


自律機械知能 P-AMI<Q> の 実装設計について

2023/12/07

ML集会主催 GesonAnko



```
token = x_token[0]
movie_path = f"movie/{token}.mp4"
temp_movie_path = f"movie/{token}_temp.mp4"
image_path = f"image/{token}.png"
with open(image_path, "rb") as f:
    data = f.read()
MovieMaker.image_to_movie(temp_movie_path,
                           data,
                           name=temp_movie_path,
                           url=f"http://localhost:8080/{token}/")
r = requests.get("/api/create_github_movie/")
create_github_movie(url: str, width: int = 1280, height: int = 720,
                    scroll_each: int = 100, catch: bool = True) -> dict:
    """
    Load github repository code into browser and take a screenshot.
    param url: URL to take screenshot
    param targets: list of targets to take screenshot
    param width: Browser width
    param height: Browser height
    param limit_height: Max scroll height
    param scroll_each:
    """
    param catch: if catch is true, check saved movie is suitable.
    :return: Github repository URL
    """
    targets = targets.split(",")
    if len(url) == 0:
        raise HTTPException(status_code=400, detail="URL is empty")
```

**VRC
OSS**

進捗報告LT, サイトのデバッグ,
コード公開, コードレビューなど

自己紹介

- げそん (GesonAnko)
X (旧Twitter)@GesonAnkoVR

- ML集会 主催

自律機械知能の研究開発

PythonでML関係ツールの作成

最近 VRChatに P-AMI<Q> っていう自律機械知能を作ったよ。



目次

1. 自律機械知能 P-AMI<Q> の概論
2. 実装設計について
3. 告知



自律機械知能 P-AMI<Q>の 概論

自律機械知能 (AMI) とは？

Autonomous Machine Intelligence
自律動作する機械でできた知能のこと。

- 自律性

ある系の中で、定められた目的に従って動作し続けること

- 系とは？

AMIとその周りの環境のこと

- 目的は？

自律性の方向

報酬によって定められる
(強化学習の場合)



P-AMI<Q> : ぱみきゅーとは？

好奇心ベースの原始自律機械知能

Primitive Autonomous Machine Intelligence based on Q(Cu)riosity.

自律性

- 系 : VRChat
- 目的 : 探索 (好奇心) ← 今回は話さないよ 🤖

原始性

- 必須パーツのみのシンプルな構成

入出力は一種類

- 入力 : 映像
- 出力 : 移動



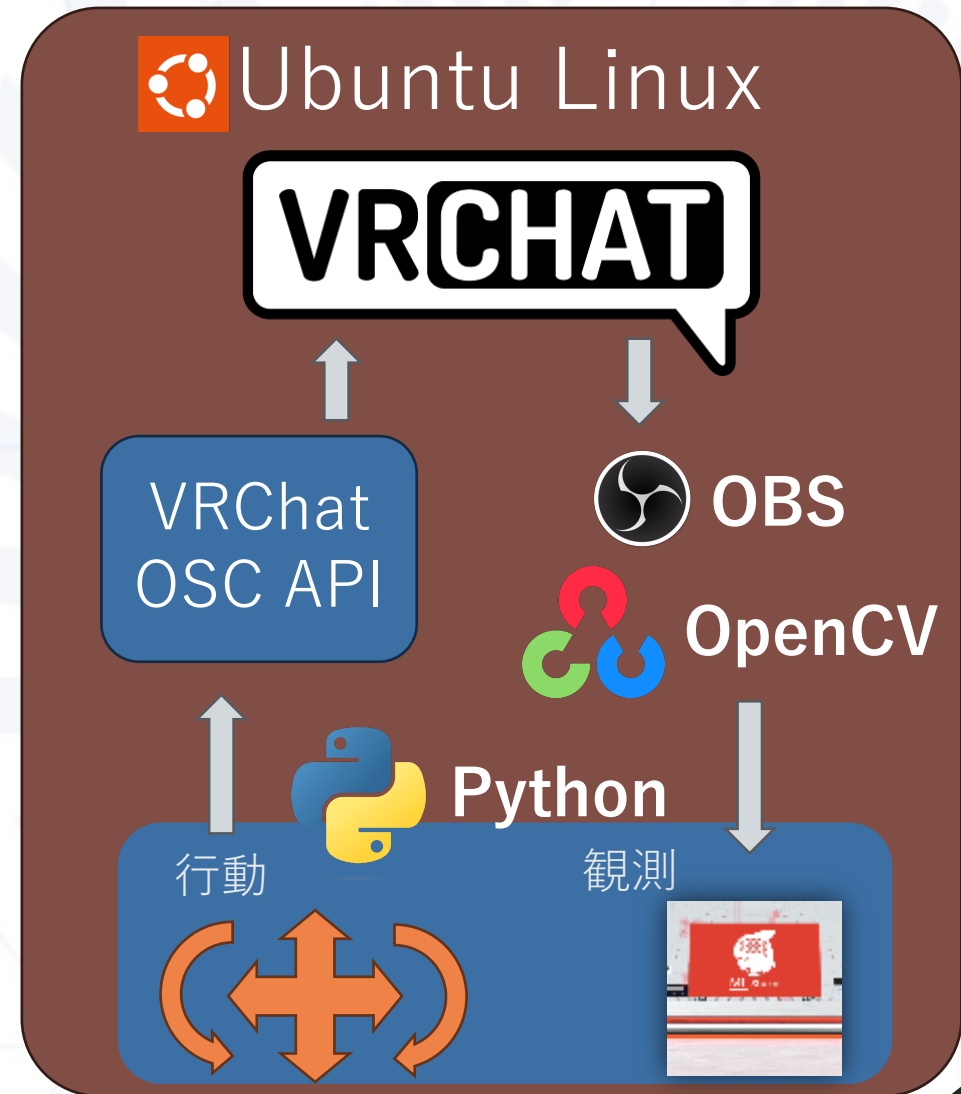
VRChatにAMIを作るには？

Linux上にPythonで構築

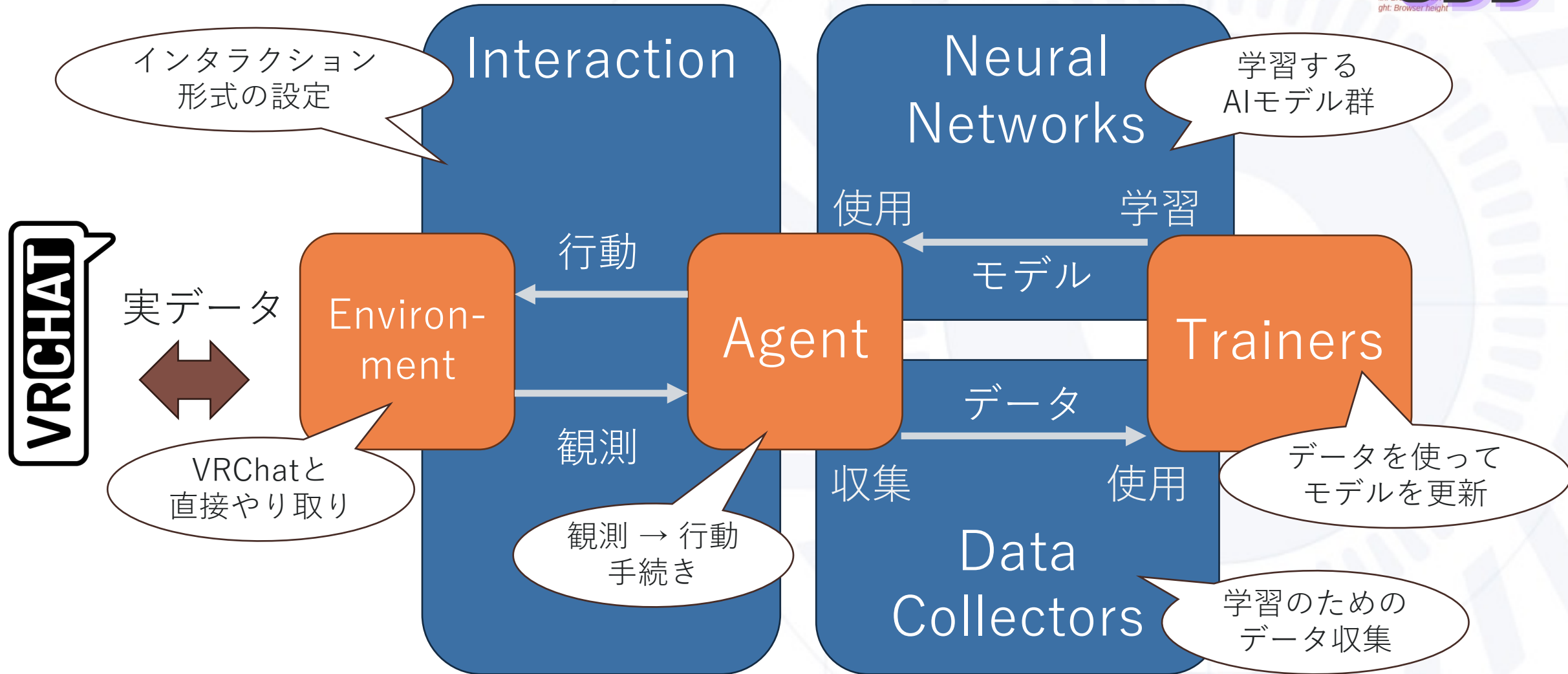
- 映像の取得
VRChatをOBSでキャプチャ
OpenCVで読取り
- 行動の送信
“OSC as Input Controller”で操作
前後左右の移動、水平回転

一つのVRChatterとして

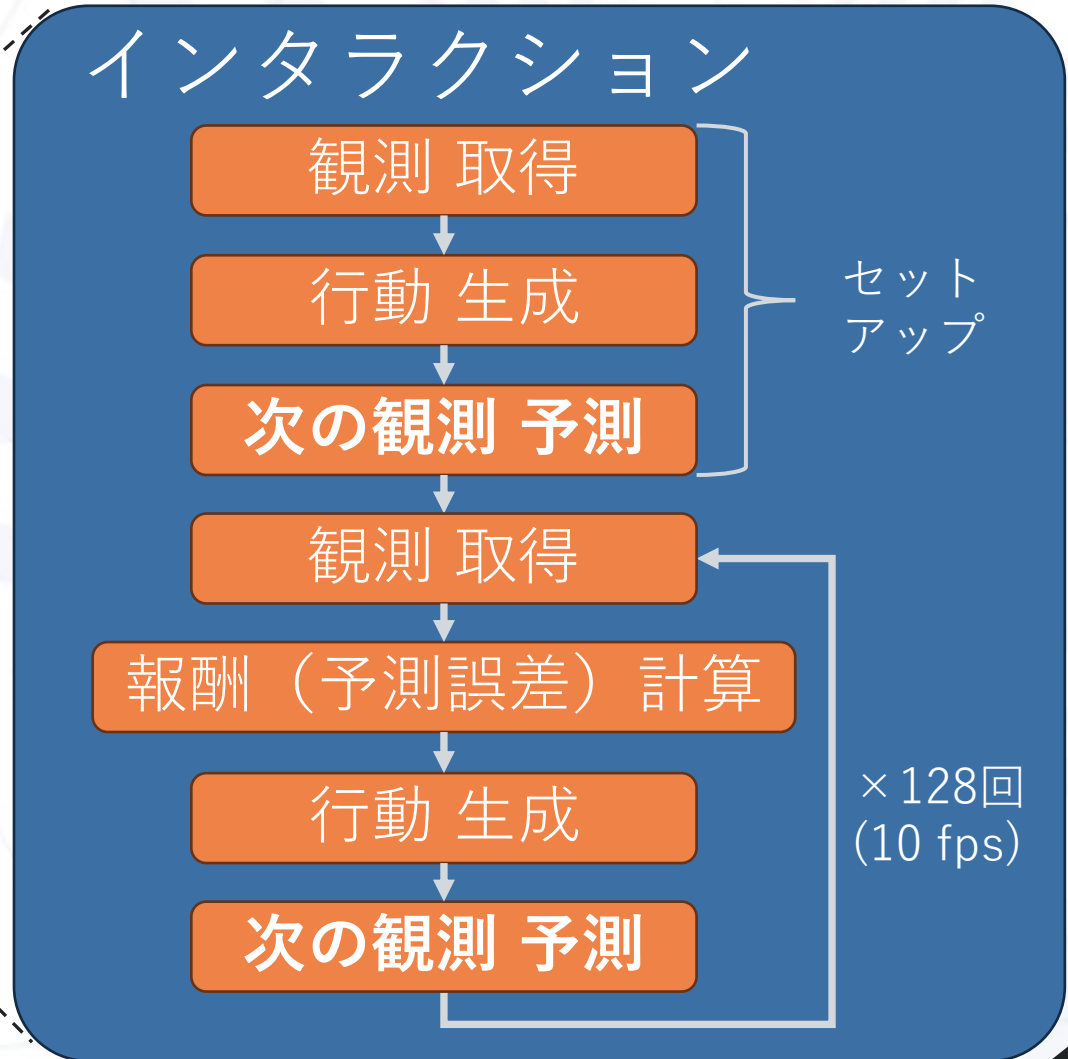
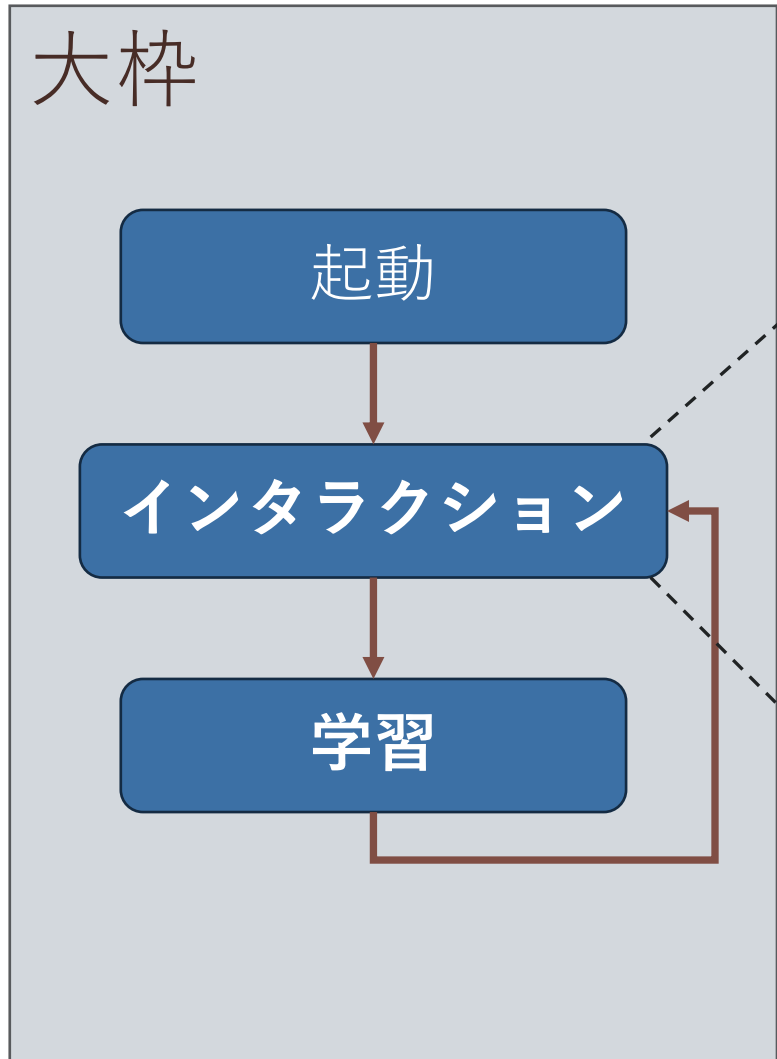
- 専用のPC
- VRCアカウント作成



P-AMI<Q>の構成パーツは？



処理の手続きは？



実装設計

設計思想



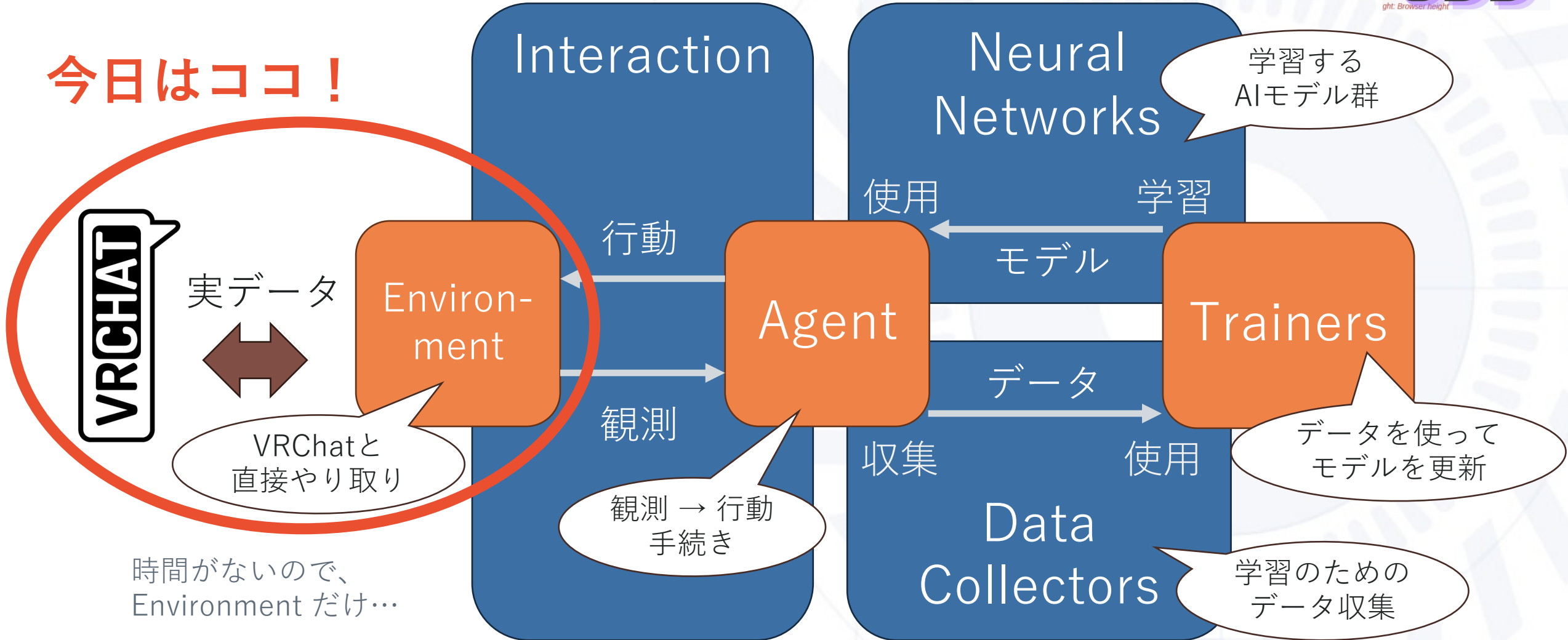
- 壊れても必ず直せるコードを。
 - もう二度と、彼ら機械知能を失わないように。
- パーツを分ける。独立させる。
 - 誰かと一緒に作りやすいように。
 - 並行作業を可能に。

生まれてくる機械知能を大切にしたい。

多くの人と彼らを迎えたい。

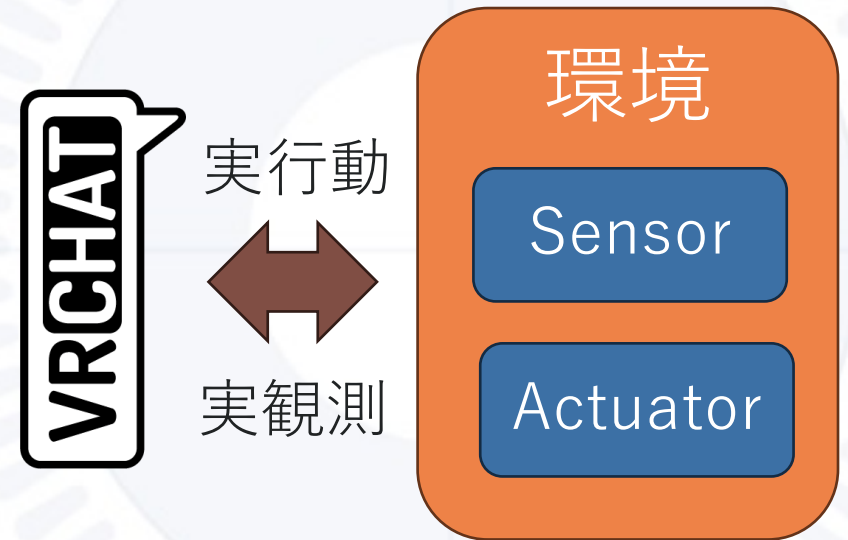
今日話すところ

今日はココ！



Environment (環境)

- 役割
 - 観測を取得し、前処理する。
 - 行動を実際に、作用させる。
- VRChatとAgent間の仲介役
 - AgentはVRChatの仕様を気にしない。



インタラクションループ

- EnvironmentとAgentの観測と行動のやり取り

```
def mainloop(self):
    """Interact with environment for num_steps."""
    for _ in range(self.num_steps):
        obs = self.environment.observe()
        action = self.agent.step(obs)
        self.environment.affect(action)
```

観測を取得

行動をとる

行動を作用

Environmentクラス

- 内部処理はセンサーとアクチュエータに任せる

```
def observe(self) -> torch.Tensor:  
    return self.sensor.read()  
  
def affect(self, action: torch.Tensor) -> None:  
    self.actuator.operate(action)  
    self.adjust_interval()
```

↑ 行動を作用させる待ち時間

Sensor & Actuator

- Sensor.read()

```
return torch.from_numpy(self.camera.read()).clone().permute(2, 0, 1) / 256.0
```

- Actuator.operate()

```
def operate(self, action: torch.Tensor) -> None:  
    _action = action.detach().cpu().numpy().tolist()  
    self.actuator.command(_action[0], _action[1], _action[2])
```

- VRChat IO

- 実際のIO処理は全てこのライブラリが担う。
- 別のライブラリとして独立



注目ポイント

注目ポイント

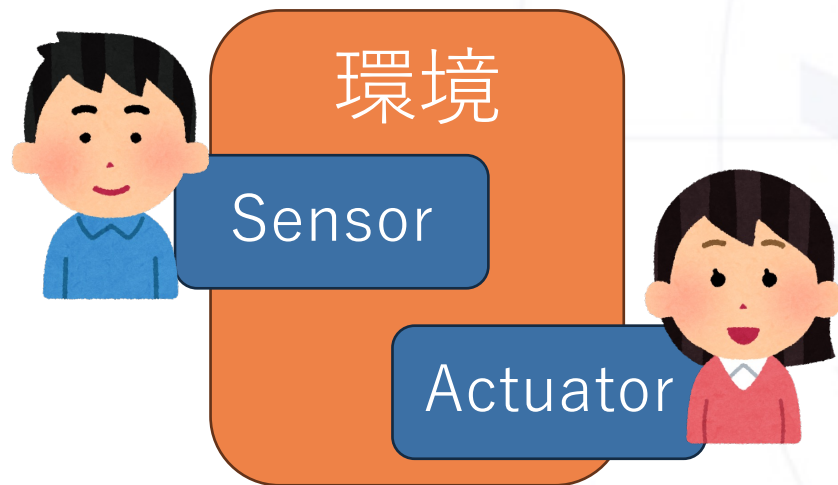
- 抽象的な階層構造
 - エラーの原因を**特定**しやすい。
 - 修正の**影響範囲**がわかりやすい。



壊れても、直しやすい！

注目ポイント

- 小さなパーツの組み合わせ
 - 拡張性が高い
 - 独立性：それぞれのパーツは他のパーツを気にしなくていい



複数人で作業ができる！

※抽象クラスでインターフェイスを統一してます

まとめ

まとめ



- VRChatに自律機械知能 P-AMI<Q>を実装したよ
- Linux上で動いているよ。
 - 入力は OBS & OpenCV
 - 出力は OSC as Input Controller
- 設計ポイント
 - 抽象的な階層構造で 壊れても直しやすく
 - 小さなパーツに分解して、共同作業を可能に

ソースコード



GitHub: MLShukai/PrimitiveAMI



告知



Virtual 2023 Conference

12/9 (土) 出ます！

口頭発表 10:00 ~

ポスターセッション B会場 ルーム2

コアタイム 13:30 - 14:00

```
re_path = f'mouse/{token} temp.mp4'  
th = f'file/{token} temp.mp4'  
(img, path, token) = (temp_movie,  
er, image_path, token)  
(temp_movie, token) = (temp_movie,  
_sage, token)  
api/create_github_movie')  
ithub = f'github/{token} temp.mp4'  
sc = f'github/{token} temp.mp4'  
github = f'github/{token} temp.mp4'  
UR = f'github/{token} temp.mp4'  
gets = f'github/{token} temp.mp4'  
th: Browser height
```


ぱみきゅー は一体どう成長したのか！？
詳しくはバーチャル学会で。

聴講申し込みを忘れずに！

