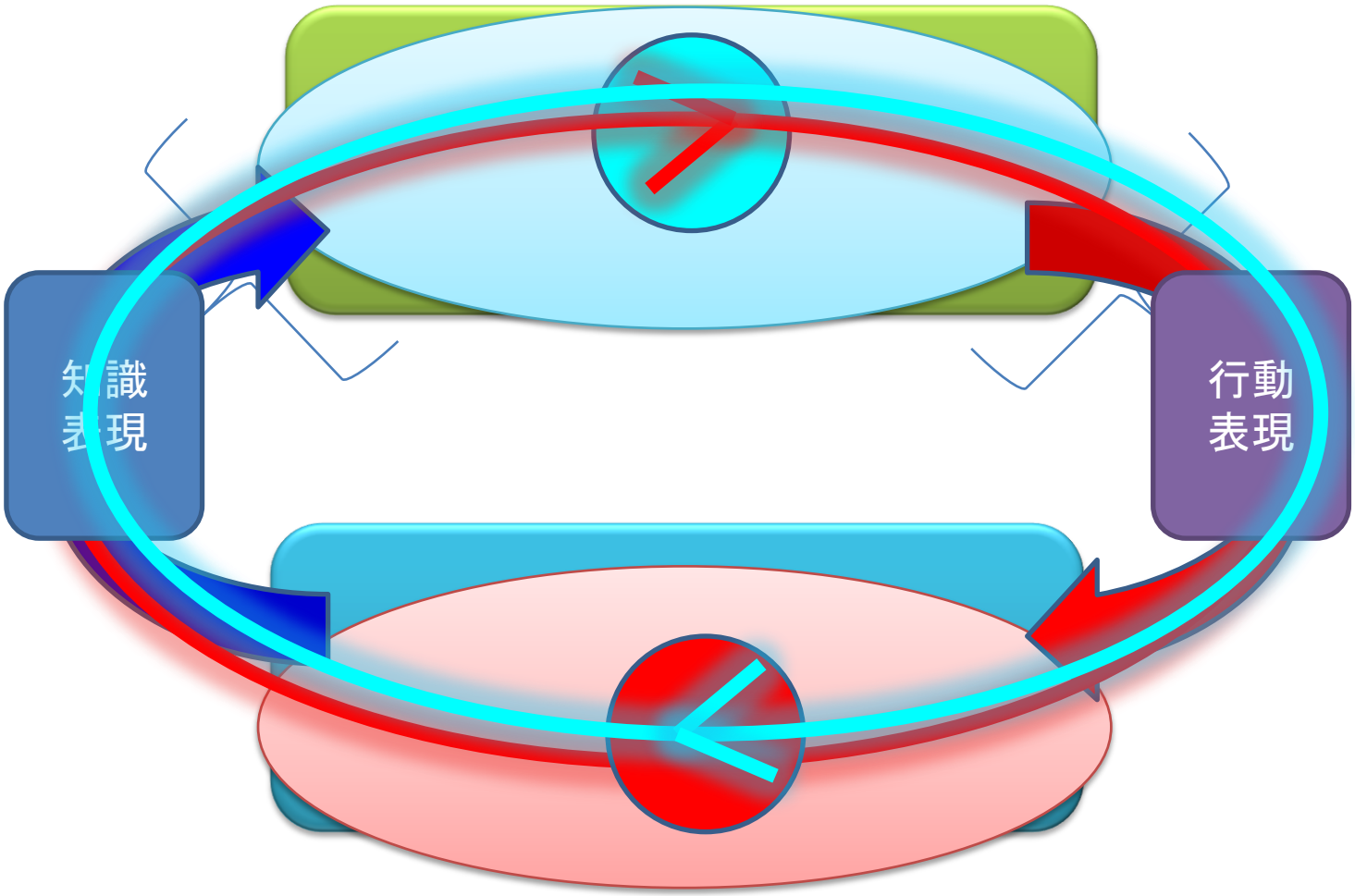
世界を起点に知能を作る

知識表現は環境・オブジェクトの中に行動の情報を持ち、
行動表現はアクションの中に世界の情報を持つ。



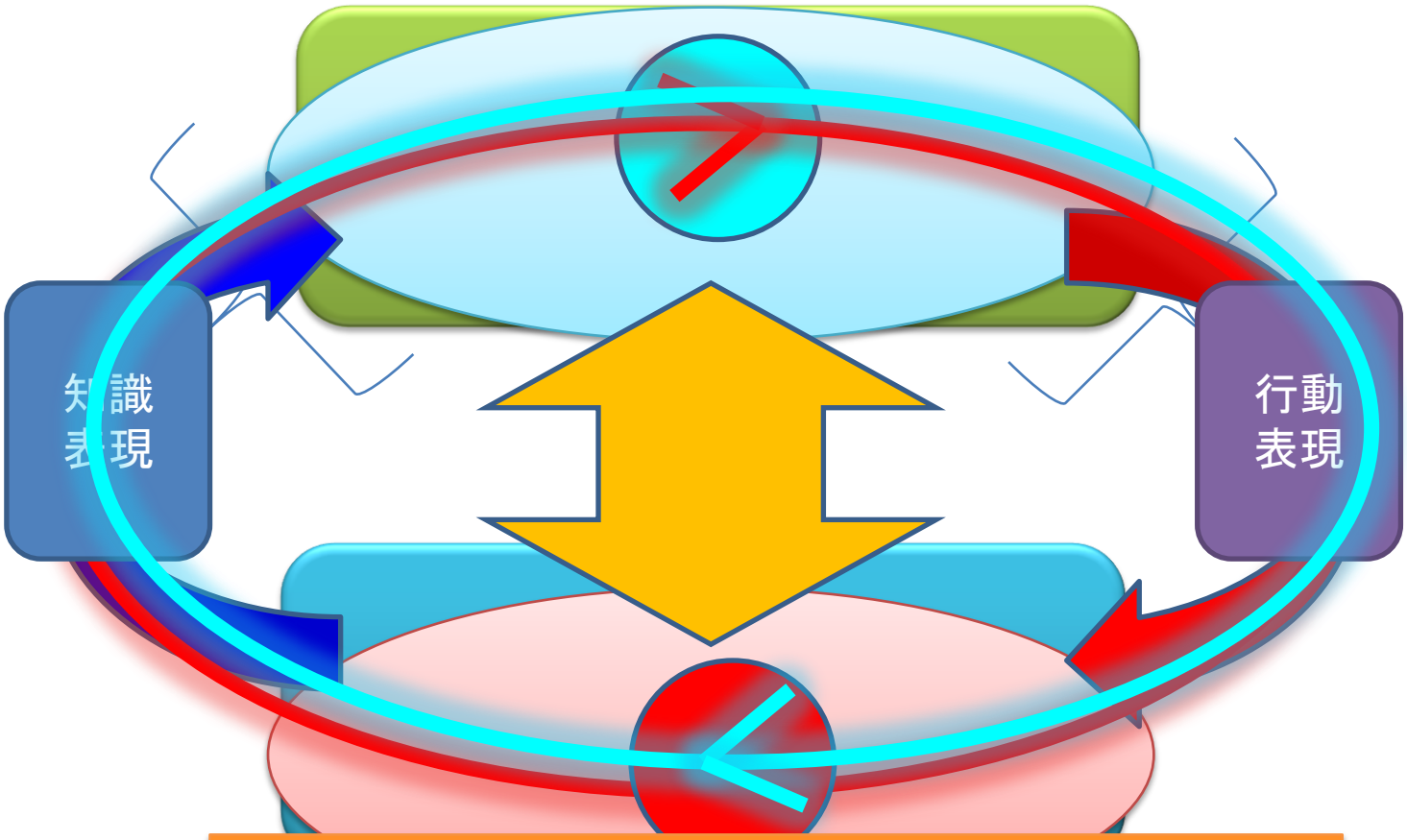
二つの極からせめぎ合う。

お互いがお互いの調和の中に引き込もうとする。



二つの極からせめぎ合う。

お互いがお互いの調和の中に引き込もうとする。



世界と知能のコミュニケーション

まとめ

第一章

人工知能と世界の結びつきは多様である。

第二章

エージェントアーキテクチャによる世界と人工知能の分離と新しい結びつき

第三章

行動表現と世界表現の対極からの結びつき