CITBrains Strategic Division（戦略班）

2018/09/01

Game AI Architecture Part 1
ゲームAIアーキテクチャー

スプラトマン　ジョシュア

今日のながれ

- Introduction
  - ゲームAIアーキテクチャは何だ？
- Knowledge Representation and Behavior:
  - 情報表現と行動とは？
- Behavior Selection
  - 行動決定のしかた、及び種類 (今日はFSMまで)

Introduction

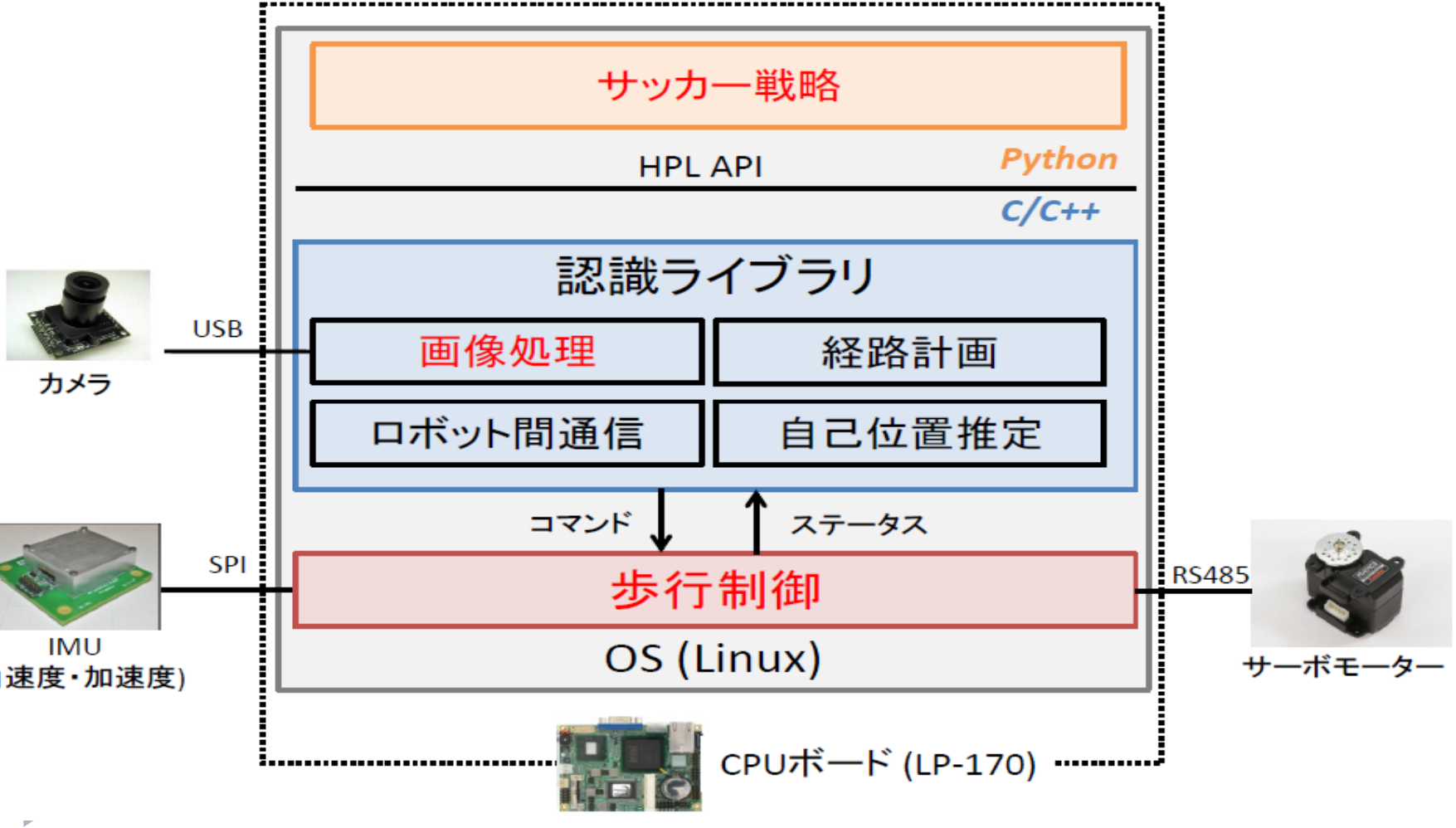
はじめに

戦略とは？

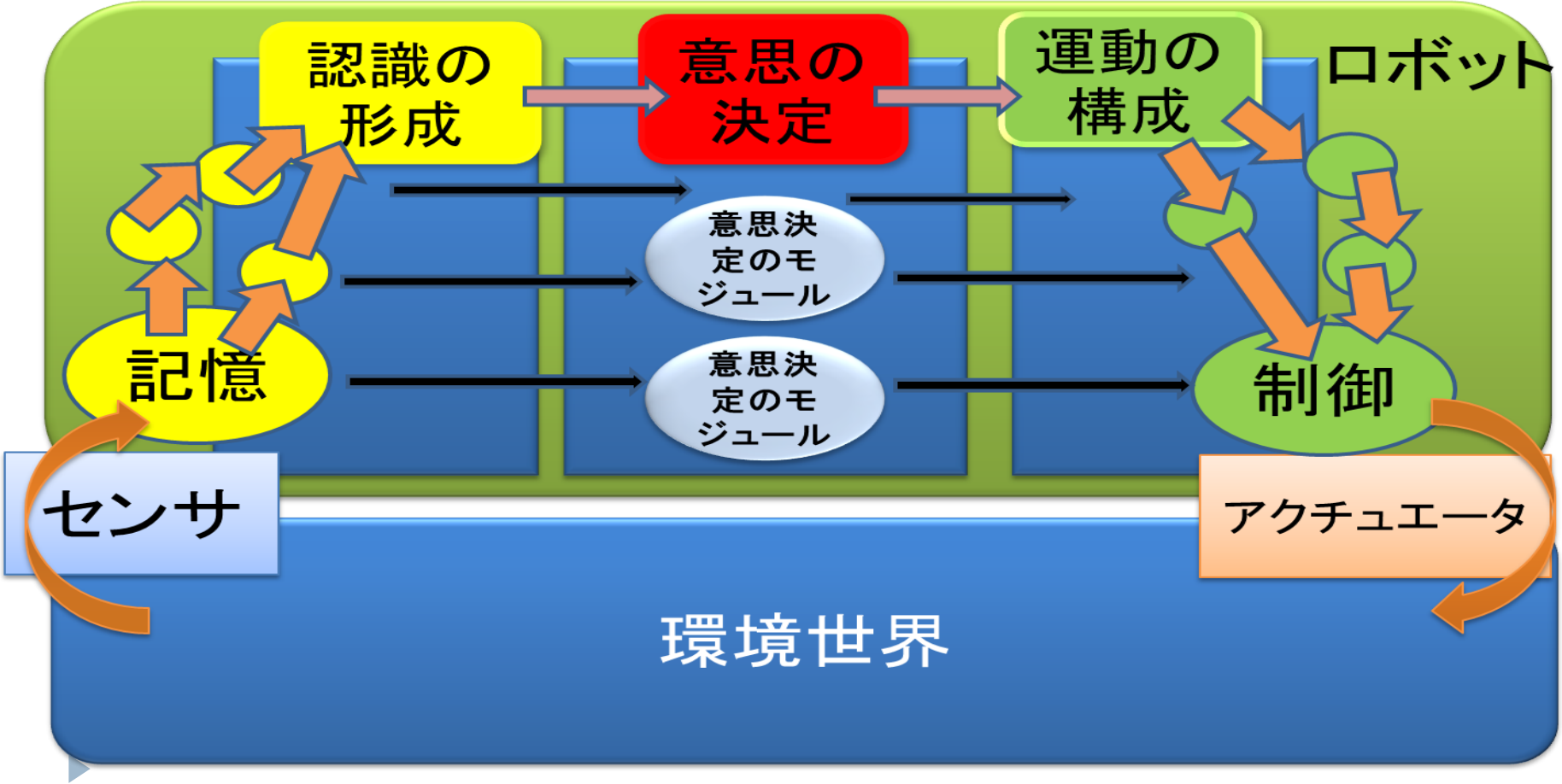
- 人工知能の一部
- 意思決定（なにをする？）
- プログラミングの構造で一番上の部分
- 簡単なものある
  - ロボット体験演習で自作ロボット（みんな戦略を書いたことある！）
- 難しいものある
  - ロボカップ

- 赤文字は重要。知って置くといい

Acceliteシステム概略



ロボット戦略の流れ

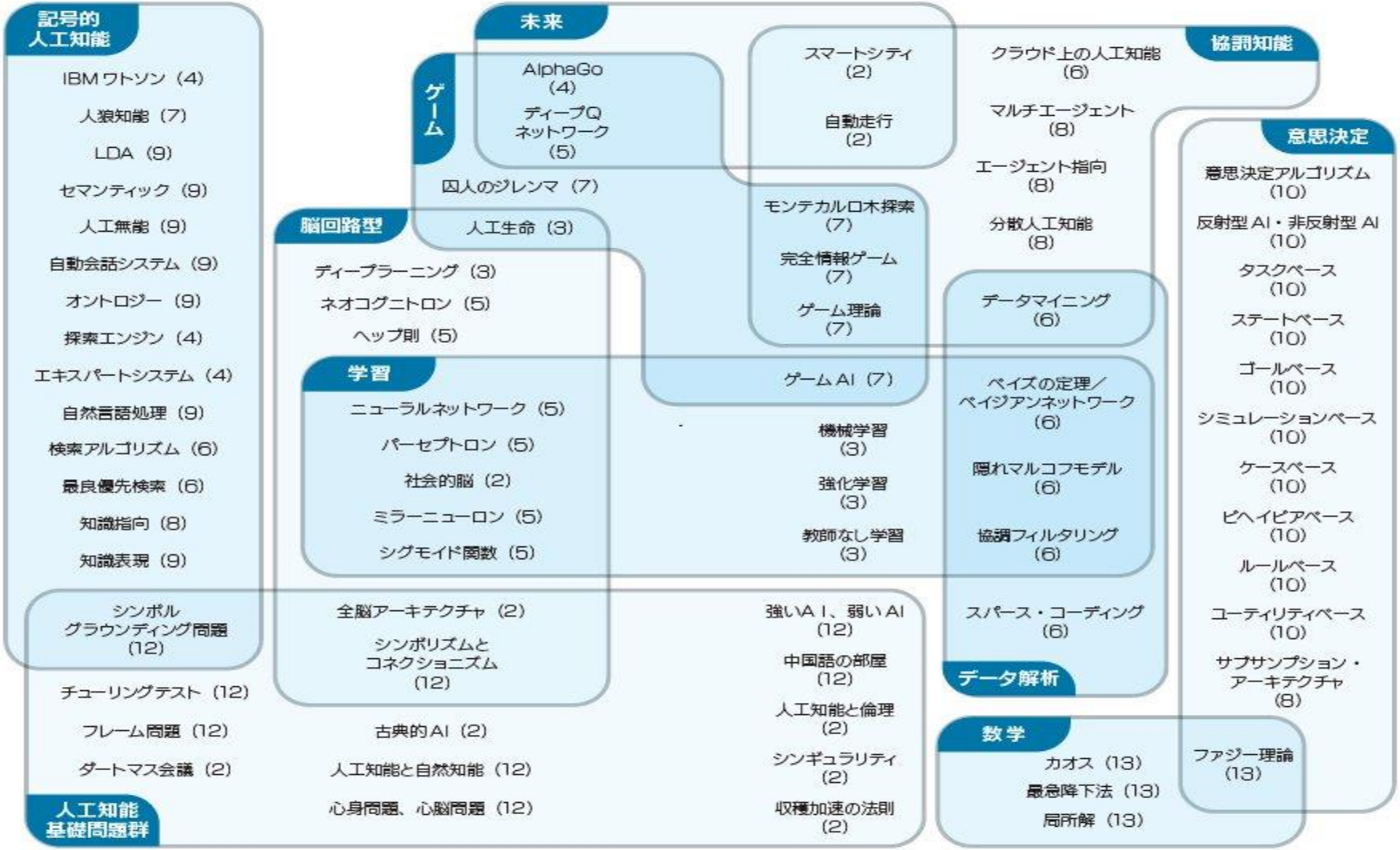

認識の形成
意思の決定
運動の構成
ロボット
意思決定のモジュール
意思決定のモジュール
記憶
制御
センサ
アクチュエータ
環境世界

人工知能(AI)とは？

- Wikiでは ”the study and design of intelligent system”
- いろいろな解釈や意味がある
- これを学問的なAIと呼ばれる

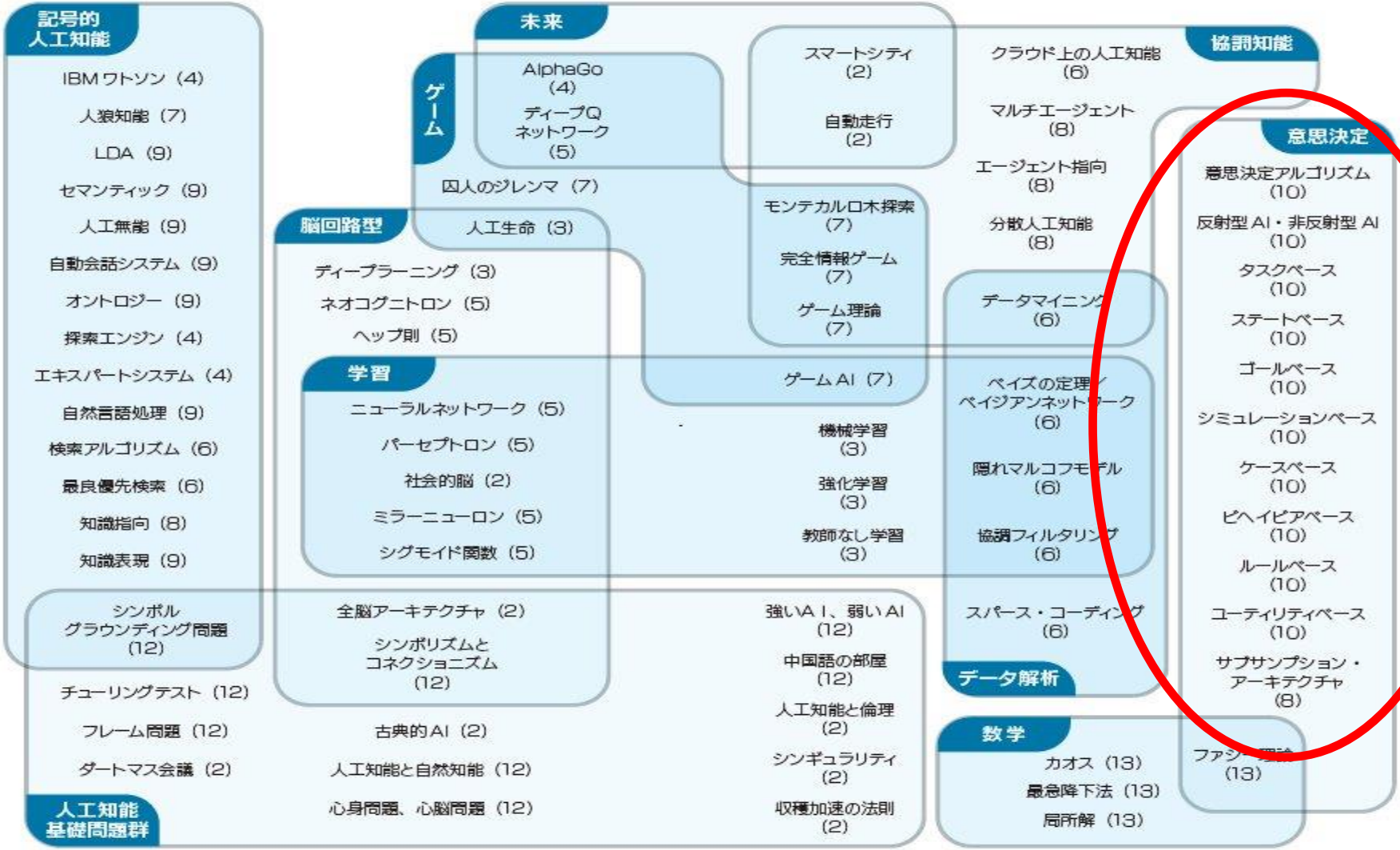
掲載キーワードの系統図

（）内は解説している章



掲載キーワードの系統図

（）内は解説している章



Game AI 部門とは？

- ゲーム業界で開発されている人工知能

- 学問的AI：人の知能をマネする
- Game AI: 知能の錯覚を作る

- 開発者が気になるのはどうやってエージェント（ロボット）「人間らしい」動きをユーザに見せればいいのか

なんでGame AI 使うの？

- 意思決定あるいは戦略が複雑になるほど管理や開発が難しくなる
  - どうやって開発に楽にするか？管理しやすくなるか？

- 意思決定をモデル化にしよう！
  - ゲーム業界も同じく考えゲームAIアーキテクチャを生み出した

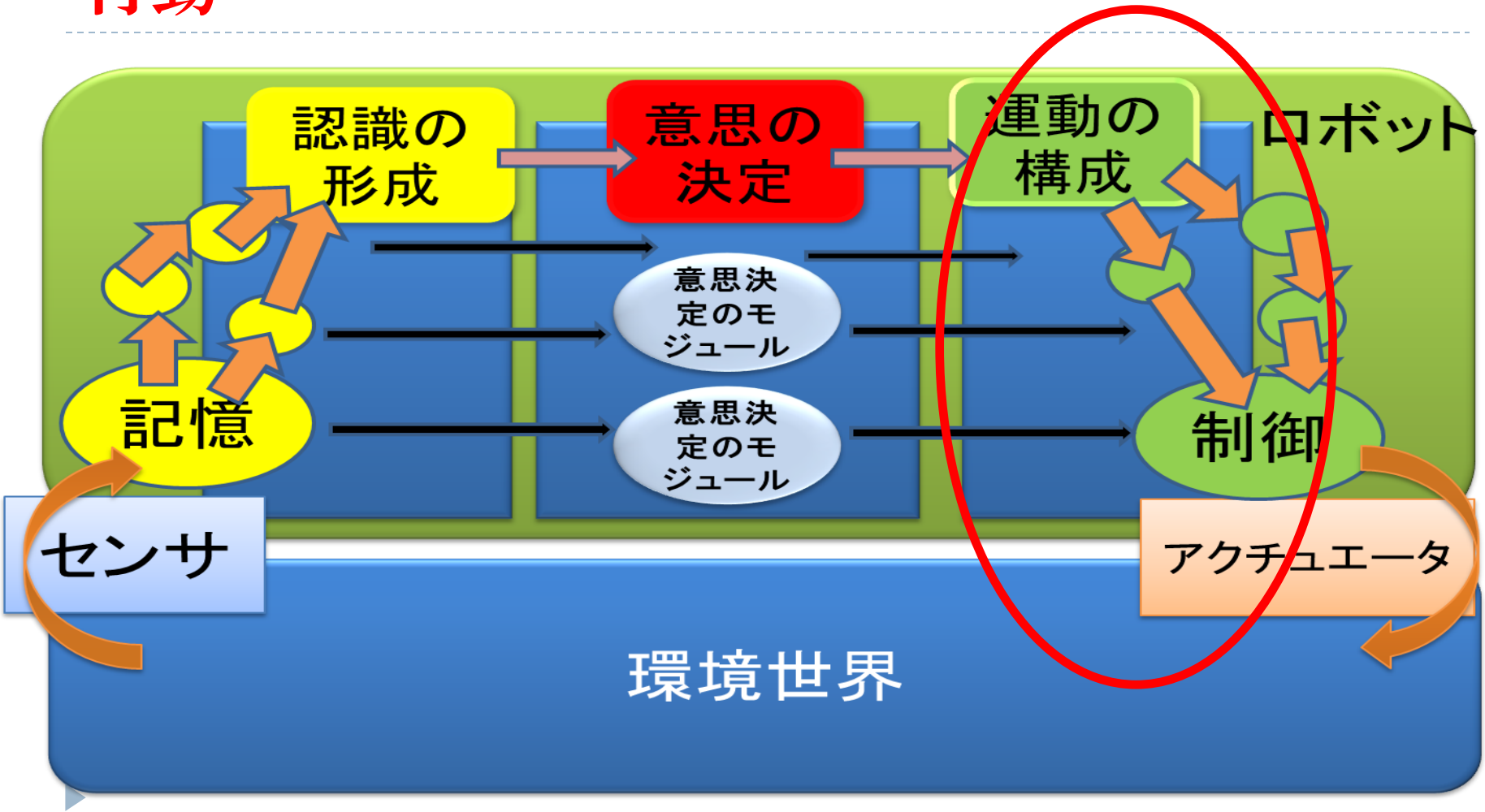
Knowledge Representation and Behavior

情報表現と意思（行動）

行動決定に入る前に…

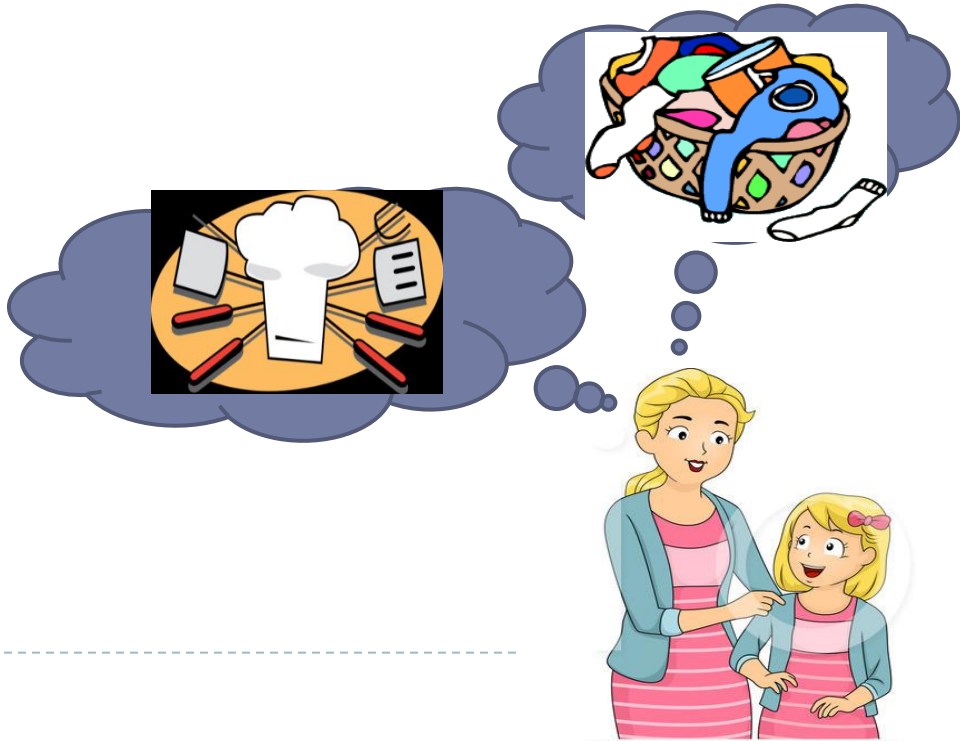
- 情報表現と行動を理解しなければいけません
- 情報表現は行動決定に入力
- 行動は行動決定から出力して実行する

行動


認識の形成
意思の決定
運動の構成
ロボット
意思決定のモジュール
意思決定のモジュール
記憶
制御
センサ
アクチュエータ
環境世界

意思（行動パターン）とは

- 具体的エージェントが何やるか
- 下の処理を一つにまとめて名前を付ける
- 例：
  - 料理する行動の処理：
    - 食料を確保
    - 野菜切る
    - 肉を炒める

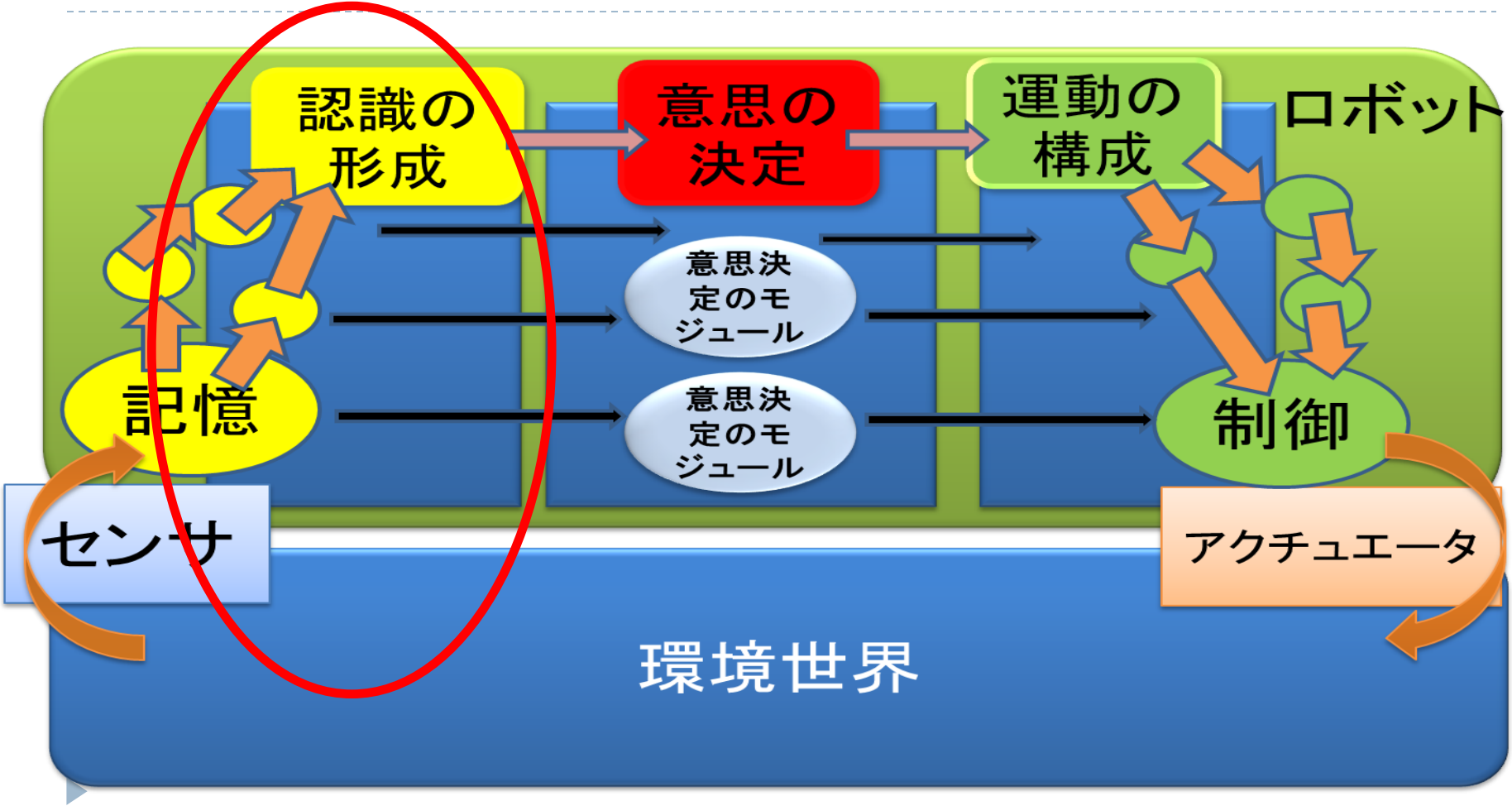


- 処理をprocessと呼ばれる

CITBrainsの基本行動パターン

- SearchBall（ボールを探す）
- ApproachBall（ボールに近ずく）
- TurnAroundBallToTarget（ボール中心旋回）
- AdjustToKickPos（蹴る位置に調整）
- KickBall（ボールを蹴る）
- 他にもいろいろな行動がありますがここではやりません。（次回にやります）

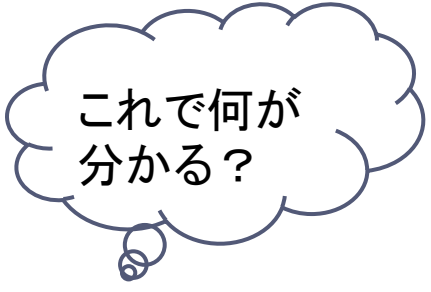
情報表現


認識の形成
意思の決定
運動の構成
ロボット
意思決定のモジュール
意思決定のモジュール
記憶
制御
センサ
アクチュエータ
環境世界

情報表現とは

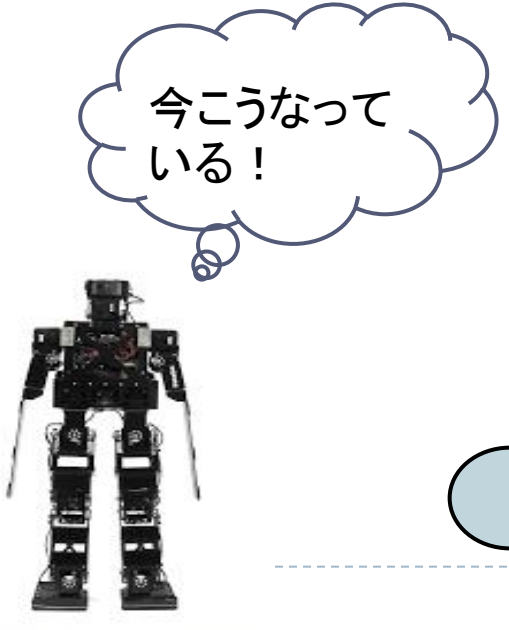
- ロボットが値（センサ）をもらってもこれは何の意味するか分からないー＞意味をつける
- 情報はどうやって表現する?

- センサの値が
  - ボールの座標とか
  - 自分の位置



情報表現とは

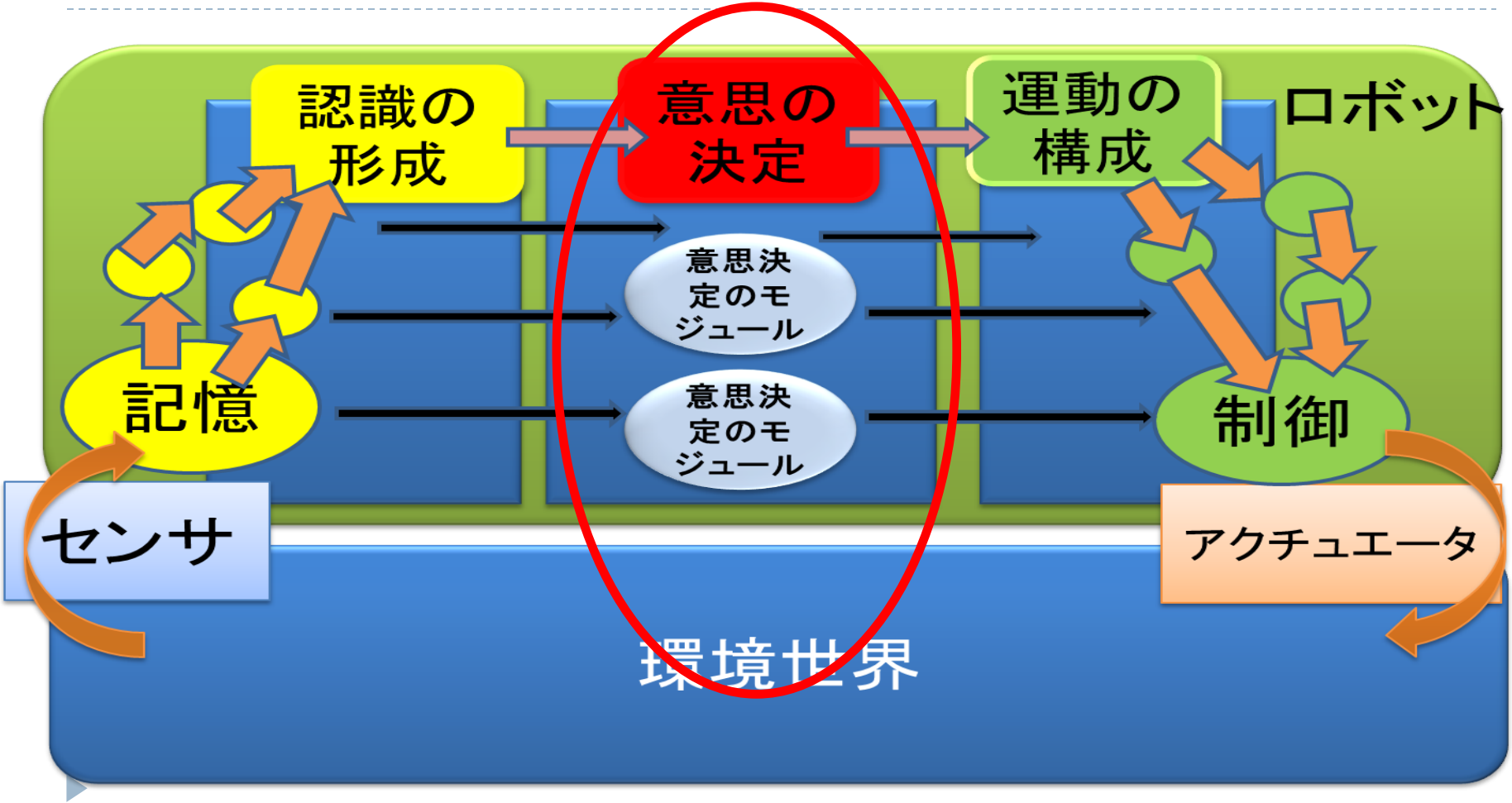
- ボールの座標
  - 足元にあるか？　T/F
  - 敵陣地にあるか？　T/F
- 自分の位置
  - ゴールに向いている？　T/F



Behavior Selection

行動決定

行動決定

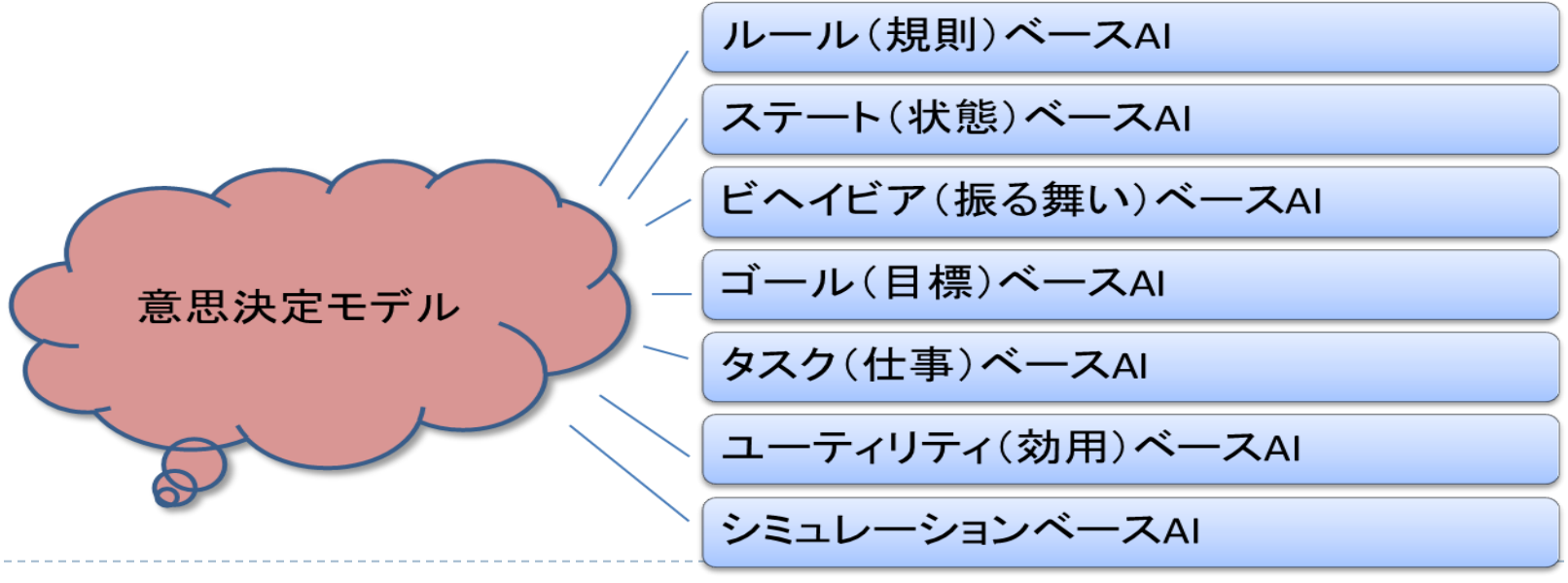

認識の形成
意思の決定
運動の構成
ロボット
意思決定のモジュール
意思決定のモジュール
記憶
制御
センサ
アクチュエータ
環境世界

AIアーキテクチャーとは

- 行動パターン（SearchBall,KickBallとか）をどう選ぶか
  - ＞行動の選び方の構造が意思決定モデル
- 複雑な動きをするロボットの戦略を作るのは難しいためモデル化する。
- いろいろな意思決定モデルがあるが一番簡単と使いやすいのはゲームAIアーキテクチャ

ゲーム用アーキテクチャ

- 高度の意思を簡単にする
- 開発とか人間らしい動きをできる。



2種類あります

- Reactive
  - 反応する
  - 状態によって判断
  - 現代に動く
- アーキテクチャ
  - ステート（状態）ベース
  - ビヘイビア（振る舞い）ベース
- 例：格闘ゲーム

- Non-Reactive
  - 行動する
  - プランニング、計画
  - 未来を予測
- アーキテクチャ
  - ゴール（目標）ベース
  - タスク（仕事）ベース
- 例：FPS

CITBrainsが使ってるAI

- Dynamo(過去キッドサイズ)
  - Finite State Machine (FSM)

  コード : fw3.py
- Accelite(現在キッドサイズ)
  - Goal Oriented Action Planner (GOAP)

  コード : runstrategy.py
- Xega (アダルトサイズ)
  - Hierarchical Task Network Planner (HTN Planner)

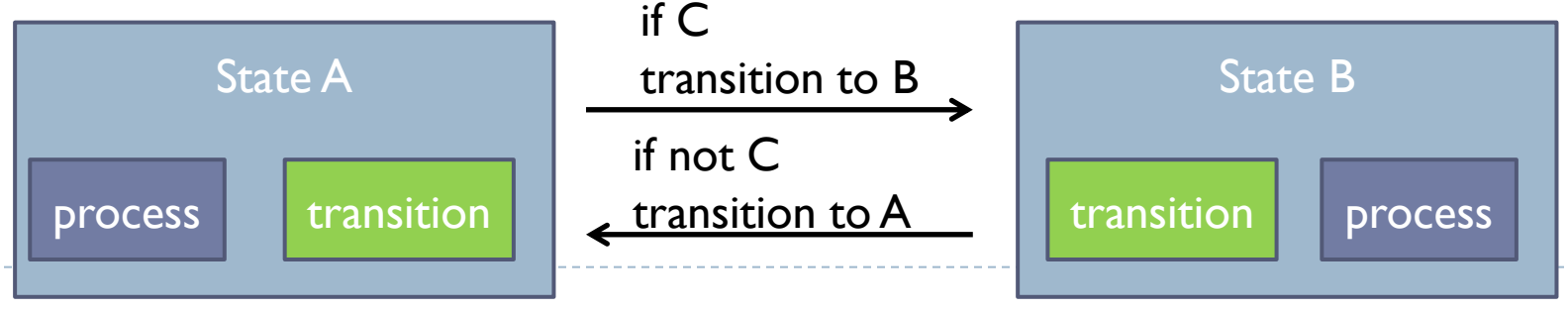
  コード : xegastrategy.py

Finite State Machine

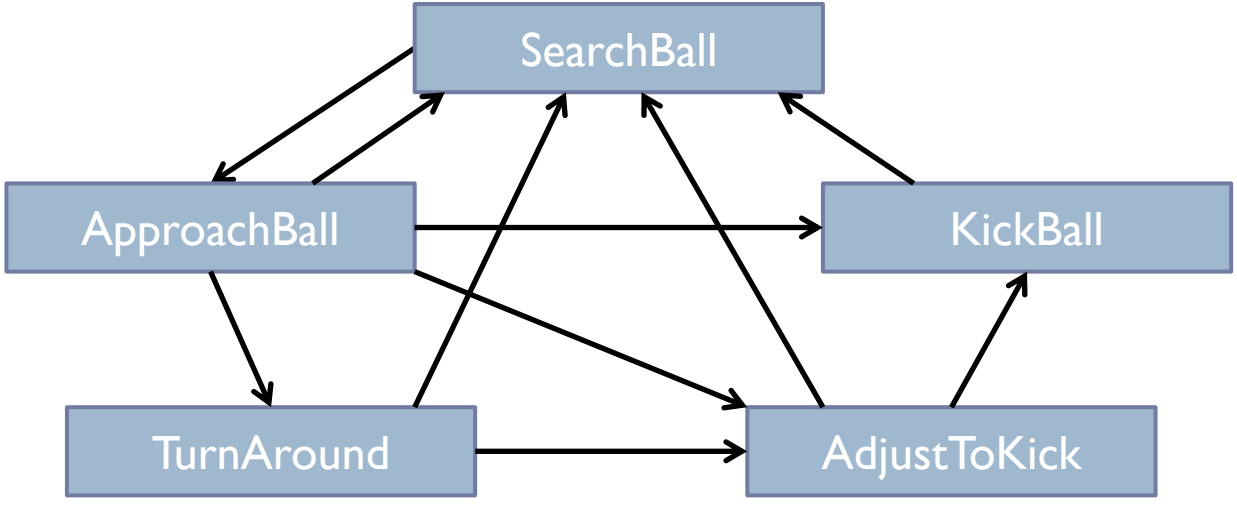
ステート（状態）ベースAI

FSMとは

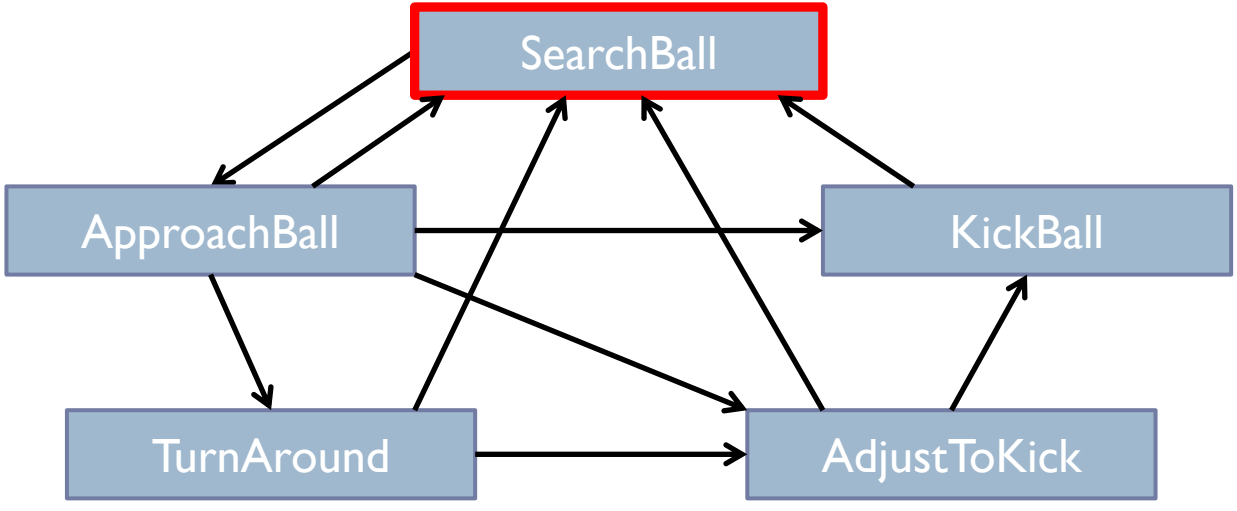
- 一番基本的な行動決定モデル
- 遷移条件（行動の切り替え）を行動の中で行う
- 一つ一つの行動はお互いとつながっている
- 行動をstateと呼ばれる
- 遷移をtransitionと呼ばれる



Accelite例

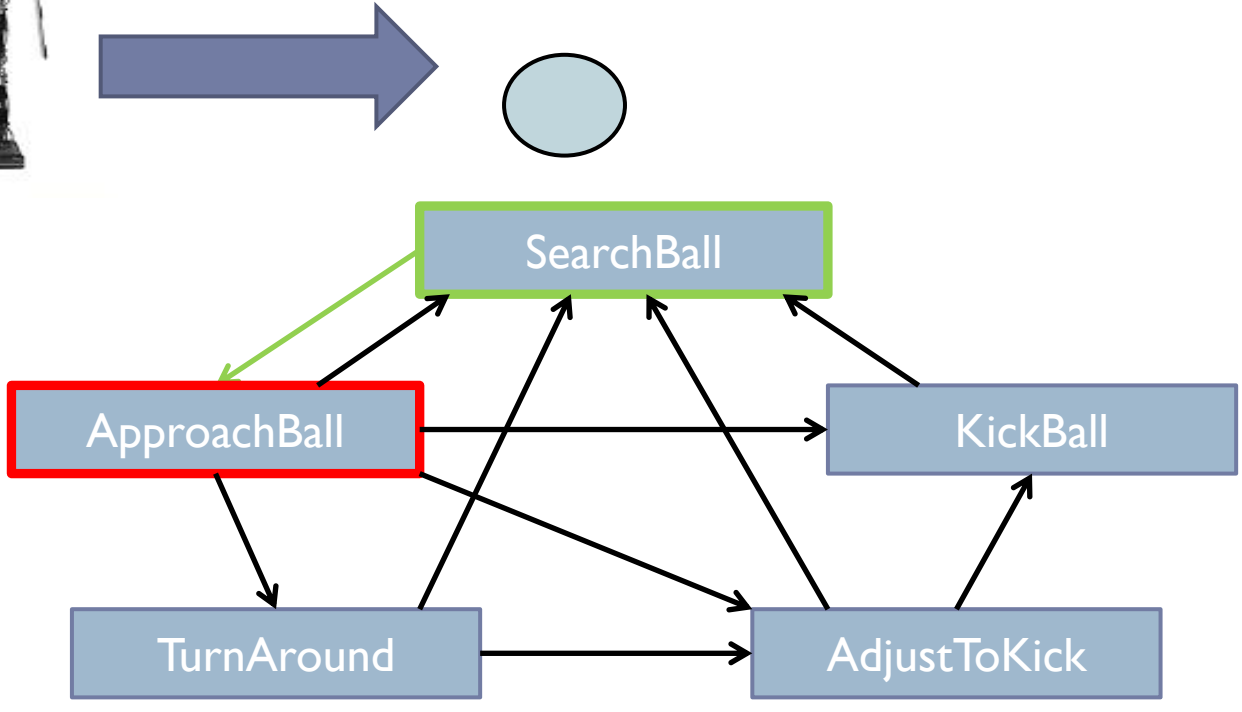

SearchBall
ApproachBall
KickBall
TurnAround
AdjustToKick

World State:ボール見えない



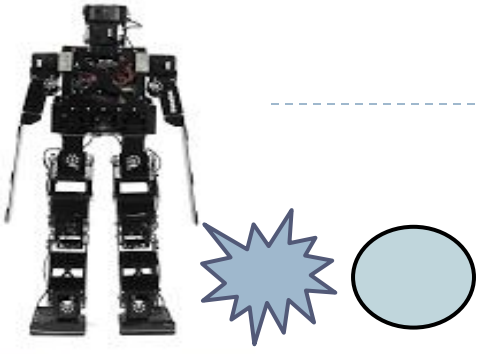
- 赤:実行中　緑:済

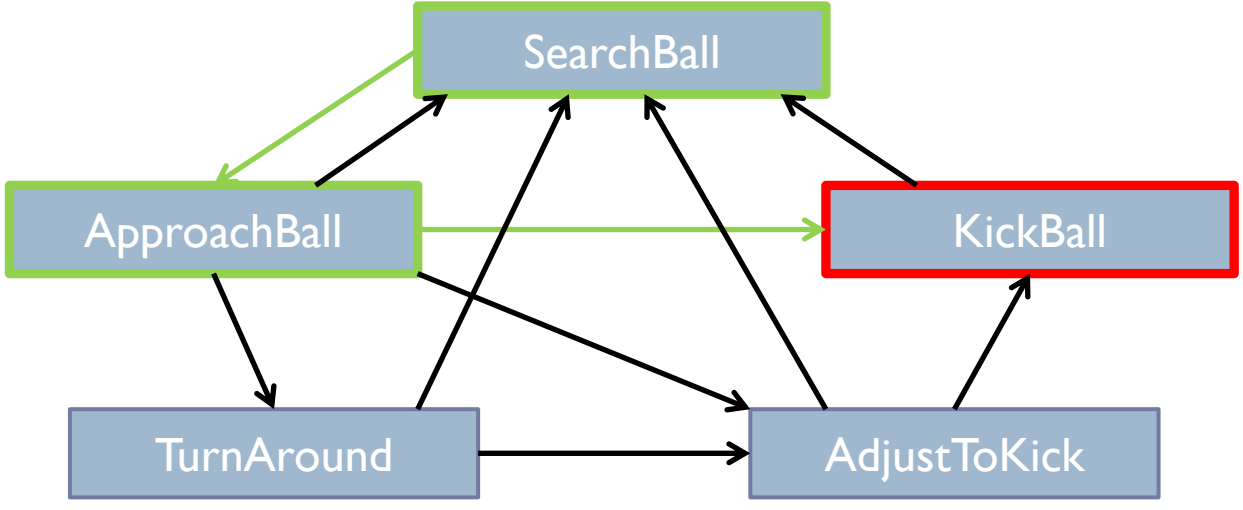
World State:ボール見える
ボールから遠い



赤:実行中　緑:済

World State:ボール見える
ボールに近い

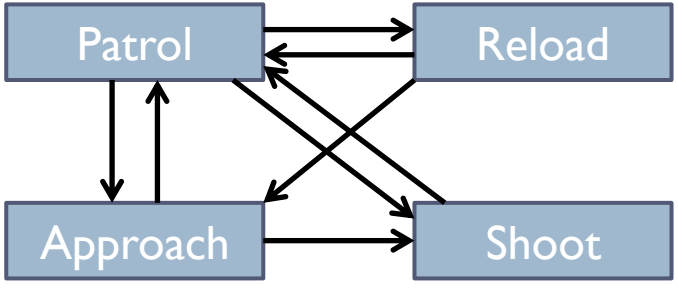




- 赤:実行中　緑:済

サンプル実行しよう！

- python fsm.py
- 初級：情報表現をいじってみよう
- 中級：新しい状態Heal(回復)入れてみよう
- 上級：ロボカップ戦略に変えてみよう



まとめ

- ゲームAIアーキテクチャーの紹介
- 行動と処理の定義
- 情報表現の大切さ
- 意思決定の種類
- ステートAIベース（FSM）の紹介

CITBrains Strategic Division（戦略班）
2018/09/04

Game AI Architecture Part 2
ゲームAIアーキテクチャー

スプラトマン　ジョシュア

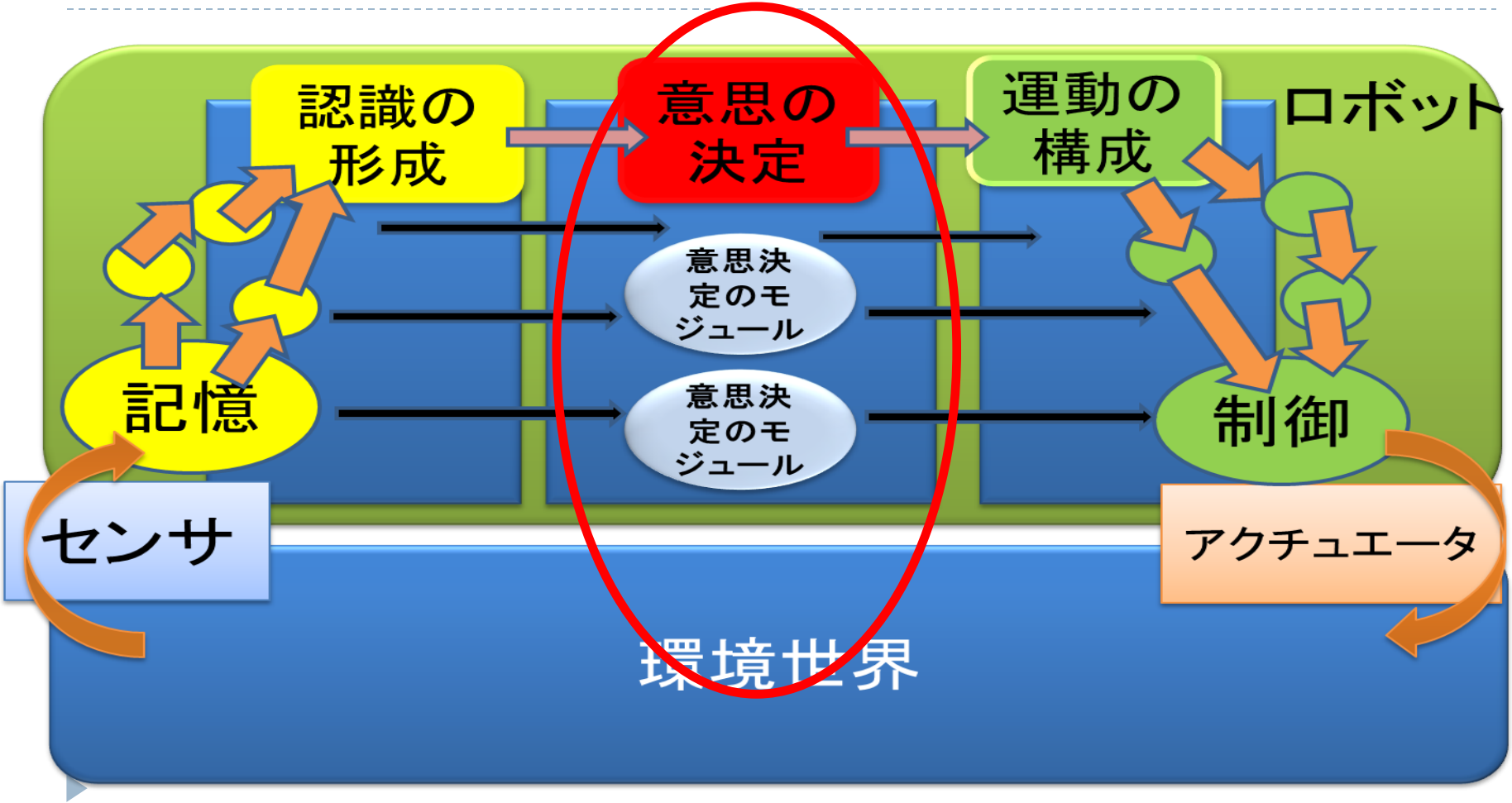
今日のながれ

- Behavior Selection
  - 行動決定の種類
    - **FSM**
    - **GOAP**
    - **HTN Planner**

Behavior Selection

行動決定

行動決定

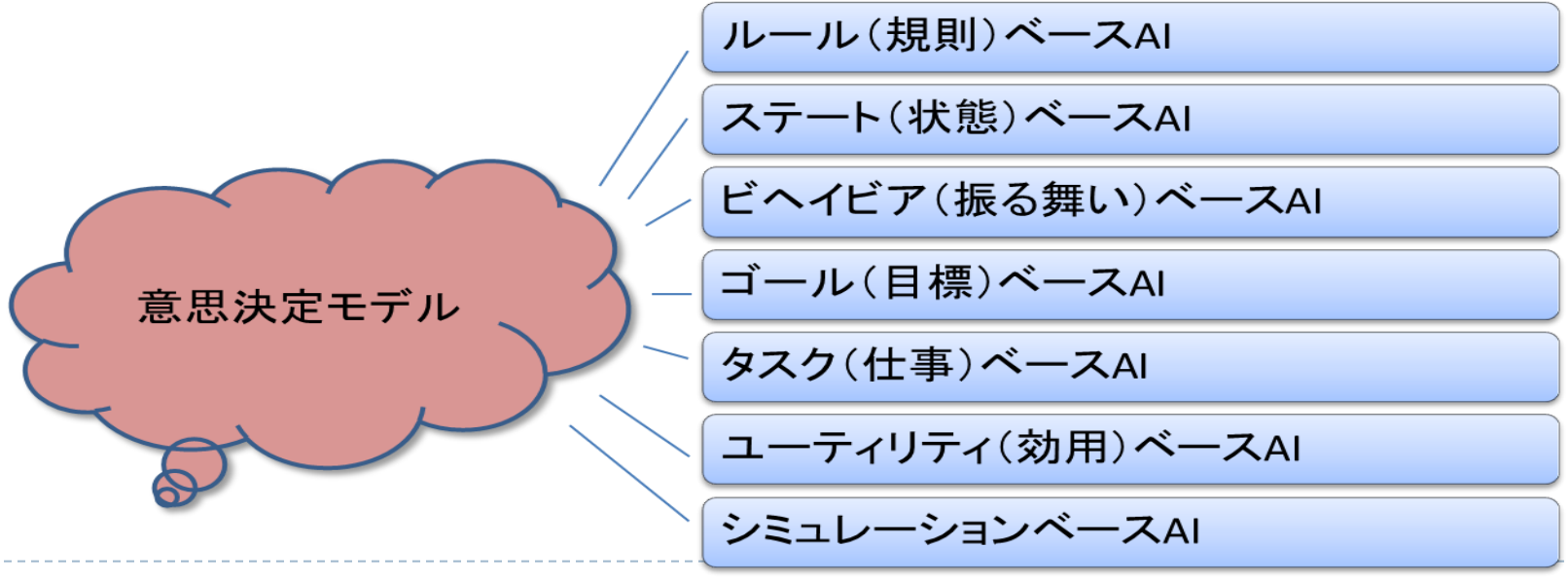

認識の形成
意思の決定
運動の構成
ロボット
意思決定のモジュール
意思決定のモジュール
記憶
制御
センサ
アクチュエータ
環境世界

AIアーキテクチャーとは

- 行動パターン（SearchBall,KickBallとか）をどう選ぶか
  - ＞行動の選び方の構造が意思決定モデル
- 複雑な動きをするロボットの戦略を作るのは難しいためモデル化する。
- いろいろな意思決定モデルがあるが一番簡単と使いやすいのはゲームAIアーキテクチャ

ゲーム用アーキテクチャ

- 高度の意思を簡単にする
- 開発とか人間らしい動きをできる。



2種類あります

- Reactive
  - 反応する
  - 状態によって判断
  - 現代に動く
- アーキテクチャ
  - ステート（状態）ベース
  - ビヘイビア（振る舞い）ベース
- 例：格闘ゲーム

- Non-Reactive
  - 行動する
  - プランニング、計画
  - 未来を予測
- アーキテクチャ
  - ゴール（目標）ベース
  - タスク（仕事）ベース
- 例：FPS

CITBrainsが使ってるAI

- Dynamo(過去キッドサイズ)
  - Finite State Machine (FSM)

  コード : fw3.py
- Accelite(現在キッドサイズ)
  - Goal Oriented Action Planner (GOAP)

  コード : runstrategy.py
- Xega (アダルトサイズ)
  - Hierarchical Task Network Planner (HTN Planner)

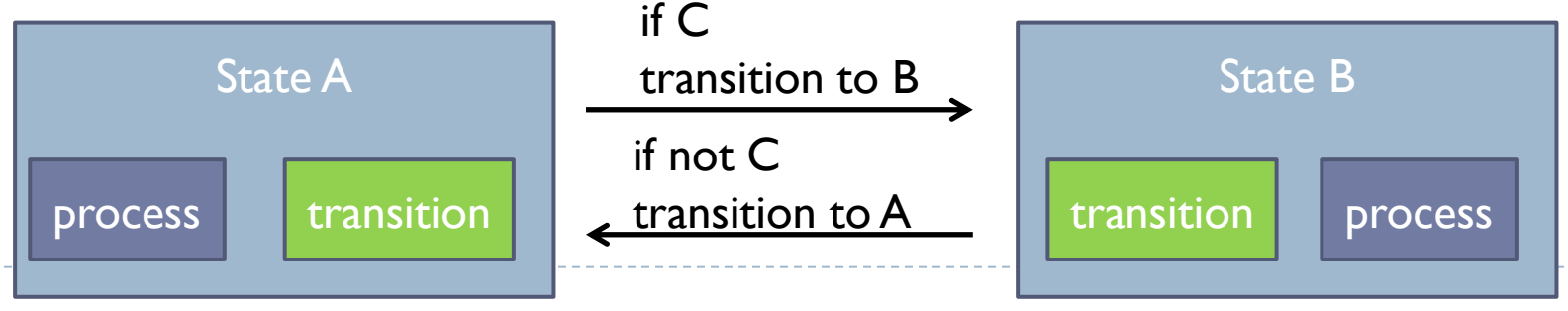
  コード : xegastrategy.py

Finite State Machine

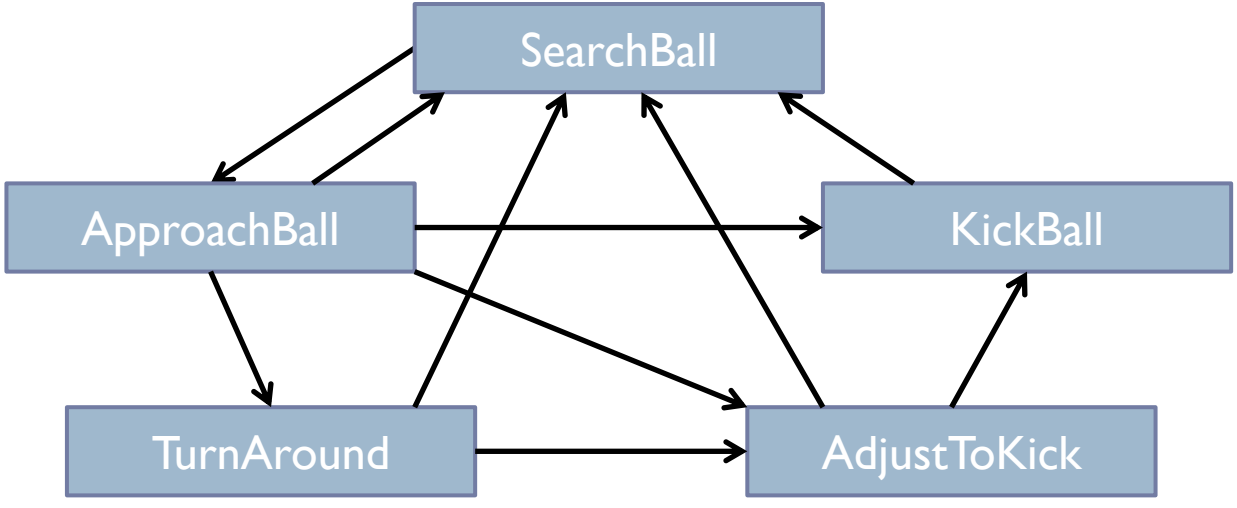
ステート（状態）ベースAI

FSMとは

- 一番基本的な行動決定モデル
- 遷移条件（行動の切り替え）を行動の中で行う
- 一つ一つの行動はお互いとつながっている
- 行動をstateと呼ばれる
- 遷移をtransitionと呼ばれる

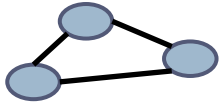


Accelite例

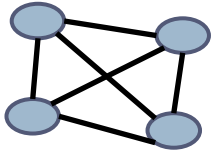

SearchBall
ApproachBall
KickBall
TurnAround
AdjustToKick

FSM Merit/Demerit

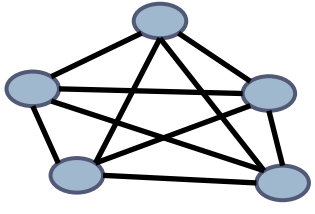
- Merit
  - 楽に作れる
- Demerit
  - 再利用が難しい
  - 行動が増えると管理が難しい



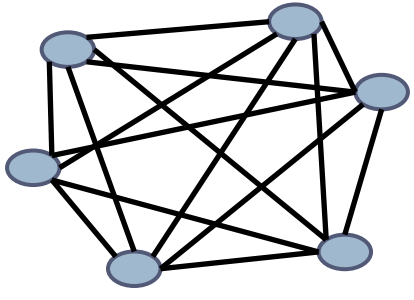
状態: 3
遷移: 3



状態: 4
遷移: 6



状態: 5
遷移: 10



状態: 6
遷移: 15

?

状態: 7
遷移: 21

Goal Oriented Action Planning

ゴール（目標）ベースAI

GOAPとは

- ある目的に対して行動を計画を行う
- 行動計画は探索（経路計画）アルゴリズムを使用（具体的にA*アルゴリズムだが今回は紹介だけなのでやりません）

- FSMの問題対策のため行動をSTRIPSベース

- 行動をactionと呼ばれる
- 目的をgoalと呼ばれる

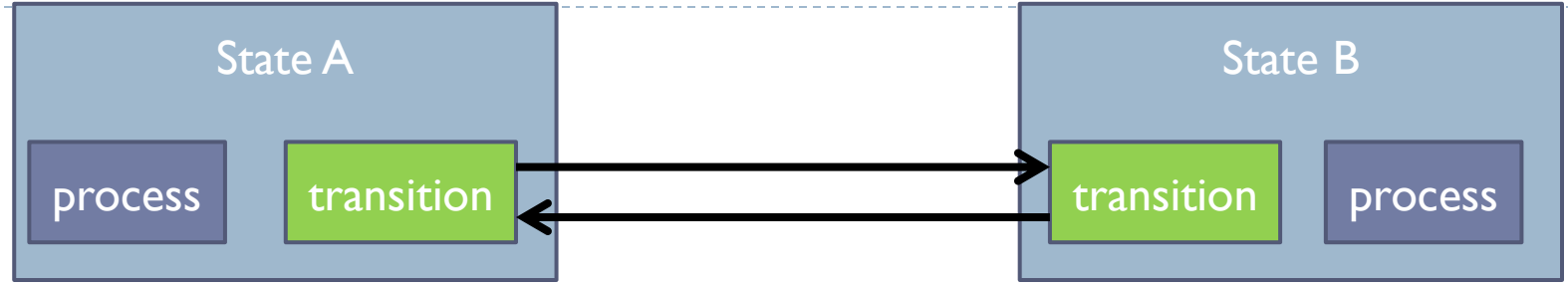
STRIPS

- **St**andford **R**esearch **I**nstitute **P**roblem **S**olver
  - **Richard FikesとNils Nilssonが開発した自動計画**
- 簡単に言うと各行動は独立して、処理と前提条件と事後条件を含む

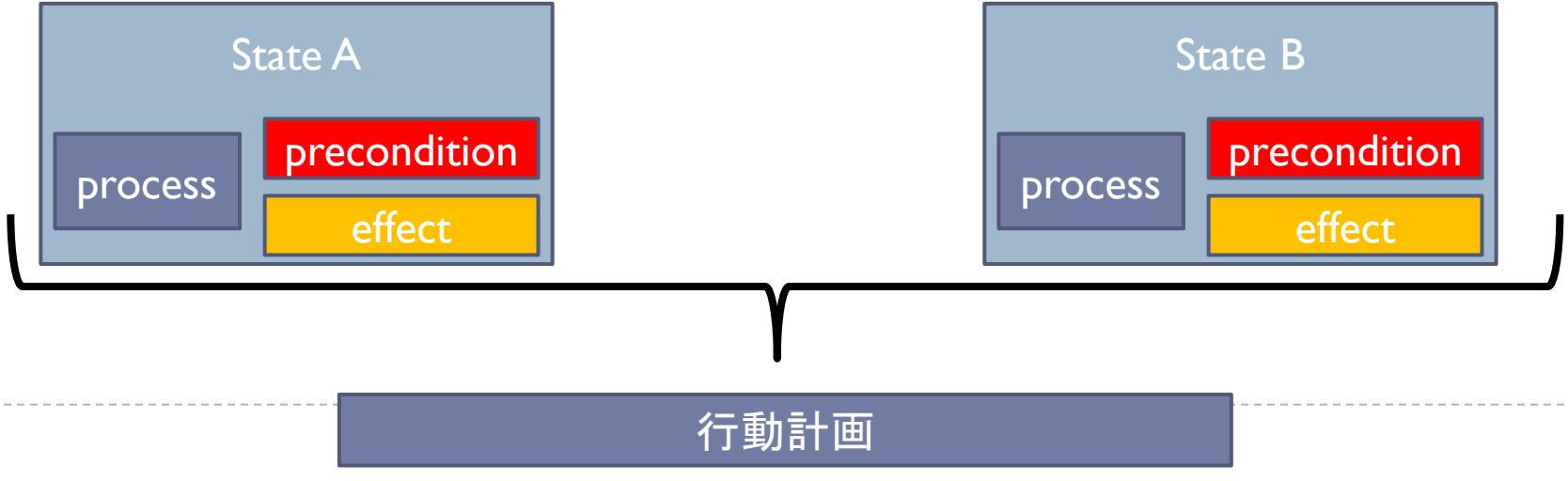
- 行動が独立しているため状態空間として扱うことができる
  - **最適化問題として扱える**

- 前提条件はprecondition
- 事後条件はpostconditionまたはeffect

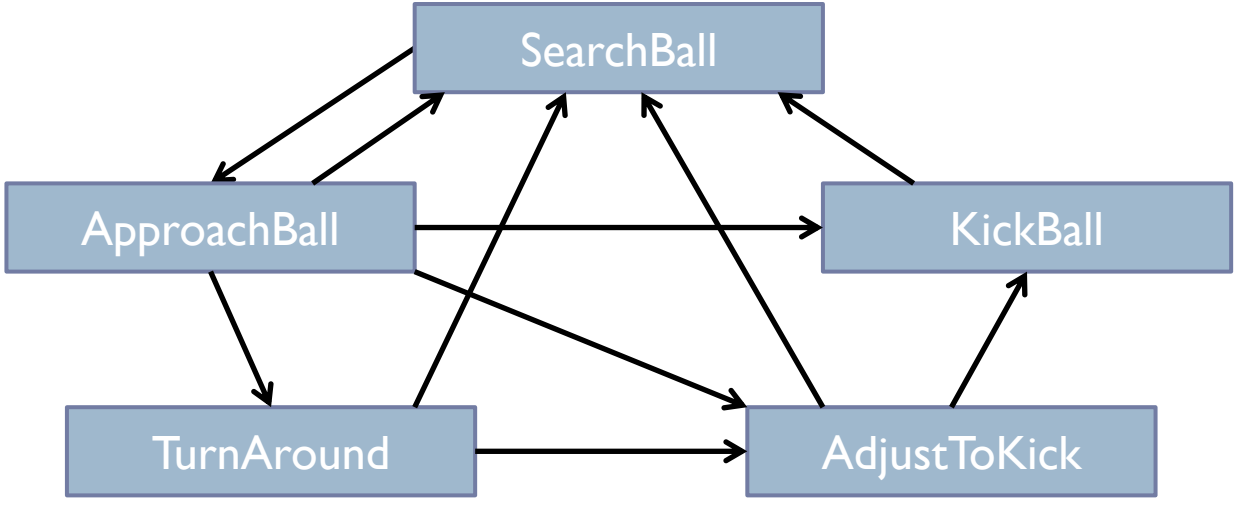
- 一般


State A
process
transition
State B
transition
process

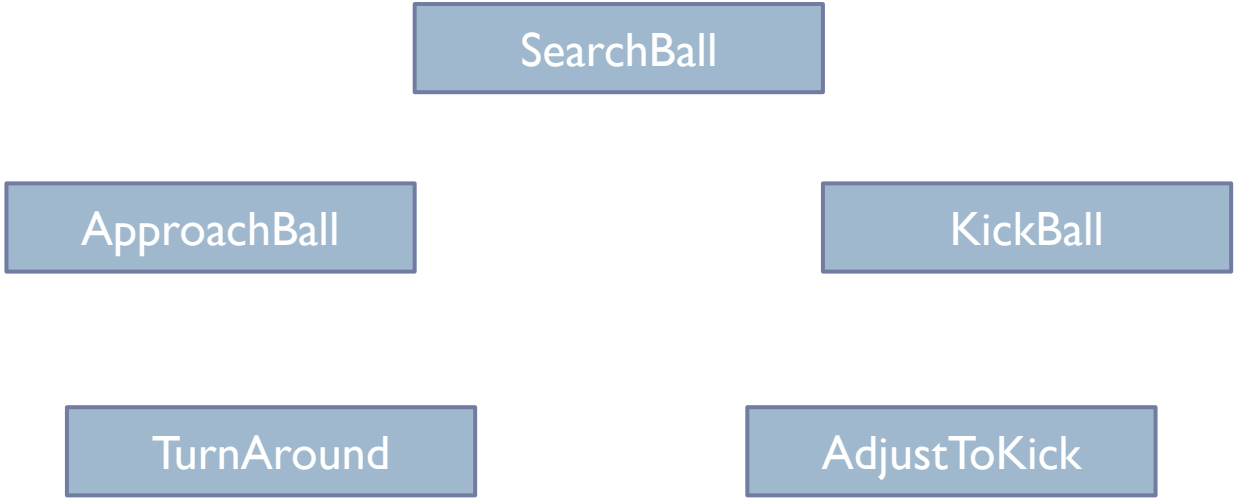
- STRIPS


State A
process
precondition
effect
State B
process
precondition
effect
行動計画

FSMの場合


SearchBall
ApproachBall
KickBall
TurnAround
AdjustToKick

GOAPの場合


SearchBall
ApproachBall
KickBall
TurnAround
AdjustToKick

SearchBall

ApproachBall

KickBall

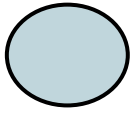
TurnAround

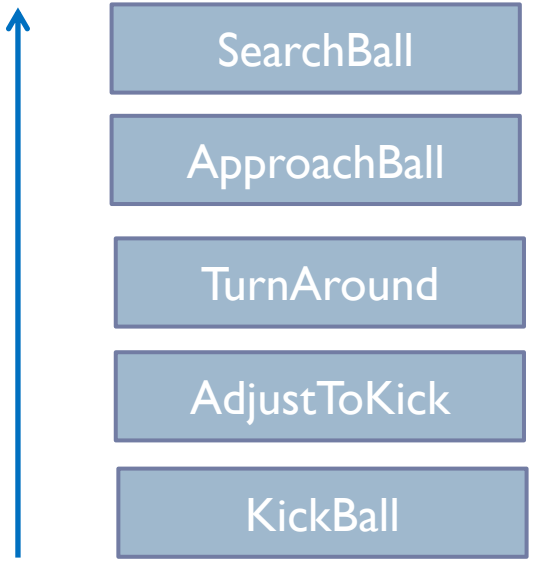
AdjustToKick

Goal State: ボールが蹴れる

- 赤：実行中　緑：済　青：計画

World State:ボール見えない

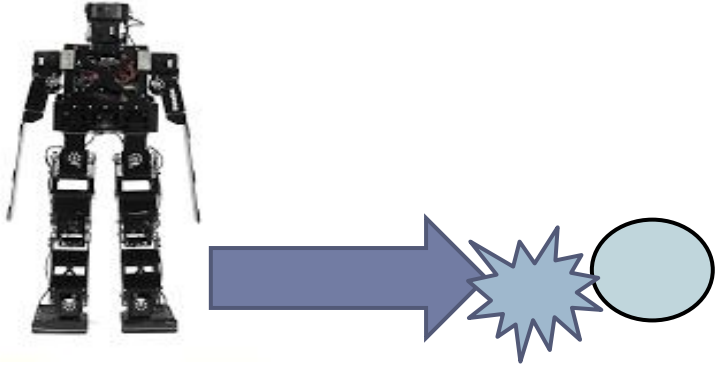


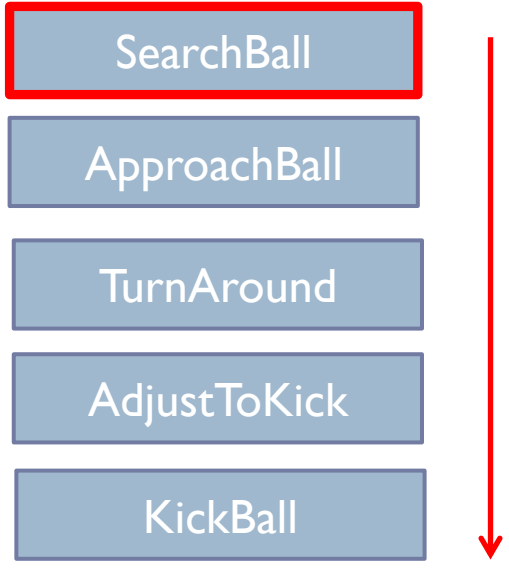


Goal State:　ボールが蹴れる

- 赤:実行中　緑:済　青:計画

World State:ボール見えない

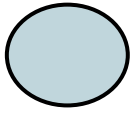


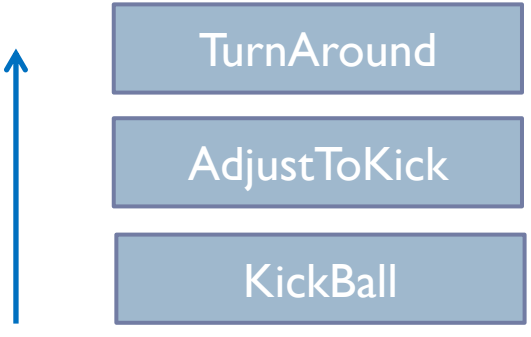


Goal State: ボールが蹴れる

- 赤:実行中　緑:済　青:計画

World State:ゴールに向いていない

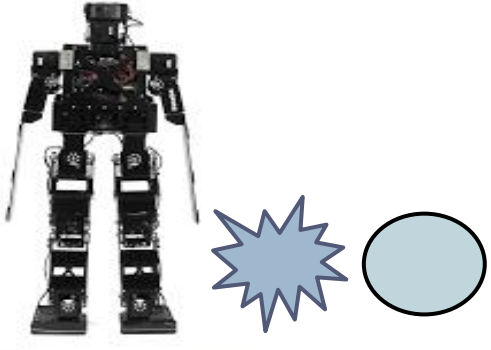


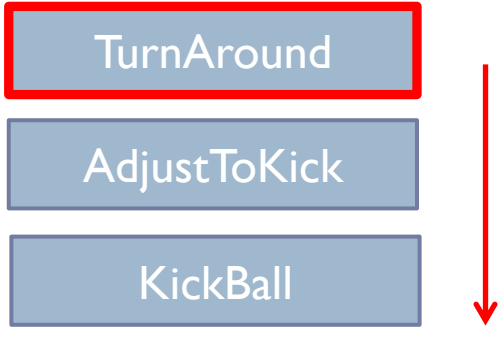


Goal State: ボールが蹴れる

赤:実行中 緑:済 青:計画

World State:ゴールに向いていない

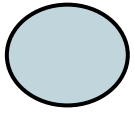




Goal State: ボールが蹴れる

- 赤:実行中 緑:済 青:計画

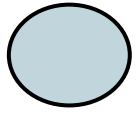
World State:ボール見えない



Goal State:　ボールに近い

- 赤:実行中　緑:済　青:計画

World State:ボール見える



SearchBall

ApproachBall

Goal State: ボールに近い

- 赤:実行中 緑:済 青:計画

サンプル実行しよう！

- python main.py
- 初級：情報表現をいじってみよう
- 中級：新しい状態Heal(回復)入れてみよう
- 上級：ロボカップ戦略に変えてみよう

Patrol

Reload

Approach

Shoot

GOAP のMerit/Demerit

- Merit
  - 新しい行動を入れるのは簡単
  - 自動計画
- Demerit
  - 好きなパターンを作るのが難しい
    - デバッグしにくい

Hierarchical Task Network Planner

タスク（仕事）ベースAI

HTN Plannerとは

- 階層化した行動の集合を分解し計画する
- 行動を抽象的に扱える
- 階層化した行動集合（task tree）は人間側が作る

- FSMの問題対策のため行動をSTRIPSベース

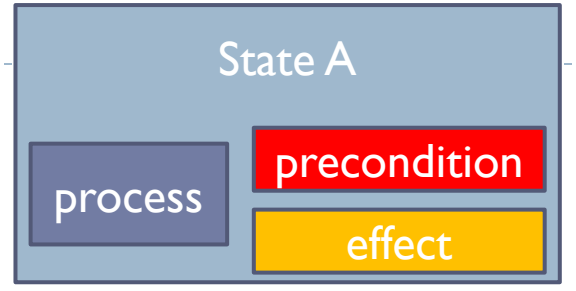
- 行動をtaskと呼ばれる
  - **実行する行動はprimitive taskと呼ばれる**
  - **抽象的行動はcompound taskと呼ばれる**

- 抽象的行動は計画するのみ使い実際に実行しない

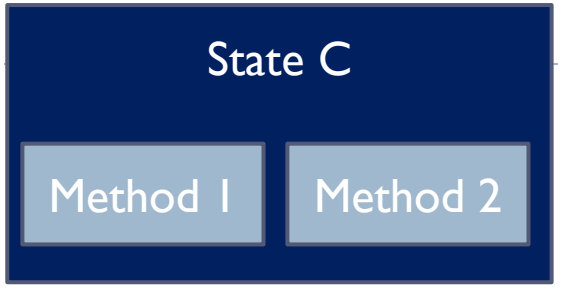
抽象的の行動扱い

- 行動を階層化可能であるため行動の集合を一つの行動として抽象的扱える（compound task）
- 行動集合はsubtaskと呼ばれる
- STRIPSの処理の部分を行動集合に入れ替え、事後条件を外したものがmethodと呼ばれる
- 抽象的行動はmethodの集まりです

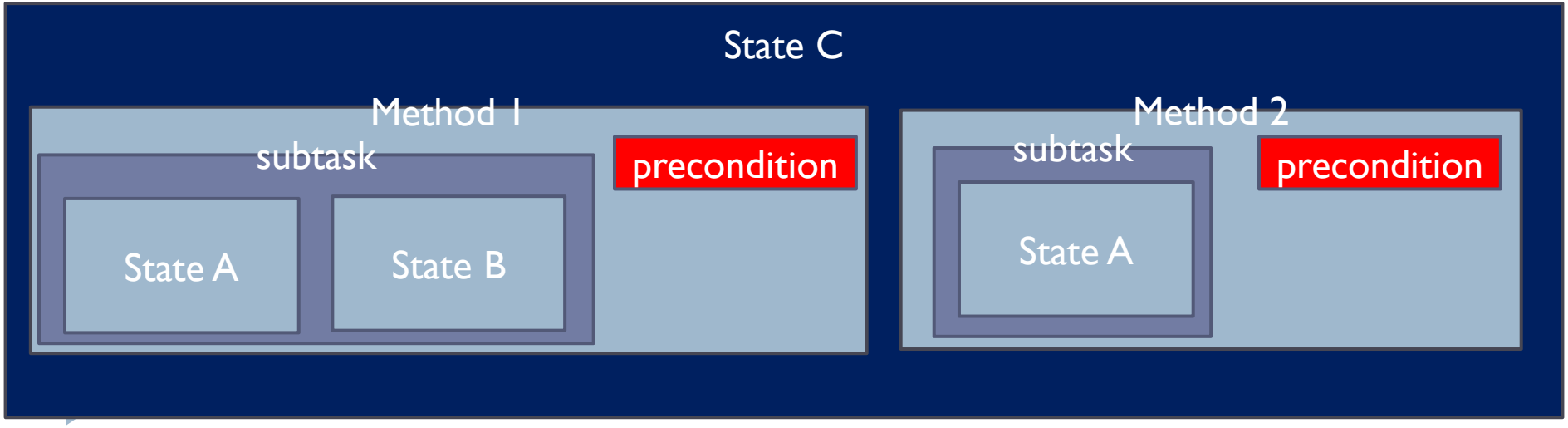
Primitive Task



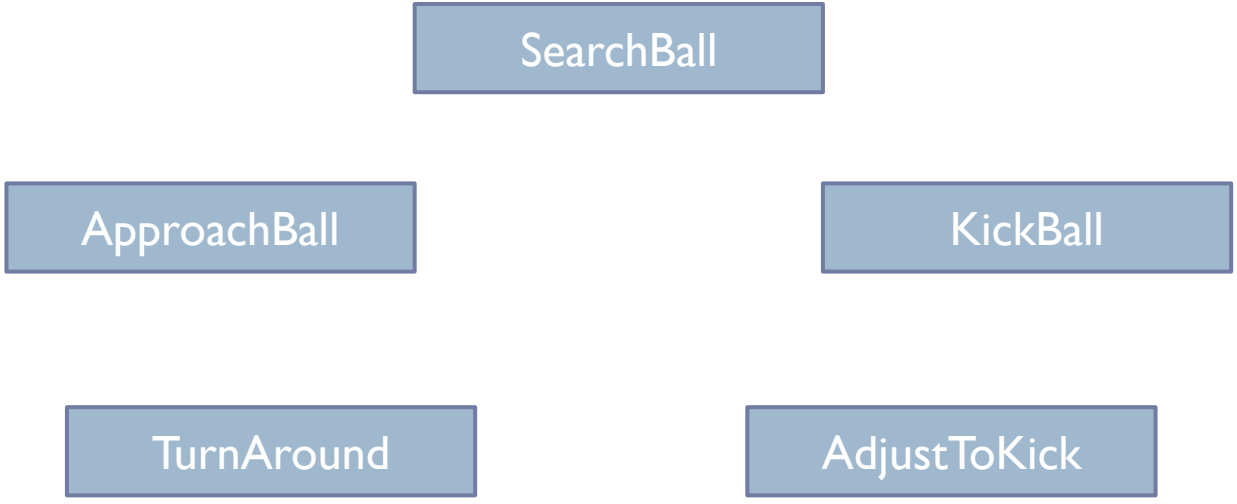
Compound Task



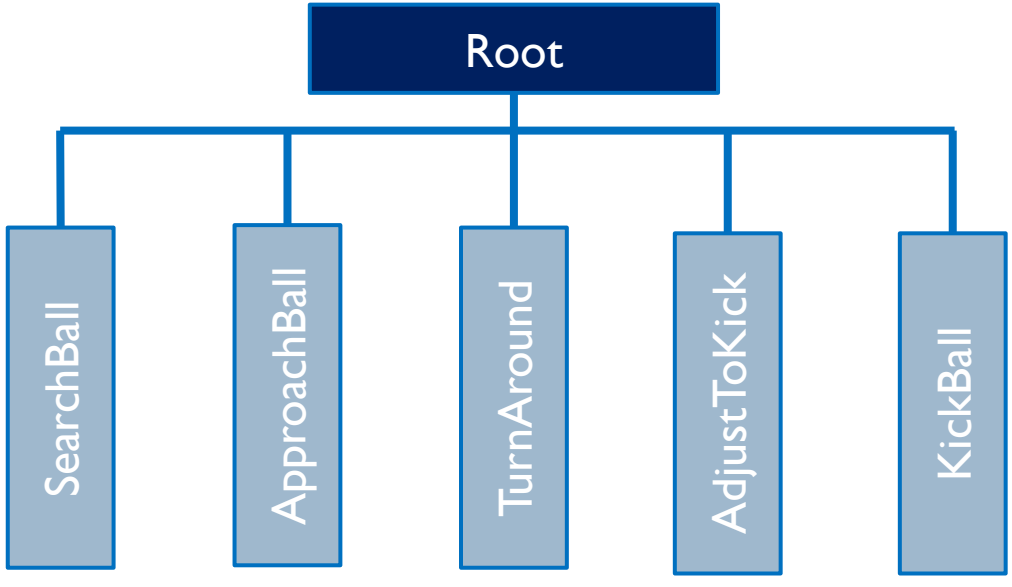
Compound Taskの中身



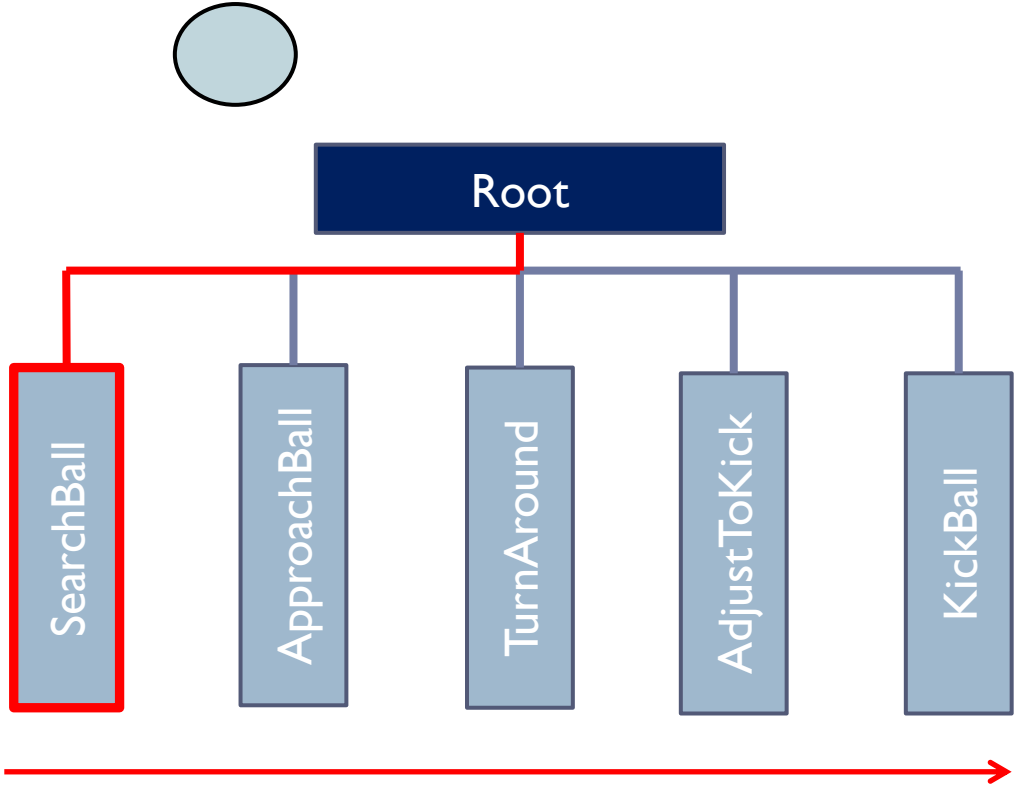
GOAPの場合


SearchBall
ApproachBall
KickBall
TurnAround
AdjustToKick

HTN Plannerの場合(1層)

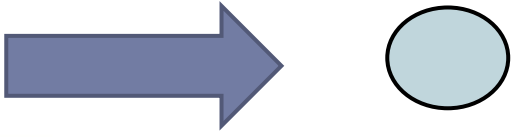


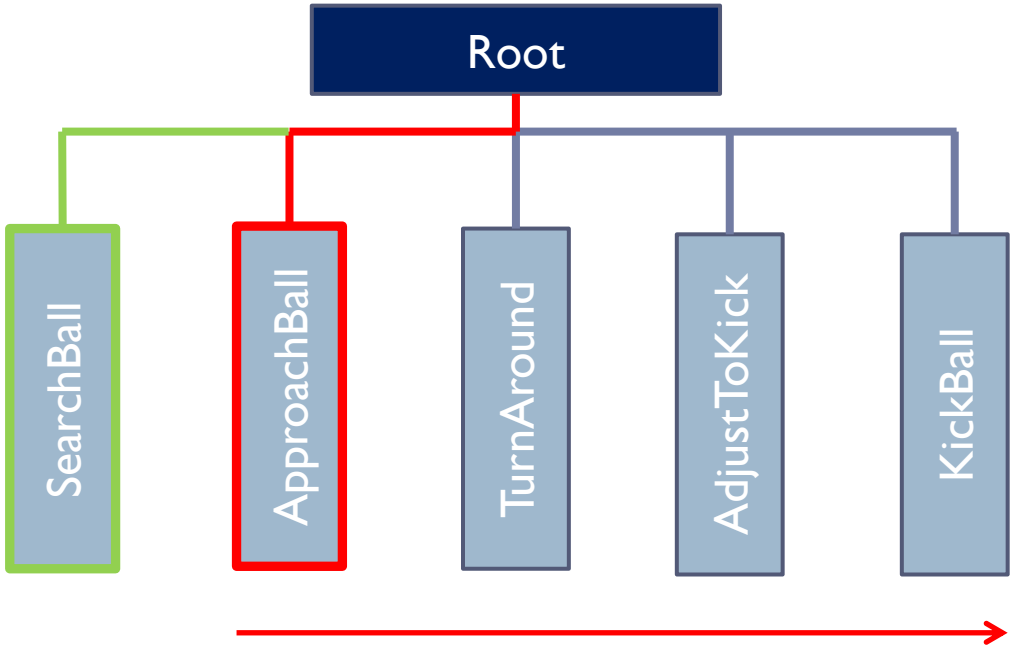
World State:ボール見えない



- 赤:実行中　緑:済　青:計画

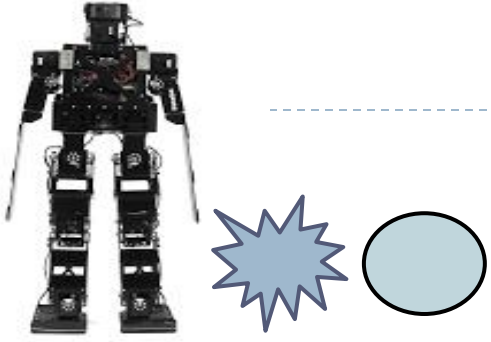
World State:ボール見える
ボールから遠い

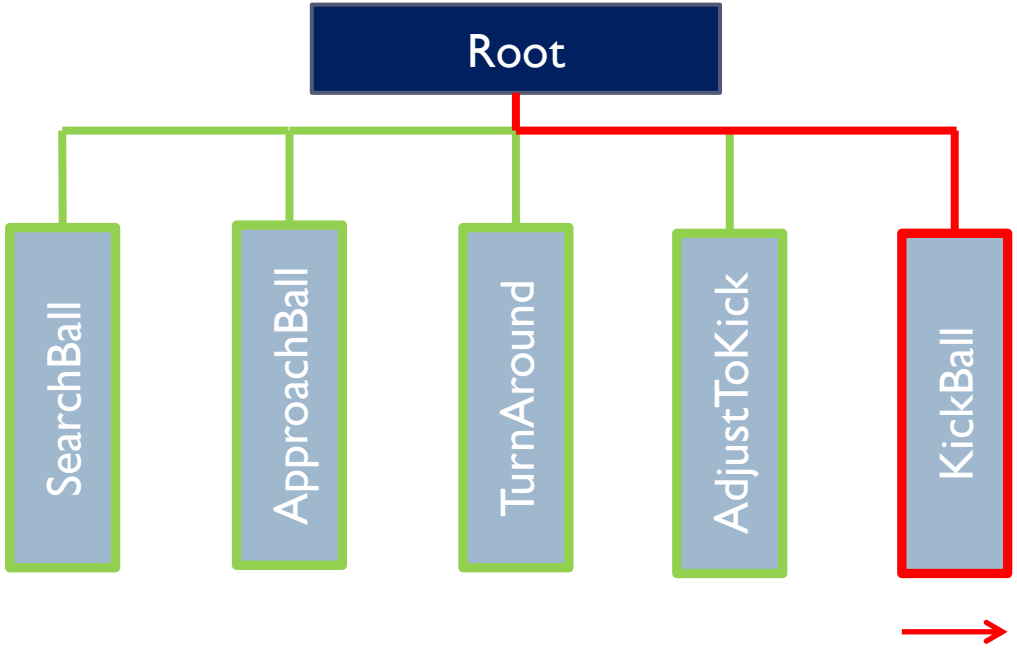




赤:実行中　緑:済　青:計画

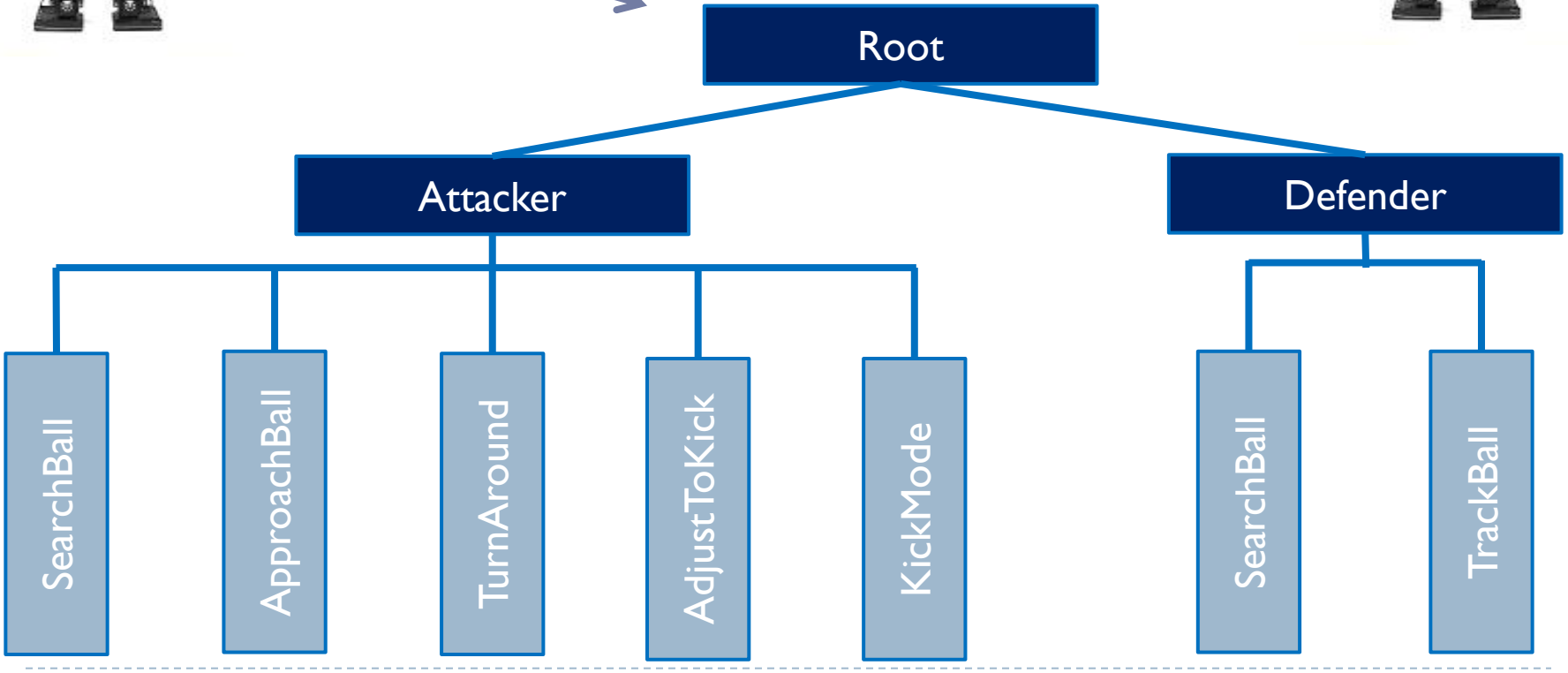
World State:ボール見える
ボールに近い





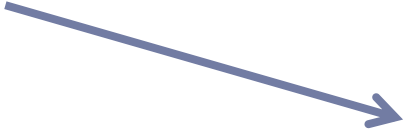
赤:実行中　緑:済　青:計画

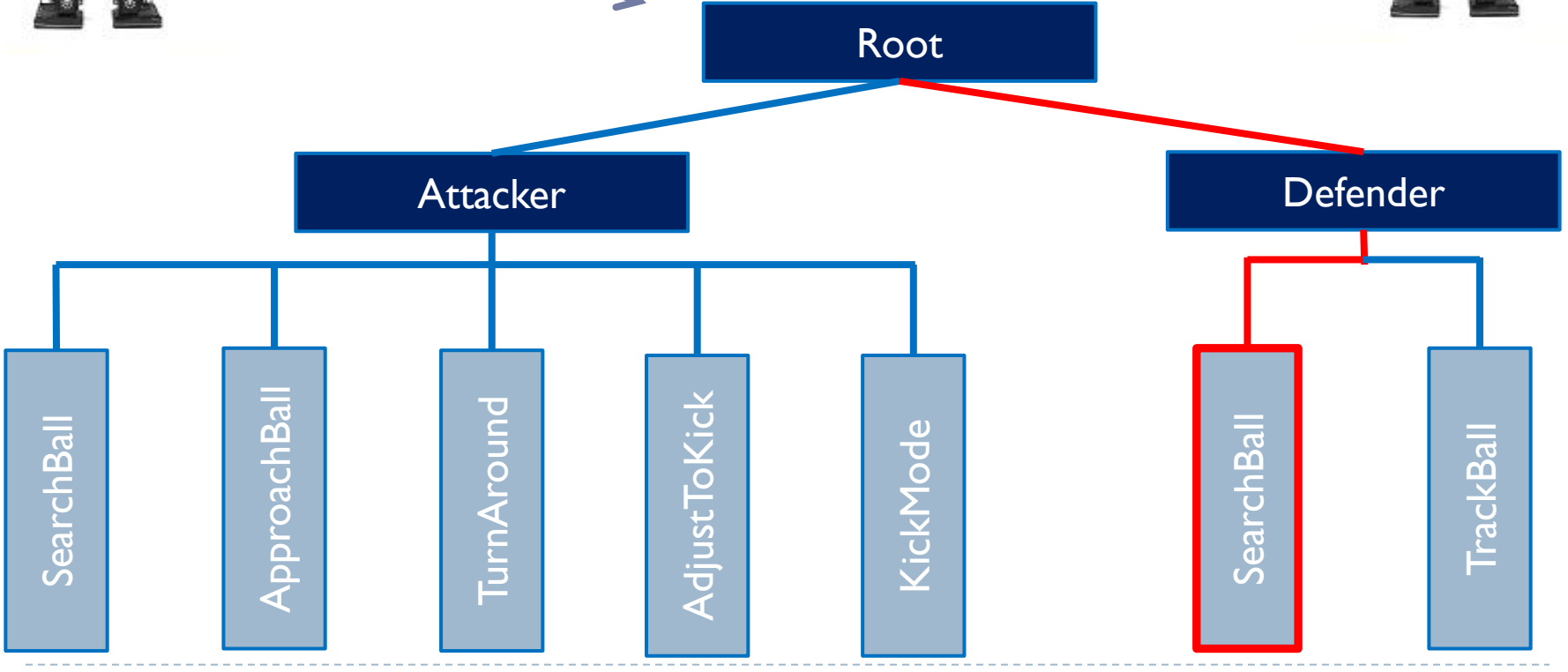

????


Root
Attacker
Defender
SearchBall
ApproachBall
TurnAround
AdjustToKick
KickMode
SearchBall
TrackBall

じゃDefender

ぼくはAttacker

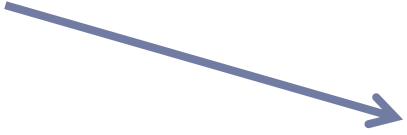




赤:実行中　緑:済　青:計画

ぼくAttacker

サービス中



Root

Attacker

Defender

SearchBall

ApproachBall

TurnAround

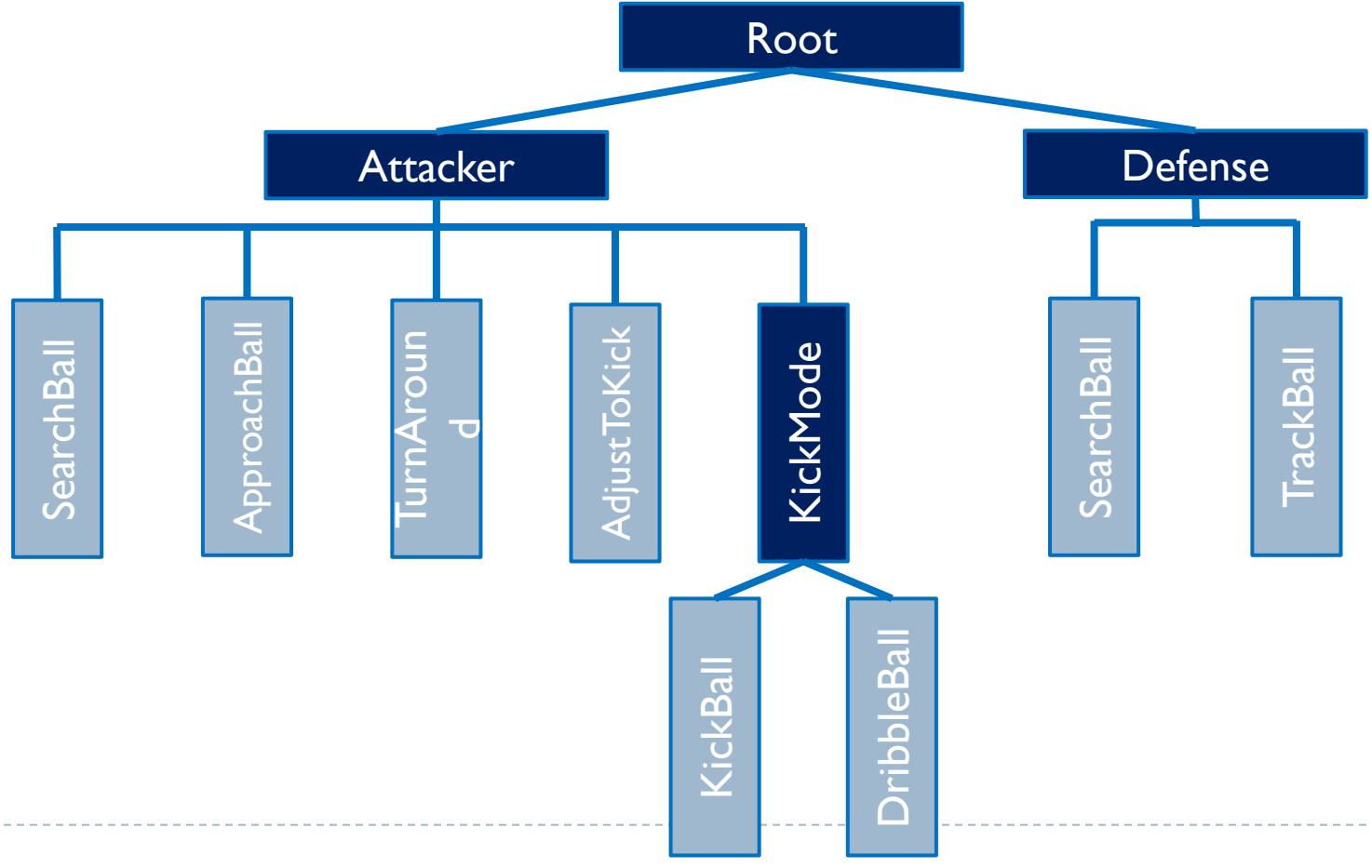
AdjustToKick

KickMode

SearchBall

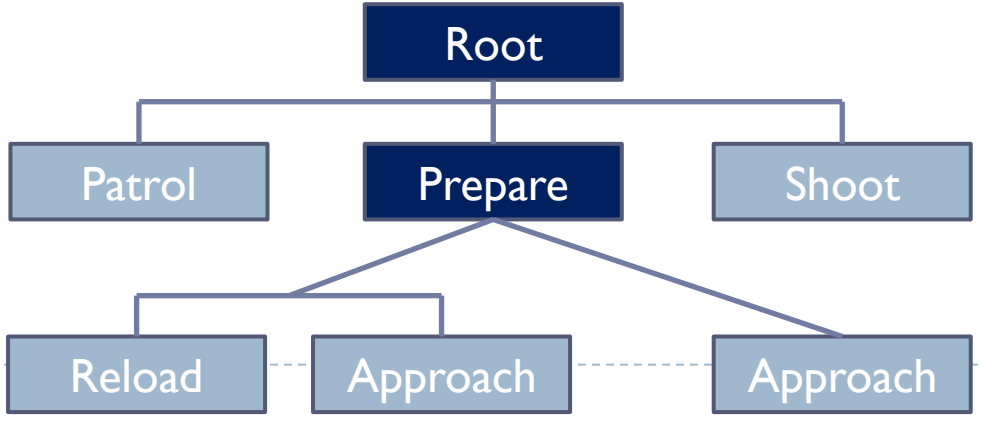
TrackBall

組み合わせても大丈夫


Root
Attacker
Defense
SearchBall
ApproachBall
TurnAround
AdjustToKick
KickMode
SearchBall
TrackBall
KickBall
DribbleBall

サンプル実行しよう！

- python main.py
- 初級：情報表現をいじってみよう
- 中級：新しい状態Heal(回復)入れてみよう
- 上級：ロボカップ戦略に変えてみよう



HTN Planner のMerit/Demerit

- Merit
  - 新しい行動を入れるのは簡単
  - 好きなパターンを作るのが簡単
    - デバッグしやすい
  - 抽象的に考えることができる

- Demerit
  - 自動計画がなくなる

まとめ

- 意思決定の復習
- ステートAIベース（FSM）の復習
- ゴールAIベース（GOAP）の紹介
- タスクAIベース（HTN Planner）の紹介

最後に…

最後にもっと知りたい

- 三宅陽一郎さん(Square Enix AI担当者)が資料をネットで出しています。
- 英語でSteve RabinのGame AI Pro シリーズをおすすめします。
- 個別質問受け取ります。