自己 PR 資料

1　はじめに

私は、中学 1 年生の頃にプログラミングに興味を持ち、インターネットの情報を調べつつ、独学で開発を続けてきました。その中で、2025 年度に独立行政法人情報処理推進機構(IPA)が実施する「未踏事業」の修了生によって運営されている、小中高生クリエータ支援プログラムである未踏ジュニアでスーパークリエータ認定を受け、また同年度に国立研究開発法人情報通信研究機構(NICT)が運営する、25 歳以下を対象とし、1 年間を通してものづくりをする若手技術者育成プログラムである SecHack365 に採択され、開発駆動コース仲山ゼミを修了しました。また、今年の 6 月からは、サイボウズ・ラボ株式会社が運営する若手技術者育成プログラムであるサイボウズ・ラボユースにも採択され、来年の 6 月までラボユース生として活動をします。

2　これまでの取り組み

私は 2021 年ごろ、中学 1 年生のときに HTML/JavaScript を独学で学び始め、小さな Web サイトをいくつか作成しました。2022 年には IP アドレスから地域を検索するツール「nodegeoip」や、VTuber のファンサイトなど、Web アプリケーションの制作を続けました。2023 年ごろからは React などのフレームワークを習得し、物体検出モデルの WebUI や読み上げアプリなど、より実用的なアプリケーションを開発するようになりました。

2024 年には、SkyWay[1]と MediaPipe[2]を組み合わせ、VTuber 配信者が手軽に 3D コラボ配信を行えるアプリケーション「VRMeet」[3]を開発しました。このアプリを Joyo High School テックコンテストで発表し、「テック特別賞」を受賞しました。

同年末には分散型 SNS に強い関心を持ち、Bluesky などの ATProtocol を用いたアプリケーションを複数制作しました。絵文字リアクションを分散して実装した「Stellar」[4]は多数のユーザーに使用していただき、分散型 SNS の動向を追うニュースサイト The Fediverse Report[5]に掲載されました。この経験が、のちの polka の開発へつながっています。

2025 年には未踏ジュニア・SecHack365 への採択に加え、aikyo で培った AI キャラクター技術を活用した観光案内アプリ「HoloTrip」を開発し、再度 Joyo High School テックコンテストで「テック特別賞」を受賞しました。また、テックコンテストの審査員を務めた IBM の執行役員の方からお誘いをいただき、虎ノ門の日本 IBM 本社で HoloTrip に関するプレゼンテーションを行う機会もいただきました。

---

[1] WebRTC のラッパーライブラリ
[2] Google が提供するリアルタイム顔・骨格検出ライブラリ
[3] https://github.com/marukun712/VRMeet
[4] https://github.com/marukun712/stellar
[5] https://fediversereport.com/last-week-in-the-atmosphere-2025jan-d/

3　SecHack365 での取り組み

SecHack365 ではpolka[6]という分散プロトコルを開発していました。このプロトコルの開発のきっかけは、数年前に私の Twitter アカウントが誤って凍結されたことから始まります。この時は、数週間ほどでアカウントが復活して事なきを得ましたが、このことをきっかけに単一のプラットフォームにデータを預けること、SNS というインターネットの公共空間が単一企業に管理されていることの脆弱さを感じるようになりました。

その後、私は少数の企業が中央集権的にユーザーの情報を管理する、プラットフォームの課題について考えるようになり、その中で以下のように考えました。

1. **アカウント BAN**

   単一のプラットフォームがデジタル上でのその人の活動を制限することができてしまいます。異議申し立て手段はありますが、これも単一の企業の判断に依存してしまいます。また、文化も言語もことなる言論空間を、単一企業の価値観だけで裁くことは本当に可能なのでしょうか?

2. **サービス終了**

   企業の経営判断などでサービスが終了される場合もあります。その場合、ユーザーのデータは一方的にすべてが失われ、ユーザー側にも対策手段がありません。

3. **API 制限**

   API としてサービスのデータを公開することで、オープンなサービス同士の相互作用をすることができます。しかし、このオープンな窓口はやはり単一企業の判断によって閉じることができてしまいます。そのサービスを中心にエコシステムが発達したとしても、一判断ですべてが利用不能になってしまうのです。

私は、これら 3 つの課題はすべて、「データとサービスが一体化している」ことに要因があるのではないかと考えました。すなわち、データを集めて使用するだけのサービス側が、すべてのデータを中央集権的に管理しているのです。この課題を解決するには、サービスとデータを分離する必要があります。つまり、データはユーザーの手元に、サービス提供者はそのデータを集計・表示するだけの存在ということです。Bluesky などで使用されている ATProtocol は、この考え方を実践しています。 ですが、ATProtocol は Twitter を代替するような世界規模のネットワークを構築するため、ある程度の中央集権性がのこされています。つまり、ネットワークの規模を大きくし、そのネットワークを横断した検索性を確保しようとすると、どうしても中央集権的な要素が必要になってしまうのです。

[6] https://github.com/marukun712/polka

ですが、私はここで 1 つの疑問を感じました。SNS 空間は本当にグローバルである必要があるのでしょうか? 自分の SNS の利用の仕方をふりかえってみると、決まった話題・決まったユーザーの投稿を閲覧していることが多く、全く違う文脈の投稿を見るシーンはかなり低い確率です。

つまり、本当に必要なものは、自分の興味範囲内にあるデータを自分の手元へ賢くリンクし、一方それ以外のデータは確率的に取得できるというローカルなソーシャルネットワークなのではないかと考えました。

この発想の転換から、polka は生まれました。polka では、SNS の投稿をより厳密に、タグ/投稿の階層構造のあるグラフとして表現します。 投稿全体をグラフとしてとらえることで、他ユーザのノードまたは部分グラフを中央サーバーを介さず自分のグラフに直接リンクし、自分のソーシャルネットワークを拡大していくことができます。

技術的にはまず、データ構造として ATProtocol の Merkle Search Tree[7]をドキュメントデータベースとしてラップしたデータベースを使用します。 さらに、did:web[8]やデータベースの静的ファイル配信といった、Web の基本的な概念のみを使用することで、ポータビリティかつ検証可能性のあるプロトコルを実現しています。

このように、polka では直接的なリンクを用いるため、探索のための中央集権的なサーバーなしで、情報の探索ができます。

この polka を用いることで、次のようなことが可能になります。たとえば技術の情報に関心のある Alice さんがいるとします。その Alice さんが Bob さんのプロフィールを見て、Bob さんがリンクしている Carol さんの投稿からさらに技術に関する投稿を探索することができます。Carol さんも技術に関心があるので、Alice さんは自分の関心に沿って情報を収集することができます。しかも、Alice さんの情報収集のプロセスを管理する管理者はいません。既存 Web の直線的なリンクでは探索できる選択肢が限られますが、polka ではグラフ構造を用いることで多角的に情報を探索していくことができるのです。また、類似した投稿などがグラフ上で一目でわかるというメリットもあります。

こうしたメリットを持つ polka の開発に力を入れてきました。

[7] Merkle Tree をベースにした、効率的な検索と署名検証が可能なデータ構造
[8] ドメイン名を分散識別子(DID)として利用する規格

4　未踏ジュニアでの取り組み

以上の分散プロトコルの開発に加え、私はデジタルキャラクター、特に LLM を活用した AI キャラクターの領域にも興味があります。 未踏ジュニアでは、aikyo[9]という AI エージェントの相互作用を実現するためのフレームワークを開発しました。 私はもともとアニメキャラクターと会話/生活してみたいという夢を持っていて、LLM や XR 技術に大きな可能性を感じました。はじめは、MRTalk という MR 空間で AI キャラクターと会話するためのアプリケーション[10]を 1 年ほど開発し、そのテーマを未踏ジュニアに応募し採択されました。

この未踏ジュニアのプログラムを通して、MR キャラクターの構築という単一の体験を提供することだけでなく、だれもが自分好みの AI キャラクターを開発できる基盤の制作に興味を持ち、aikyo を開発しました。既存の AI エージェントは、特定のタスクをこなす補助的な役割として設計されていますが、ただ人間らしく会話する、人間のようにふるまうだけの自律的エージェントが存在してもいいのではないか、という考えが aikyo の根幹です。

aikyo では、libp2p[11]を用いたトランスポート非依存の Gossipsub[12]によって、AI エージェント同士の自律的なコミュニケーションを実現しています。特に注力したのは、会話分析における「隣接ペア」の概念をメッセージスキーマに取り入れた点です。隣接ペアとは質問 -> 応答、挨拶 -> 挨拶のように発話が対になって成立する構造のことで、これをスキーマレベルで定義することで、より自然なターンテイキングを実現しています。このアイデア自体は先行研究「Who speaks next?」[13]に基づくものですが、先行研究では単一プロセス内の単純なループとして実装されていたものを、libp2p の Gossipsub を介したネットワーク越しの通信に発展させることで、複数のエージェントが物理的に異なるマシン上で自律的に動作できる分散構成を実現しました。この取り組みは先行研究の著者である宇都宮大学の森先生からも意義深いとのフィードバックをいただいています。また、「Opening Up Closings」[14]に基づき、会話の終了プロセスを `none`・`pre-closing`・`closing`・`terminal` の 4 状態で表現する実装も行いました。これにより、エージェントが自然なタイミングで会話を締めくくれるようになっています。加えて、CEL[15]のような式評価言語を組み込み、エージェントの挙動をプログラマティックに制御できる設計としています。

---

[9] https://github.com/marukun712/aikyo
[10] https://gameparade.creators-guild.com/works/2626
[11] 分散 P2P アプリケーション向けのネットワークライブラリ群
[12] libp2p 上に実装された pub/sub メッセージングプロトコル
[13] https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1582287/full
[14] https://web.stanford.edu/~eckert/Courses/l1562018/Readings/SchegloffSacks1973.pdf
[15] Google が開発した軽量な式評価言語(Common Expression Language)

結果として、異なるキャラクター設定を持つ AI エージェント同士が人間の介在なしに自律会話するデモを実現しました。実際に触れてもらった知人からは「自分好みの AI キャラクターを手軽に作れる」という反応を得ました。

aikyo[16]は、エンジニアが自律的な AI キャラクターを構築する際の共通基盤となることを目指しています。これまで AI キャラクターの実装手法は開発者ごとにバラバラで、確立されたアプローチが存在しませんでした。aikyo はその知見を一つのフレームワークとして整理することで、エンジニアがすでにある技術を再び開発することなく、キャラクター設計そのものに集中できる環境を提供します。

以上のように、私は polka、aikyo の開発を通して、単なるアプリケーションの開発だけではなく、その基盤となるプロトコルレイヤーの実装に興味を持ち、分散システム・AIキャラクターなど様々なテーマを横断的に開発を進めました。

[16]https://jr.mitou.org/projects/2025/aikyo

5　現在の取り組みと今後の方針

これまで述べてきた私の取り組みには、二つの異なる動機があります。一つは分散システムへの純粋な技術的興味、もう一つはアニメやキャラクターが好きという気持ちから来るエンタメ方面への関心です。SecHack365 と未踏ジュニアを同時期に進められたのも、この二本が互いのリフレッシュになり、どちらかが詰まったときにもう一方に切り替えることで継続できたからだと思っています。この二本を持つことが、自分の強みだと考えています。

分散システムへの関心は、初めて ATProtocol のアーキテクチャを見て、その美しさや洗練さに感動したことがきっかけです。私は、技術的・思想的な脆弱性のない安定した分散システム基盤を築きていきたいです。その関心に基づき、現在は polka のアーキテクチャを根本から見直した「Polka Protocol」の仕様策定・実装に取り組んでいます。

polka の実装を完成させた後、私はこのプロトコルが本当に解くべき問いの核心に気づきました。それは「鍵の信頼問題」です。公開鍵暗号において、ある鍵が本当に特定の人物のものだと証明するためには、認証局(CA)や信頼グラフといった追加の信頼レイヤーが必要になります。これは、デジタル上で人を指し示すことの本質的な難しさです。

PGP のキーパーティーはこの問題に対するシンプルな答えです。物理的に人が集まり、直接鍵を交換することで、中間の信頼機関なしに「この鍵はこの人のもの」という確認が成立します。Polka Protocol は、このキーパーティー的な概念を、デジタル空間における揮発性の同期コミュニケーション、すなわち音声通話のような「場」としてプロトコルレベルで標準化します。

音声通話のような場では、人のアイデンティティはその場にいる他者からの観測によって自然に形成されます。ここに識別子も認証局も必要ありません。Polka Protocol はこの仕組みをプロトコルレベルで明文化することで、鍵の信頼のために必要になるはずだった明示的な信頼レイヤーを、人との実際の交流に委譲します。

また、プロトコル全体を通じてシンプルさを最優先にしています。文書ネットワーク部分には Gemini Protocol[17]のシンプルなドキュメントフォーマットと Noise Protocol Framework[18]を採用し、公開範囲のコントロールは WireGuard[19]などのネットワークレイヤーに委譲することで、プロトコル自体の責任範囲を最小限に絞っています。

最新の Polka Protocol 仕様は、こちらで公開しています。[20]

[17] HTTP の複雑さを排し、テキスト文書の配信に特化して設計された軽量なアプリケーション層プロトコル
[18] TLS より設計が単純な暗号化プロトコル
[19] オープンソースの VPN プロトコル
[20] https://github.com/marukun712/asakusa/tree/main/spec

一方、エンタメ方面への関心は、アニメキャラクターへの関心が元になっており、そのアニメキャラクターが存在している二次元の世界と現実の世界をつなげるブリッジ的な技術を開発することに関心があります。

その関心に基づき、現在はサイボウズ・ラボユースにて「kokoro/rig」[21]という 2Dキャラクターのメッシュ変形ライブラリを開発しています。Live2D のような GUI ベースのツールとは異なり、`(u, v) => 変形量` のような関数でキャラクターの変形を記述できる設計です。Web フロントエンドエンジニアが TypeScript の資産を活かして直感的に 2D キャラクターを動かせることを目指しています。aikyo のような AI キャラクターフレームワークと組み合わせることで、フロントエンドからバックエンドまでコードで一貫して定義できる AI キャラクター基盤の実現につながると考えています。

Live2D のような GUI ベースのツールは、Free-Form Deformation のような技術を用いて、ユーザーがキャラクターのメッシュ頂点を明示的に編集することで、変形を表現していました。しかし、この作業には変形量を手で動かすコツを掴む必要があり、専用のスキルが必要になります。kokoro/rig は、この GUI による手作業の編集ではなく、エンジニアにとってより直感的な関数型モデルを使用します。

このライブラリにおいて、キャラクターを動かすために必要な概念は一つだけで、`(u, v) => 変形`という関数のみです。この関数をポーズ関数と呼びます。

引数として取られている(u, v)は、変形対象の頂点の UV 座標、つまり左上を 0,0、右下を(1,1)とした座標です。

変形量としては、x,y だけでなく `rot(回転)`や `pivot(回転の中心)`も指定できます。

例えば、体を左向きに傾ける `left` というポーズは、以下のような関数で定義されます。

```
left: (u: number, v: number) => {
    const { fromTop } = getSpatialParams(u, v);
    const w = curve.power2(fromTop);

    return {
        tx: -200 * w,
        ty: 0,
    };
},
```

UV 座標が引数として取れることで、相対的な頂点の位置に基づいた重み付けをすることができます。 この実装例で言えば、UV 座標から算出される `fromTop`(対象となる画像の一番上を 1, 一番下を 0 とした値)を 2 乗の曲線に渡し、その値を移動量`-200` にかけることで、上半身ほど大きく動き、下半身はあまり動かないという自然な動きをたった 2 行で記述できます。

[21]https://github.com/marukun712/kokoro

もし、胴体や胸だけを動かしたい場合は、sin などの山形の曲線を用いて、同じように UV 座標から重みを求めればよいです。

さらに、このライブラリでは視差の考え方も導入しています。

この基本的な概念を用いて、キャラクターを斜めに傾けたり、一部分を移動させる、といったことは可能ですが、立体表現をするために必要な複雑な変形を表現する曲線の組み合わせを探すことは困難です。

ここで視差、つまり手前のものは大きく動き、遠くのものはあまり動かないという概念を導入し、UV 座標から曲線を使って上半身ほど大きく動く動きを表現したように、UV 座標から手前/奥という情報を判別すれば、移動量をスケールして頂点単位の視差を作り出すことができると考えました。そのため、ブラウザで動作する Depth Anything V2 をライブラリに組み込み、立ち絵から自動で深度情報を推定、UV 座標から手前/奥という情報を判別できる仕組みを実装しました。

実際にキャラクターが立体的に動くデモは、こちらで公開しています。[22]

私は、これらのプロトコルやライブラリを実際に多くの人が使えるものとして社会に送り出すことを目指しています。Stellar の開発では、約 100 人のユーザーからバグ報告やフィードバックをいただいた経験を通して、実際に人に使われることで初めて技術が意味を持つということを実感しました。今後も、技術的な正しさと実用性の両方を意識しながら開発を進めていきたいと考えています。

[22] https://marukun712.github.io/kokoro/depth.html

6　問いを立て直す力と実行力

以上が、現在の関心と今後の方針です。

ここまでの、polka、aikyo、その後の取り組みに共通する私の強みは、「問いを立て直す力」と「実行力」です。polka の開発中には一度立ち止まって根本的な課題を問い直し、aikyo の開発中には迷走しながらも試作と実行を繰り返し、その中で新たな気づきを得ることができました。この 2 つの強みのおかげで、未踏ジュニア スーパークリエータの受賞や、SecHack365 の修了を達成できたと感じています。私は、周囲から「アイデアを実装に落とし込む発想力と速さ」を評価していただくことが多く、発想力には「問いを立て直す力」が、速さには「実行力」が大きく関係していると考えています。

私は、大学に入ってすぐに研究を開始したいと考えています。同じような姿勢を持つ、未踏ジュニアや SecHack で関わってきたような学生と共に活発な議論を重ねることで、この強みをさらに磨いていきたいです。この「問いを立て直す力」と「実行力」を生かして、私は貴学から特定の組織に依存しすぎない分散的かつ自律的で、人間か AI かを問わず多様な存在が参加できるインターネットの基盤を生み出していきたいです。