第 28 回日本バーチャルリアリティ学会大会論文集（2023 年 9 月）

ソーシャル VR プラットフォームにおける エージェント API の提案

倉井龍太郎 [1)], 平木剛史 [2]
Ryutaro KURAI , Takefumi HIRAKI

1) クラスター株式会社（〒 141-0031 東京都品川区西五反田 8-9-5, r.kurai@cluster.mu）
2) クラスター メタバース研究所（〒 141-0031 東京都品川区西五反田 8-9-5, t.hiraki@cluster.mu）

概要: 多数のユーザが空間を共有するソーシャル VR プラットフォーム上の自律的なエージェントを実装するには多様な情報が必要である。具体的には他ユーザとの距離、お互いの向き、アバタのポーズ、テキストや音声によるメッセージを認識し、それに応じた振る舞いが求められる。本稿では稼働中のソーシャル VR プラットフォームである cluster 上で上記のような情報を取得する API と API を利用したエージェントを提案する。

キーワード： ソーシャル VR プラットフォーム, エージェント, API

1. はじめに

人工物が人間と自律的に会話を繰り広げたり、人体のような身体の動作を表現することで、人間と見紛うようなコミュニケーションをとるという姿は、オートマトンやチューリングテスト [3] といった問題設定として従来からよく知られている。一方で、これらの問題は主に現実空間における存在を前提として設定されているが、バーチャル空間においては人間のように振る舞うソフトウェアに、空間の参加者であるユーザと同様の表現力を与えることが可能である。よって、バーチャル空間によって構成される VR 環境は、人工物が人間と区別のつかないエージェントとして活動するにあたり、現実空間よりも適した環境であると考えられる。

また、Unity、 Unreal Engine のようなゲームエンジンや HTC Vive、 Meta Quest のようなヘッドマウントディスプレイ (HMD) の発展と普及により、VR 環境における体験を設計、実装することは容易になってきている。そのため、すでに参加者が多くいる VR 環境で上述のようなエージェントを活動させることができれば、開発者・研究者にとって有望な実験フィールドとして利用できると考えられる。一方で、多人数が参加できる VR 環境を作るにはユーザ間の情報の同期といった技術的困難や、そもそも参加するユーザの絶対数を確保することの難しさが存在している。

ここで、前述の情報同期の問題などを解決した上で、すでに多人数が参加して交流を行っている VR 環境としてソーシャル VR プラットフォームが存在し、よく知られたものとして VRChat がある。VRChat は世界最規模の参加者を抱え、アバタの表現やバーチャル空間の構造について制約が少なく、実験フィールドとしての利用も可能なプラットフォームである。しかし、VRChat において、バーチャル空間内の情報をエージェントに伝えたり、 VRChat 内のアバタを VRChat の外から操作する API は提供されておらず、VRChat 以外のソーシャル VR プラットフォームについても現在のところ、上記のような API は普及していない状況である。

図 1: cluster 上でのエージェントとの会話の様子

そこで本研究では、多人数が参加し、かつアバタの表現力も高いソーシャル VR プラットフォームである cluster 上で、エージェントの活動を可能にする API を提案する。提案するエージェント API を利用することで、開発者は HTTP 通信についての最低限の前提知識でエージェントの実装を行うことができ、実装されたエージェントは数多くのユーザが利用している VR 空間で実際に動作することができる。また、API から呼び出して制御でき、アバタの操作入力などを実現できるエージェントソフトウェアについても合わせて用意することとした。このエージェントソフトウェアを経由して他の web サービスと接続することで、エージェントに多様なふるまいをさせることも可能である。

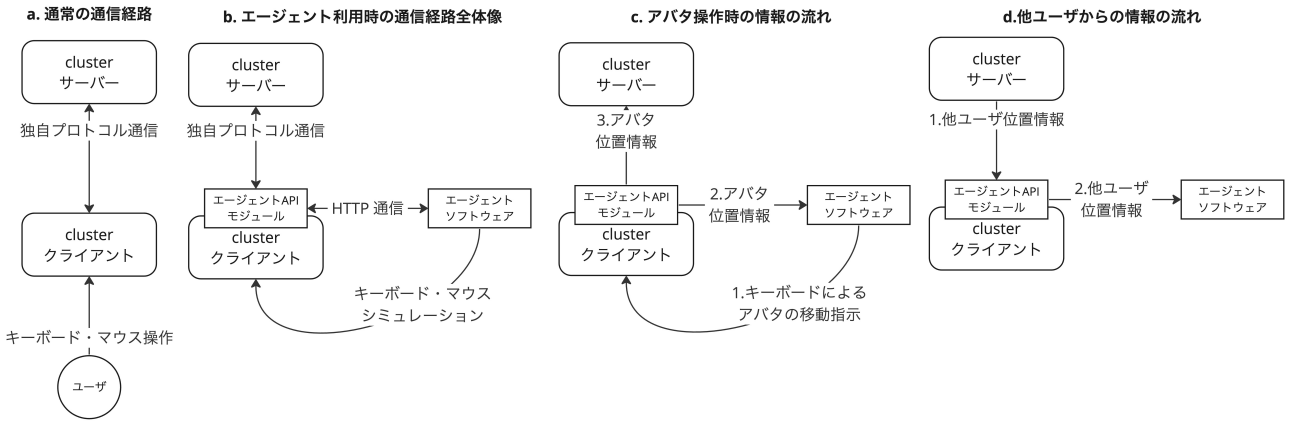

図 2: cluster において提案するエージェント API を用いた際の通信の模式図

本稿では、提案するエージェント API とエージェントソフトウェアについて設計、実装を行い、その動作を確認した。実際に cluster 上でエージェントが動作している様子を図 1 に示す。また、エージェント API の応用アプリケーションとして ChatGPT [2] と接続したエージェントを実装し、ソーシャル VR プラットフォーム上における会話エージェントの実現可能性を確認する。

2. 提案手法

本研究では、VR 環境として複数人が同じ VR 空間を共有し相互にコミュニケーションを行うことができるソーシャル VR プラットフォームにおいて動作するエージェント API を提案する。ソーシャル VR プラットフォームとしては、クラスター株式会社が運営する cluster を利用する。

VR 環境において動作するエージェントを実現するには、バーチャル空間内の情報をエージェントに伝える経路と、エージェントの動作についての指示をバーチャル空間内のアバタの動作として実現する経路について、それぞれ必要な情報を通信によってやり取りする必要である。以下、この経路と通信について用語を説明した後に説明する。

2.1 用語説明

まず本論文で使用する用語について説明する。説明する用語を用いた、VR 環境において動作するエージェント API の動作の模式図を図 2 に示す。

2.1.1 ユーザ

cluster を利用する人物のことをユーザと呼ぶ。ユーザは固有の ID をもち、希望するアバタを用いて、VR 空間内で他のユーザとコミュニケーションをとることが出来る。

2.1.2 cluster サーバ

VR 空間内のユーザ同士の動きを共有するための機能を担うソフトウェアのことを cluster サーバと呼び、クラスター株式会社が管理する計算機上で動作している。cluster クライアントからの通信によりアバタの動きを受け取り、他の cluster クライアントへその動きを共有することが主な役割である。

2.1.3 cluster クライアント

cluster の操作を行うソフトウェアで、Windows、OS X、Android、iOS など複数の環境で動作する。本稿で対象にするのは Windows と OS X で動作するクライアントである。cluster クライアントは、ユーザからのキーボード入力やマウス入力を受け取り、入力を cluster の空間内でのアバタの動きとして表現する。表現された動きは、他のユーザと共有するために cluster サーバに送信される。

2.1.4 コメント

言語によるコミュニケーションの方法としてユーザに提供されている機能である。キーボードからの入力により VR 空間内でテキストを共有することができる。テキストは同じ空間内にいるユーザすべてに共有される。

2.1.5 エモート

ユーザの感情を示すコミュニケーション方法として提供されている機能である。マウス操作により表現したいエモートを選ぶと、アバタが感情を表現するポーズを取ったり、感情を表現するハートマークのようなアイコンをアバタの直上に表示することが出来る。

2.1.6 エージェント API モジュール

本研究で提案する cluster クライアントに追加するソフトウェアを示す。cluster クライアントと cluster サーバの通信の間に入り、エージェントソフトウェアを作成するのに必要な API を提供する。

2.1.7 エージェントソフトウェア

エージェント API モジュールが用意する API を用いて、エージェントの動きを実装するソフトウェアを指す。具体的には他のユーザの動きを API モジュールから取得し、アバタによる応答を指示するソフトウェアである。エージェント API モジュールからエージェントソフトウェアへの通信が HTTP で実現されるため、エージェントソフトウェアは、HTTP サーバおよび HTTP クライアントとして実装される。

2.2 設計・実装

2.2.1 エージェント API モジュールによる通信の監視

まず、バーチャル空間内の情報をエージェントに伝える経路を検討する。バーチャル空間内の情報とは、同じ空間内にいる他のユーザの位置、ユーザが利用しているアバタの姿勢、他のユーザがコメントとして書き込んだ文字列、他のユーザが発した音声がこれにあたる。cluster の通常の利用であれば、これらの情報はレンダリングされた VR 空間の風景や文字列、スピーカーやヘッドフォンから流れる音声として表現される。しかし、エージェントの制御を考えると、これらの情報は直接分析が行える、空間の座標や音声データとして利用できることが望ましい。

そこで、これらの空間内の情報をエージェントに望ましい形で提供する API モジュールを開発した。このモジュールは図 2b のような形で cluster クライアントと cluster サーバの間に存在し、cluster クライアントが動作する PC で cluster クライアントと同時に動作する。このモジュールは、cluster クライアントと cluster サーバの間の通信を監視しエージェントにとって必要な情報を使いやすい形でエージェントソフトウェアに提供する。

cluster クライアントと cluster サーバ間の通信を監視する形でこのモジュールが作成した理由としては、cluster の機能は日々追加・変更が行われており、もし、cluster クライアントや cluster サーバの実装（本体側の実装と呼ぶ）と強く結合されたモジュールを用意してしまうと、そのモジュールは日々の変更に追従した更新が必要になることが挙げられる。一方で、cluster クライアントと cluster サーバ間での通信プロトコルは社内向けには定義が公開されており、また通信プロトコルの変更は頻度が少ないため本体側の変更にモジュールが追従するのが容易である。加えて、本体の開発やリリースと無関係に開発更新を行うことができるため、サービス開発と研究活動が競合することが少なくなることも大きな利点である。

このエージェント API モジュールが通信を監視し、エージェントソフトウェアに VR 空間内の情報を提供する手順を、位置情報を例に 3 つに分けて説明する。また、位置情報についてのデータの流れを例にとって説明するが、同様の方法で音声情報や、アバタの姿勢情報も更新、取得が可能である。

cluster クライアントを通じたアバタ操作

このシステム構成においてエージェントソフトウェアは cluster クライアントを経由してアバタの操作を行う (図 2c-1) 。一般のユーザが使う cluster クライアントと同一のソフトウェアを利用しているため、エージェントソフトウェアからの入力はマウス操作とキーボード操作のみが可能である。そのため、エージェントソフトウェアは、マウス操作、キーボード操作を Python からエミュレーションできるライブラリである PyWinCtl と PyAutoGUI を利用し、cluster クライアントを操作する設計とした。cluster ではキーボードの w、a、s、d キーがアバタの移動に割り当てられているので、このキーを操作することでアバタを自由な方向に移動させることができる。また、マウスのドラッグ操作でアバタの向きを変更することができるので、こちらも同様にマウス操作のエミュレーションを介して、アバタを好きな方向へ向かせることもできる。

表 1: エージェント API がエージェントソフトウェアにリクエストする内容

パス	送信されるデータの内容
/location	他のユーザのアバタ位置、 エージェントの操作するアバタ位置情報
/emote	他のユーザが表示したエモートの情報
/comment	他のユーザーが入力したコメントの内容が、 コメントのタイミングで POST される

cluster クライアントからサーバへの通信の監視

cluster クライアントを介して操作したアバタの動きは、アバタの位置情報としてサーバに送信される (図 2c-3)。この通信をエージェント API モジュールは監視しており、エージェントソフトウェアが操作したアバタがどの座標に移動したのか知ることができる (図 2c-2)。これは例えば、アバタを移動させたい目標座標があるときに、目標座標と、現在の座標の差分を計算することで、次の cluster クライアント操作の決定に利用できる。このクライアント操作は前述のエージェントソフトウェアを通じて行われるため、アバタの移動であれば前進、左右への回転、後退といった操作として決定する必要がある。

cluster サーバからのクライアントへの通信の監視

cluster サーバから cluster クライアントへの通信には、他のユーザの位置情報が含まれている。これをエージェント API モジュールは監視することで、他のユーザがどの座標に存在しているか知ることができる (図 2d-2)。こうして得た他のユーザの位置情報を利用することで、エージェントソフトウェアは操作するアバタを他のユーザの位置へ近づけたり、その方向へ向きを変えたりすることができる。

2.2.2 エージェント API モジュールから、エージェントソフトウェアへの通信

エージェント API はモジュールは、監視を行っている通信の内容の中から、エージェントの動作に必要な情報のみを選択し、利用しやすい形に変換しエージェントソフトウェアに送信している。具体的には、他のユーザの空間内の位置情報、コメント、エモートと呼ばれる感情表現である。エージェント API モジュールからエージェントソフトウェアへの HTTP リクエストの一覧と、詳細な内容を表 1 に示す。

エージェント API モジュールから、エージェントソフトウェアへの通信は HTTP を利用している。全ての HTTP リクエストは POST メソッドで行われ、必要なデータは JSON で paylod に格納されている。API モジュールからエージェントソフトウェアへの通信を HTTP としているの

は、エージェントソフトウェアを実装する開発者の学習コストを低減するためである。ここで、HTTP ではなく独自プロトコルを利用するなどの代替手段も考えられるが、HTTP サーバの実装は多くのプログラミング言語で用意されていることから、開発者は通信プロトコルの実装ではなく、エージェント制御の実装に集中することができる。また、エージェント API モジュールはクラスター社から提供され、将来的には開発者はエージェントソフトウェアのみの実装で様々な実験が可能になる状態を目指している。

2.2.3 エージェントソフトウェアからエージェント API モジュールへの通信

エージェントソフトウェアは、アバタを操作するために、マウスとキーボードのエミュレーションを用いて直接 cluster クライアントを操作する。しかし、コメントの入力や、エモートの選択をマウスとキーボードのエミュレーションを用いて行う事は複雑な操作になり、入力の不確実性が高まる。そこで、エージェント API モジュールは、コメントの入力と、エモートの入力を受け付ける機能を持っている。この機能は HTTP サーバとして実装されている。そのため、エージェントソフトウェアから、エージェント API モジュールへ HTTP でリクエストを送ることで、マウスやキーボードのエミュレーションなしに、コメントやエモートを cluster サーバへ送信できる。

上述したような設計のエージェント API モジュールとエージェントソフトウェアについて、Python を用いて実装を行った。

3. VR 環境におけるエージェントの実装と評価

提案したエージェント API モジュールとエージェントソフトウェアを用いて、cluster 上で他のユーザとのコミュニケーションが可能なエージェントを実際に実装した。コミュニケーションに必要な文面の生成や、アバタのコントールの一部に Large Lunguage Model (LLM) を用いることとし、LLM として OpenAI 社の gpt-3.5-turbo を利用した。エージェントソフトウェアは、エージェント API モジュールから送信されるユーザ位置情報の変化を計測することで、ユーザがエージェントの操作するアバタに接近していることを検知することができる。図 1 はユーザとエージェントのやり取りの様子をユーザの一人称視点画像で示しており、エージェントはユーザの接近に応じてコメント欄で挨拶を返すようにプログラムされている。

ここで、挨拶は定型文を用意しているのではなく、Open AI 社の API を利用し都度生成している。利用しているプロンプトの例を以下に示す。

- ”<ユーザ名>”という名前の人が近づいてきました。何か声をかけてください。
- ”<ユーザ名>”という名前の人が、”<ユーザのコメント >”と言いました。なるべく短く返事をしてください。
- ”<ユーザ名 >”という名前の人が、離れていきました。何か声をかけてください。
- ”<ユーザ名 >”という名前の人が、”<エモートの種類 >”という表情をしました。何か声をかけてください。

プロンプトの例にあるように、アバタの状況と、ユーザの距離を説明することで、適切な挨拶文が生成されることが確認できた。また、ユーザとアバタが近づいたり、離れたりした履歴を保存しているので、挨拶の文章もその行動を反映した文面になることがわかった。

会話の応答速度は、ユーザの体験に大きく影響することから、エージェントの会話応答に要する時間について計測を行った。エージェントへの質問文として、AI 王 公式配布データセット [1] のクイズから 110 問のクイズを用意した。このクイズをエージェントに繰り返しコメントにより質問し、質問のコメントを投稿した時刻から、エージェントからの回答がクライアントに届くまでの時刻を計測した。質問者側のクライアントの時刻で計測しているため、この計測時間にはインターネットを介した往復の通信時間も含まれている。結果、会話の応答に要した時間は平均 1.34 秒で、標準偏差は 0.27 秒であった。この応答時間から、実装したエージェントはコミュニケーションの応答としては十分な反応速度を有していると考えられ、ユーザにとって自然な会話になることが期待できる。

4. おわりに

本稿では、ユーザが自由に設計したエージェントが VR 環境の中で動作するようなシステムの実現を目指し、エージェント API の提案を行った。実際に大規模なユーザが実際に利用しているソーシャル VR サービスである cluster において、エージェント API モジュールを設計、実装した。また、実際に API を用いて VR 環境で動作するエージェントの実装を行い、LLM を利用することで、ユーザと自然なコミュニケーションが可能であることが示された。

今後の展望として、API モジュールとソフトウェアのオープン化が挙げられる。現状の実装においては、クラスター社にとってのセキュリティリスクを鑑みて広く公開することができていない。クラスター社との簡易契約によって、多くの研究者がこの API を利用できるような環境をこれから構築していく必要がある。また、サービスの一部として広くユーザに届けるためには、エージェントソフトウェアが cluster クライアントを操作する形ではなく、cluster サーバ側にエージェントが存在するような形も必要になると考えられるため、そのような実装形態についても検討していく。

参考文献

[1] abc/EQIDEN 実行委員会. AI 王 公式配布データセット. https://sites.google.com/view/project-aio/dataset.

[2] OpenAI. ChatGPT (Mar 14 version) [Large language model]., 2023. https://chat.openai.com.

[3] A. M. Turing. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 10 1950.