未踏の第29期スーパークリエータたち：編集にあたって

情報処理学会・学会誌「情報処理」
2023年9月15日 08:30

竹内郁雄（IPA未踏IT人材発掘・育成事業　統括プロジェクトマネージャ）

未踏事業で採択され，優れた成果や成長を示した人たちを未踏スーパークリエータと呼ぶ．実は数年前から，未踏事業は3つの事業に拡大しており，ここで「未踏事業」と呼んでいるものは正確には「未踏IT人材発掘・育成事業」という名称で，年度始めに25歳未満の若い人たちを対象とした事業である．

未踏事業は2000年度にミレニアム事業として始まったが，この年次報告は，2012年度から「未踏IT人材発掘・育成事業」（以下，この事業だけを指すときは，「未踏IT」と略記する）に採択され，スーパークリエータとして認定された若い人たちの業績や人となりを紹介するものである．日本の若い才能溢れるIT人材の元気さや多様性を広く産業界や学界に知っていただきたい．若い彼・彼女らは，必ずや日本のITを活性化してくれるものと信じている．実際，20年以上の歴史を持つ「未踏」はすでにブランドを確立しており，2023年度からはビジネス化や社会的に意義の高いプロジェクトを実施する「未踏アドバンスト事業」が大幅に増強された．この勢いはさらに続く予定である．

この記事では若いIT人材を育む「未踏IT」で素晴らしい成果を出したクリエータだけを紹介するが，これからの時代を担う若い人材に刺激を与えたいからである．

2022年度の第29期未踏クリエータは計37名（21プロジェクト）で，そのうちの25名（15プロジェクト）がスーパークリエータとして認定された．2014年から認定率は右肩上がりに増え続けてきて，28期は50%と一服

した印象だったが，今年はまた67.6%に戻った．スーパークリエータの認定は相対基準ではなく，絶対基準であることをお断りしておく．そんなに多くといいのかという声も出そうだが，この記事で紹介された彼・彼女らを見ると首肯いていただけるだろう．

今年度，特に印象的だったのは，過去にも未踏に応募したが採択されず，今期採択されてスーパークリエータに認定された人の多さである．具体的には，皆川達也（5回目），内田郁真，スコット・アトム（3回目），三林亮太（3回目），阿部優樹，辻口 輝（2回目）竹村大希（2回目）の7氏である．過去と同じようなテーマの人もいれば，まるで違うテーマで応募してきた人もいる．

未踏では伝統的に不採択の人にも，丁寧な不採択理由を送るようにしている．これが捲土重来の，さらに先鋭度を増した提案につながっているのではないかと自負している．

いつものことであるが，今期も低レイヤからWebアプリまで幅広くバランスよくスーパークリエータが選ばれた．未踏の最大の特徴の1つである多様性の面目躍如である．

2022年度の未踏ITもコロナ禍の影響を受けたが，ハイブリッドな会議や会合が増え，常態に少しずつ戻ってきた感がある．コロナ禍の副作用か，クリエータたちはもうリモート開発にすっかり慣れてきたようだ．

この紹介記事は．情報処理学会誌がWeb化への大きな転回をした2021年度から，担当PMにスーパークリエータの紹介をWeb記事として書いていただき，この導入記事からは，そこへのリンクを貼ることにした．それぞれの紹介には短い統括PM追記として，お邪魔かもしれないが，私のほうで少しエピソード的な情報を追加している．

リンクの紹介は，これまでに倣い，代表者であるクリエータ名の50音順とする．タイトルは正式なものではなく，読みやすく憶えやすい「名は体を表す」キャッチに変えてもらった．なお，2023年2月18〜19日の2日間にハイブリッドで開催された成果報告会 （Demo Day） のすべての動画はIPA channel
https://www.youtube.com/user/ipajp/
で見ることができる．最近のプロジェクトはデモなど，動画で見ないと面白さや意義が分からないものが多いので，興味を持たれた方，スーパークリエータ以外の発表にも関心がある方は，つまみ食い的に見ることもできるので，ぜひそれをご覧いただきたい．特に，今期の山形・麻プロジェクトは瞬間芸イノチのフィギュアスケートの練習支援なので，動画を見ないことには始まらないと思う．

この記事ではスーパークリエータだけを紹介しているが，スーパークリエータ認定に洩れた人のプレゼンにも興味深いものが多い．それは未踏修了生のその後の活躍を見ていても同様である．

ぜひ未踏修了生の今後の活躍に注目したいただきたい．

（2023年7月3日受付）
（2023年9月15日note公開）

■ 饗庭 陽月
ハードウェアを意識しない組込み開発環境

■ 阿部 優樹，辻口 輝

祭り運営を支援するシステム temaneki

■ 飯田 圭祐，柚山 大哉
複数のARMマシンを1つに集約するハードウェア仮想化レイヤ

■ 井阪 友哉
HDCアクセラレータ「HPU」

■ 伊藤 謙太朗
直和型の代わりにユニオン型を持つ関数型言語
Cotton

■ 稲葉 皓信
レイアウトの自由度とキー操作性を両立したノートテイキング「鍵記」

■ 内田 郁真，スコット アトム
サッカー試合映像の検索・分析システムTASC

■ 大神 卓也，奈良 亮耶，天野 克敏，今宿 祐希
麻雀プロのためのAI牌譜解析ツール 極

■ 栗本 知輝，黒木 琢央，松田 響生
VRと電動トレーニング機器を用いた筋力トレーニングシステム TRAVE

■ 島元 諒
UVプリンタを用いたラインストーン造形システム

■ 蘇 子雄，方 詩涛
スマートフォン向けにカスタマイズが可能なサイレントスピーチインタフェース

■ 竹村 太希
翻訳IME「Konjac」とInput Method抽象化レイヤ

■ 皆川 達也
抜かない型の設計支援ツールKatalystによるものづくりの自在化

■ 三林 亮太

ラップバトル対話システム

■ **山形 昌弘，麻 大輔**

フィギュアスケートの練習を支援するSkate Jump Board

ハードウェアを意識しない組込み開発環境

情報処理学会・学会誌「情報処理」
2023年9月15日 07:53

饗庭 陽月（あいば ひづき）

センサ・アクチュエータといったハードウェアを用いる組込みシステムの開発を，さもWeb向けソフトウェアの開発のように行うことを可能にするmCnを開発・提供した（**図-1**）．饗庭さんの言葉を借りれば「ハードウェアの無意識化」，「ハードウェアのAPI化」である．

図-1　mCnのアイディア

ハードウェアとしては，センサやアクチュエータを搭載したモジュール群（**図-2**）と，モジュールをいくつか搭載できるベースを開発した．ベースにモジュールを搭載し，それを，スマホやタブレットに接続して用いる．スマホ・タブレット向けにソフトウェアを開発する場合，どのモジュールが相手でも，JSON形式でデータを受

信，指示を送信できる．これは簡単だ．また，mCnを用いた応用をいくつか開発した（**図-3**）．

図-2　モジュールのラインナップ

図-3　料理に適用した例

この形のmCnに辿り着くまでには，紆余曲折あった．当初は，スマホを使うのではなく，マイコンを搭載したモジュールを頭脳とする構成を採っていた．2022年7月に開催された，全プロジェクトに加えてOB・OGも参加するブースト会議にて，どなたかからいただいた「そういう応用なら，マイコンではなくて，スマホやタブレットを使いたいよね．画面もタッチパネルもあるし」というコメントによって，饗庭さんは大胆に方針を変え，スマホを拡張するハードウェアの開発に舵を切った．担当PMの首藤としては，mCnが数多あるセンサモジュールのうちの亜流になって個性が薄れる，という心配をしたが，より有用なものになる方針転換だったので，賛成した．10月になると，ベースといくつかのモジュールができてきた．饗庭さんはすごい開発成果でもサラッと紹介するので，あまりすごいように聞こえないのだが，2月の成果報告会ではプレゼンも迫力あるものとなって，成果物の真価が多くの方に伝わった．

饗庭さんは実は，mCn対応スマホの開発も進めていた！　ベースを内蔵したスマホXphoneである．スマホに接続する通常のベースには，USBポートの給電能力の限界から，モジュールは2つまでという制限がある．そこで，より多くの，4つのモジュールを使用できる専用スマホを開発しようとしたのである．Raspberry Pi CM4を中心に，必要な周辺回路を独自設計した．優先順位を鑑みて，開発期間中に行ったのは基盤の設計までとしたが，饗庭さんの，招来したい未来をどこまでも追う姿勢には驚かされた．

（担当PM・執筆：首藤 一幸）

[統括PM追記] まず，数学で習う組合せの数を表すmCnというネーミングが，名は体を表していて素晴らしい．実際，同じ大きさでこれだけのセンサやアクチュエータを作って，応用事例も完成してしまった．途中，大きなピボットがあったが，いつのまにか，誰も想像していなかったスマホの開発にまで着手したのは大きな驚きであった．饗庭さんは応募時にはまだ19歳の高専生だった．毎度のことだが，高専生パワーのすごさを思い知らされた．

（2023年7月3日受付）
（2023年9月15日note公開）

祭り運営を支援するシステム temaneki

情報処理学会・学会誌「情報処理」
2023年9月15日 07:56

阿部 優樹（あべ ゆうき）
辻口 輝（つじぐち ひかる）

祭りを運営する管理者が，さっくりと手間少なく，大勢のボランティアを管理できるスマホ・タブレット向けアプリケーションtemanekiを開発した．temanekiは**図-1**に示した3つの特徴を持つ．また，開発しただけでなく，15回にわたって運用し，そのたびに改善を続け，北海道大学での金葉祭ではボランティア170人という規模での利用を達成した（**図-2**）．

図-1　temanekiの3つの特徴

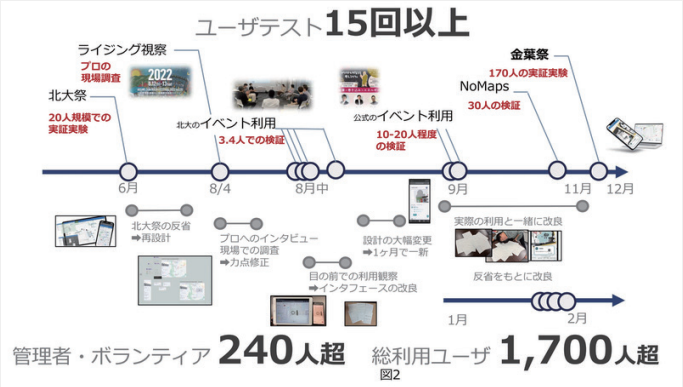

図2

図-2　temanekiの運用実績

その金葉祭では，temanekiなしでの以前の運営では，ボランティアからシフトへの苦情が殺到して，当日到着率は60%程度だったところ，temanekiを使った運用ではシフトへの苦情2〜3件，当日到着率95%を達成した（**図-3**）．

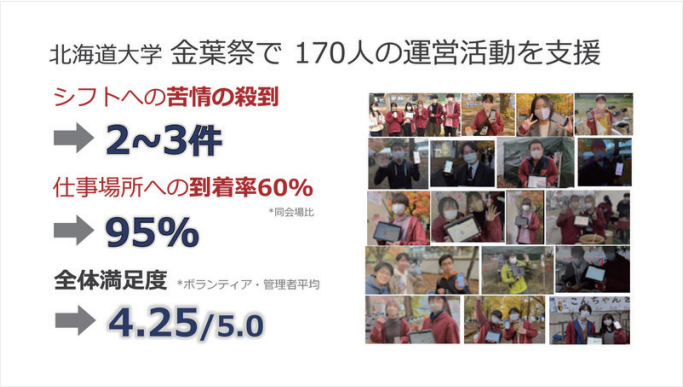

図-3　temanekiによる満足度の向上

クリエータの2人，特に阿部さんは祭りが好きすぎて，年間50以上の祭りやイベントに参加し，また，YOSAKOIソーランサークルの幹部として活動してきた．そこで気づいた運営およびその継続についての課題を，クリエータ2人で，情報技術で解決して見せてくれた．秘訣は，図-1に示した特徴そのもの，つまり連絡作業の大幅な軽減，シフトの自動作成，地図での役割説明，である．

祭りでは，前日までボランティアが増えていく．金葉祭では，当日直前の1週間だけで40人以上増えた．その全員に対して，祭り管理者はメッセージアプリなどで連絡をとることになり，忙殺される．ここでtemanekiを使うと，シフトの作成・調整に応じて，ボランティアに自動的に連絡が送られ，管理者は連絡作業から解放される．実際，祭り当日1日目の夜中に，2日目のシフトを変更するような離れ技も可能となった．シフトの作成・調整は，ボランティア全員の希望に基づいて，最適化ソルバを用いて行われる．ここに，辻口さんの大学院での専門である最適化が活きている．

ボランティアの仕事は，管理者が地図上に紙芝居のように入力し，ボランティアもそれをスマホの地図上で見る．仕事の伝達は，従来は紙のマニュアルで行ってきた．地図で示すこの方式の効果は顕著で，広い広い北海道

大学でのお祭りで，従来の運営方法では，時刻どおりに到着したボランティアが60%ほどだったところ，temanekiを使った運営では95%を達成した．

阿部さん，辻口さんの頭の中には，理想のアプリケーションがあるのではない．あるのは理想の祭りである．アプリの開発はゴールではなく，ただの手段である．アプリは，現実世界で効果を発揮してなんぼなので，こうあるべきである．未踏での開発期間が始まったころから，周囲の大人は皆「祭りだけでは対象が狭い．イベント一般を対象にして汎用性を高めるのはどうか？」と尋ね，コメントした．私も当初そう考えた．しかし2人は，祭り愛ゆえ，祭りファースト（というより祭りオンリー）を貫いた．汎用性はソフトウェアにとって諸刃の剣で，応用先が広くなる利点と，逆に誰にも刺さらなくなる危険とがある．本プロジェクトでは，祭りだけを見据えて磨き続けることが重要であった．

しかし結果としては，祭りを含めた多様なイベント運営を劇的にスムーズにできる可能性を見せてくれた．阿部さんはアプリのほとんどを開発し，辻口さんはシフトのアルゴリズムを担当し，また，temanekiの現場オペレーションを主導した．

（担当PM・執筆：首藤 一幸）

[統括PM追記] 首藤PMも書いておられるように，多くのPMも私も「祭りだけでいいの？」と心配し，そういうコメントもした．しかし，2人ともまさに祭り命のキャラクタであった．当然のことながら，彼らのプレゼン自体がお祭りだった！　その祭り愛は，彼らと面と向かうだけでオーラのようにまぶしく伝わってくる．プロジェクトにおいて，まず一点集中することにより，むしろ強力な汎用性のタネが生み出されるという，これもまた未踏の醍醐味だ．

（2023年7月3日受付）
（2023年9月15日note公開）

複数のARMマシンを1つに集約するハードウェア仮想化レイヤ

情報処理学会・学会誌「情報処理」
2023年9月15日 07:59

飯田 圭祐（いいだ けいすけ）
柚山 大哉 （ゆやま だいや）

ARMアーキテクチャは，低消費電力で高い性能を発揮し，スマートフォンやタブレット，組込みデバイスなどで広く採用されているが，Raspberry Piに代表されるような教育用や趣味に使われる小型コンピュータにも採用され，手軽に利用できるためのエコシステムが構築されている．

このように手軽に手に入る価格性能比の高いコンピュータを複数組み合わせて，手軽に1つのコンピュータクラスタとしてユーザが利用できるプラットフォームを構築するのが本プロジェクトである．スケールアウトできるコンピュータクラスタは人類の1つの夢とも言え，過去から何度もチャレンジがなされてきたが，本プロジェクトの飯田さんと柚山さんがチャレンジした，x86やSparcなどの既存アーキテクチャではなく，ARMベースの安価なRaspberry Piをソフトウェアだけでクラスタリングさせようという試みは，非常に斬新である．

実際，ハードウェアをそのままで，かつ既存のアプリケーションに手を入れずにクラスタ上で稼働させることは非常に困難であり，アーキテクチャへの理解度やプログラミング能力といった技術力だけでなく，どのCPUとメモリ空間にユーザプログラムを配置するかや，筐体を超えたメモリアクセスの保障，動的なスケールアウト・スケールインの手法など，アイディアとアルゴリズムの設計力も重要になってくるため，当初より実機で動かすという未踏性の高いプロジェクトであった．

本プロジェクトでは複数の物理ARMマシンに跨って動作するハイパーバイザを実装する．実装されるハイパーバイザ上で動作するVMは，ゲストOSからは集約された1つの物理マシンとして認識され，その上で動作するアプリケーションはクラスタリング環境であることを意識することなくリソースにアクセスできるようにする．つまり，クラスタリング環境上で既存のOS・アプリケーションがそのまま動作するようにする．具体的には，クラスタのノードとしてRaspberry Pi 4とGigabit Ethernetで構成されるクラスタに跨って動作するハイパーバイザを実装し，ゲストOSとしてLinuxを想定した上で，クラスタ上の資源を集約した単一のシステムイメージを提供するため，以下の5つを目標とした．

1. 複数のノード上に実装したハイパーバイザ上でゲストOSが動く．
2. ゲストOSが集約されたCPUとRAMを認識して透過的にアクセスができる．
3. ゲストOS上で動くマルチスレッドアプリケーションが本システムで動く．
4. 本システムでスケールするアプリケーションとスケールしないアプリケーションを示す．
5. 実行中にノードが追加されたときにゲストOSが増えたCPUとRAMを認識できる．

比較対象のプロジェクトとして，MOSIXやKerrighedなどが挙げられる．これらのプロジェクトは，Linuxを分散OSに改造してクラスタ上の資源を集約した単一のシステムイメージを提供していた．しかし，Linuxでしか動作しない，Linuxの書き換えが必要，Linuxのアップデートに追従する必要があるという問題があった．

また，vNUMAやVirtual Multiprocessor，GiantVM，vSMPなどは，それぞれ特定のハードウェアに対応した実装となっており，改変が必要な場合があったり，メモリの集約が行われなかったり，ホストOSのオーバヘッドがかかるなどの欠点があった．

これに対して，本プロジェクトでは，ARMアーキテクチャ向けに実装されたPilevisorを提供することで，既存のエコシステムを利用しながらクラスタ上の資源を効率的に活用することを目指した．

ハイパーバイザは，主に飯田さんが担当し，C言語とARMアセンブリ言語を使用してフルスクラッチで実装した．また，デバイスドライバ，ノード間通信プロトコルスタック，CPU集約機構，仮想電源コントローラ，仮想割り込みコントローラ，メモリ集約機構，IOフォワーディング，ノード間のクロック同期など，さまざまな機能を実装した．

ARMv8には，CPUの特権レベルとしてException Level（EL）が存在し，0〜3の4レベルに分かれている．EL0から数字が上がるにつれて権限が強くなって，アクセス可能なレジスタがより増えていき，各ELで実行するソフトウェアの種類は，以下の通りである．

- EL ソフトウェア種別
- EL0 ユーザモードのソフトウェア
- EL1 カーネルモードのソフトウェア
- EL2 ハイパーバイザ
- EL3 トラストゾーン

本プロジェクトでは，ARMv8アーキテクチャ向けにEL2上に構築されたハイパーバイザの実装を行った．このハイパーバイザは，ゲストOSが必要な場合にのみトラップして動作するため，演算命令など，ハイパーバイザの介入が必要のない場合は何も処理を行わずに済む．ハイパーバイザの介入が必要な場合には，ARMv8の仮想化支援機構を利用して高速にトラップを行い，ハイパーバイザの処理を行うことができる．

EL2はEL1よりも強い権限を持っており，EL2で実行されるハイパーバイザは，EL1で稼働するホストOSの保護を行うことができる．また，EL2はEL1の割り込みをトラップすることができ，EL2からホストOSに割り込みを転送することができる．この機能を利用することで，ハイパーバイザはゲストOSが発生させた割り込みをトラップし，必要に応じてホストOSに割り込みを転送することで，ゲストOSとホストOSの間で割り込みを共有することができる．

ハイパーバイザは薄いハイパーバイザであり，単一の仮想マシンのみをサポートする．この薄いハイパーバイザの実装により，仮想化によるオーバヘッドを最小限に抑えることができる．

メモリアクセスについては，自ノードへのメモリアクセスと他ノードへのメモリアクセスを区別するために，2段階ページ変換を使用している．これは，ゲスト空間から見た物理アドレスと実際の物理アドレスの変換を透過的に行う仮想化支援機構の1つである．ARMでは，Stage 2 Page Translationと呼ばれる．2段階変換ページテーブルを使用して，仮想共有メモリの実装を行っている．このページテーブルは，ページサイズ（4,096バイト）単位でマッピングを行い，他ノードから受け取ったメモリページをキャッシュする．また，メモリのownerを追跡し，キャッシュ一貫性制御を保つ仮想共有メモリマネージャを実装している．

ノード間通信は，飯田さんとともに柚山さんが実装した．Ethernetを介してノード間でメッセージを伝える必要があり，そのためにノード間通信プロトコルを制定し，プロトコルスタックの実装を行った．送信メッセージのサイズが最大4,160バイトで，Ethernetフレームのデフォルト上限サイズである1,536バイトを超えるため，ジャンボフレームを使用することにした．メッセージは17種類あり，それぞれ異なる目的を持っている．ペイロードにEthernetヘッダを付ける方法については，Pilevisorではポインタ演算と最小限のコピーのみでヘッダの追加・削除ができるデータ構造を実装しており，処理のオーバヘッドを最小限に抑えている．

各ノードが起動したときは，自分以外のノードが誰なのか，自分はどこのノード担当なのか分かる術を持っていない．そのため，最初に認識のための通信を行って，各ノードがクラスタの状態を知る必要がある．ノード認識は，6つのステップで行われる．最初に，メインノードがLAN内にいる全員に発見ブロードキャストを行い，その後，サブノードが応答して各種情報を送信する．その情報を元に，クラスタ情報を構築し，全ノードにブロードキャストする．その後，各ノードはクラスタ情報を元に初期化を行い，初期化終了通知をメインノードに送信する．メインノードは，全ノードからsetup doneが返ってくるまで待ち，前準備が行ったら，boot sigメッセージをブロードキャストする．最後に，sigを受け取ったノードは，割り込みコントローラの機能を使用して，ゲストOSに仮想割り込みを注入する．また，物理CPUのCPU番号はノードごとに0から割り当てられているため，仮想マシンが直接物理CPUのCPU番号を直接参照すると，矛盾が生じる．そのため，ノード間認識通信のときに決定された仮想CPU番号を仮想マシンに見せるようにする必要があり，ARMv8の仮想化支援機構を使用した．

これらの実装により，ARMマシンのクラスタの資源を仮想化技術によって集約し，単一の仮想マシン上で複数のオペレーティングシステムが同時に動作することを可能にし，複数のマシンを組み合わせて，効率的にリソースを利用することができるようになった．

なおクリエータたち自身は，これまで「車輪の再発明」しかしてこなかったと述べており，実際に既存の実装を改良することでさまざまな開発を行なってきたようであるが，今回のプロジェクトにおいては先行の事例がきわめて少ない中で，試行錯誤をしながらさまざまな新しい実装を進めてきたことが素晴らしい．

実際にプロジェクトの途中では実装が進まない時期もあり，かつ情緒面でも心配されることがあったり，実装が危ぶまれたりしたこともあったが，典型的な追い込み型で成果報告会の発表直前に実機稼働を成し遂げるという，ヒヤヒヤしながらも結果の出せるプロジェクトになった．

これらの成果物はオープンソースで公開しており，ほかの研究者や開発者がこれを利用して，より高度な仮想化技術を実現することができるようになる．これにより，ARMマシンを用いたさまざまなアプリケーションやシステムの開発が促進され，より効率的かつスケーラブルなシステムの構築が可能になることが期待される．

（担当PM・執筆：田中邦裕）

[統括PM追記] 田中PMが非常に詳細な実装内容を記述されていて，図がなくても実際にどんなことを2人がしたかが手に取るように分かる．当人たちは「すべて車輪の再発明」と謙遜しているが，そういったものを積み上げて，これまでになかったシステムを作り上げることもまた未踏の醍醐味である．成果報告会の前日にシステムが動いたという綱渡りだったので，これがどのような問題に対してスケールするのかしないのかはぜひ確かめて，普及させてほしい．

（2023年7月3日受付）
（2023年9月15日note公開）

HDCアクセラレータ「HPU」

情報処理学会・学会誌「情報処理」
2023年9月15日 08:02

井阪 友哉（いさか ゆうや）

井阪さんは，HDC（Hyperdimensional Computing）のハイパーベクトルを効率的に処理できるHDCアクセラレータ「HPU」をFPGAで開発した．HDCは1990年代に高い次元性とランダム性に依存する認知モデルとして提唱されたが，1950年代に登場した深層学習と比べると比較的新しい概念で歴史が浅い．そのため，HDCはいまだメジャーではなく，実際に高速性と低消費電力が実現できるハードウェアの開発や，世の中のさまざまな分野の開発者がそれぞれの認識問題の課題に対してHDCを活用して解くまでに至っていない．

HDCは，数千から数万次元のベクトルのHyper Vectors（以下，超次元ベクトル）を用いた計算モデルであり，さまざまな認知タスクに応用可能と言われている．HDCでは超次元ベクトルに対しての独自の演算が定義されており，バイナリの超次元ベクトルとして応用することでコンピュータで扱いやすくなる．

HDCアクセラレータ「HPU」では，**図-1**に示すHDC独自の演算である3種の演算（Bind，Bound，Permutation）すべてに対して高速に処理できる演算器を実装したアーキテクチャを設計・開発した．

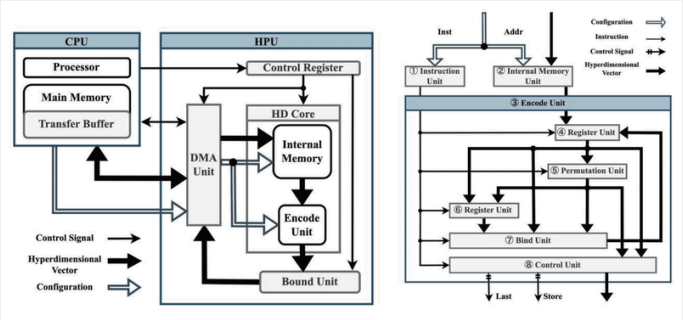

図-1　HDCアクセラレータ「HPU」のシステム構成図

図-1の左側で示すように，CPUとHPUが協調動作するアーキテクチャとして開発した．このアーキテクチャによりHDCアクセラレータの応用先が広がる．HDCを応用したエッジサーバのプログラムを簡単に開発できるように，C言語を使って簡単にCPUから実行できるライブラリも開発した．これにより，開発者はライブラリの関数を呼び出すだけで，簡単にCPUからHPUに指示を出すことが可能になっている．

実際に実用可能性を検証するために，HDCを用いた画像認識・音声認識・言語認識の3つのアプリケーションテストを開発し実行速度を計測した．HPUはFPGAとARM-v7 CortexA9プロセッサをワンチップに搭載したZynq XC7Z020 SoCを搭載したXilinx PYNQ-Z1ボード上に実装し，ホストのCPUであるARM-v7から扱い，1,024次元の超次元ベクトルを処理する．いずれの結果でもM1 Max（3.00GHz），Intel Core i7（1.80GHz），ARM-v7 Cortex-A9（0.666GHz）と比較した場合，大幅な速度改善が確認され，最大で169倍の高速化に成功した．また，電力効率を示す指標の1つであるエネルギー遅延積（ED積）は最大13,469倍も改善した．それぞれの認識タスクにおける各CPUとの速度比較のグラフは**図-2**の結果となった．

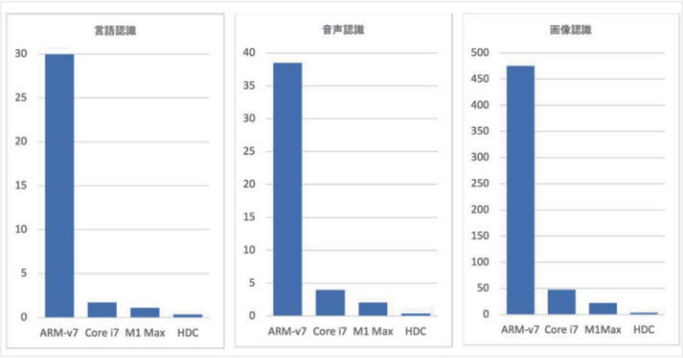

図-2　アプリケーションテストにおける速度比較

今回の成果を広めることができれば，これまでよりも手軽で高速にHDCを実行できるようになるため，HDCアプリケーションの研究者や開発者は実験や開発に必要な試行錯誤のサイクルを大幅に改善できるようになる．最近，別の応用物理の分野ではRNNの一種であるRC（Reservoir Computing）が注目されており，物性デバイスとの実現性の相性の良さから，ハードウェアの試作やエッジIoTデバイスでの概念実証実験が進みつつある．HDCの研究も今後盛り上がってくれると嬉しい．

また，今回はFPGAで実装したが，ASIC化，つまりHDCアクセラレータに必要な機能のみを組み合わせてチップを製造することで，チップ単価が安くなり，消費電力も低減し，熱対策の改善などが可能となる．RISC-Vの組み込みも親和性が高い．

このように高速な計算アーキテクチャを一から検討するという技術的に難易度の高い低レイヤのプロジェクトであったが，DMA転送や並列実行可能性など，ボトルネックを解消できそうな各種ステージを井阪さん自らが見つけて特定し，消費電力当たりで高い計算効率性能を出すことができた．今後，この成果を論文の形できちんとまとめ，HDCを利用するための環境やツールチェーンの基盤を整備することによって，世界中でHDCアルゴリズムの検証やHDCのプロトタイプが作りやすくなり，HDCの普及と発展につながることを期待したい．

（担当PM・執筆：竹迫 良範）

[関連URL]
https://github.com/yuya-isaka/HPU （開発中）

[統括PM追記] HDCは一般のニューラルネットワークよりも単純と言われ，人間の小脳のような情報処理をするらしい．並列の基本演算が3種類しかないので単純で均質な構造で実現できる．だから，省エネなのだろう．応用がある程度限定されるようだが，はまればドツボなのは，井阪さんの実験結果を見ても明らかだ．大昔，ホモジニアスな計算対ヘテロジニアスな計算というパラダイム論争があったが，HDCはホモジニアスのほうだろう．ホモとヘテロが階層的に重なることで，本物のAIができるという話があったことを思い出した．

（2023年7月3日受付）
（2023年9月15日note公開）

直和型の代わりにユニオン型を持つ関数型言語 Cotton

情報処理学会・学会誌「情報処理」
2023年9月15日 08:04

伊藤 謙太朗（いとう けんたろう）

伊藤さんは高校生のころから自分の理想のプログラミング言語を追求し，直和型ではなくユニオン型を採用した型システムを持つシンプルな静的型付けプログラミング言語Cottonを開発した．直和型とはプログラミング言語Rustのenumであったり，HaskellやOCamlのデータ型のように複数のコンストラクタを持つデータ型のことで，パターンマッチが使えるプログラミング言語によく見かける機能である．それに対して伊藤さんが大好きなユニオン型は，最近のTypeScriptやCrystal，Juliaなどで採用されている機能で，ユニオン型に便利さを感じているプログラマも徐々に増えてきている一方，静的型付け関数型言語の世界はユニオン型を採用する言語はまだ少ない．またあったとしても直和型やシールドクラスなどほかの機能の補助的な役割を担うのみにとどまっており，ユニオン型が持つ表現力の追求が行われていない．

伊藤さんはユニオン型を持つ静的型付けプログラミング言語Cottonを設計し，コンパイラとランゲージサーバを開発し，以下の機能を実装した．

- ユニオン型
- 再帰的な型エイリアス
- 網羅性チェック付きのパターンマッチ
- 型推論

- 演算子定義
- 高階多相
- オーバーロード
- インタフェース
- モジュールシステム

たとえば，Cottonでフィボナッチ数を出力するコードは**図-1**のように書ける．

```
1   (.) : A -> (A -> B) -> B forall { A, B } =
2       | a, f => f(a)
3
4   infixl 10 .
5
6   fib : I64 -> I64 =
7       | 0 => 0
8       | 1 => 1
9       | n => (n - 1).fib + (n - 2).fib
10
11  main : () -> () =
12      | () => 10.fib.println
```

図-1　フィボナッチ数を出力するコード

図-1の1〜2行目で，右辺の関数に左辺の値を適用する演算子 . を定義している．forall { A, B }は，AとBが型変数であることを宣言している．この . 演算子によって，関数呼び出しをオブジェクト指向言語のメソッド呼び出しのように書くことができる．たとえば，図-1の2行目にあるf(a)はa.fと書くことができ，さらにprintln(fib(10))は図-1の12行目のように10.fib.printlnと書くことができる．図-1の4行目のinfixlは左結合を表し，優先順位を10と定義している．ちなみにこのCottonの.演算子は暗黙的にインポートされるモジュールpreludeで定義されているため，自分で図-1の1〜4行目のようにわざわざ定義しなくても最初から使えるようになっている．

図-1の7〜9行はこれで1つのラムダ式で，引数に0が来たら0を返し，1が来たら1を返し，それ以外の値nが来たら(n-1)項のフィボナッチ数と (n-2)項のフィボナッチ数を足した値を返している．このようにパターンマッチとラムダ式が1種類の構文で表現されており，直感的に理解しやすいコードが書ける．

次に，CottonでFizzBuzzを出力するコードは**図-2**のように書ける．

FizzBuzzとは英語の言葉遊びで，1から100までの数字を画面に出力するプログラミングの問題だが，3の倍数のときは数字ではなく文字列Fizzを出力し，同様に5の倍数のときはBuzzを出力し，15の倍数のときはFizzBuzzと出力する．

```
1  (..) : I64 -> I64 -> List[I64] =
2      | a, b => (a < b).
3          | True => a /\ (a + 1..b)
4          | False => Nil
5
6  infixl 4 ..
7
8  map : List[A] -> (A -> B) -> List[B] forall { A, B } =
9      | Nil, _ => Nil
10     | h /\ t, f => h.f /\ t.map(f)
11
12 fizzbuzz : I64 -> String =
13     | n => (n % 3 /\ n % 5).
14         | 0 /\ 0 => "FizzBuzz"
15         | 0 /\ _ => "Fizz"
16         | _ /\ 0 => "Buzz"
17         | _ /\ _ => n.i64_to_string
18
19 main : () -> () =
20     | () => do
21         (1..101).map(
22             | i => i.fizzbuzz.println
23         )
24         ()
```

図-2　FizzBuzzを出力するコード

事前の準備として，1..101と書くと1,2,3,4〜100までの数字のListを返せるように図-2の1〜6行目で .. 演算子を定義している．また，図-2の8〜10行目では，Listの各要素に関数を適用して新しいリストを返す関数mapを定義している．図-2の21行目では（1..101）.mapと書いてそれらを呼び出している．

図-2の12〜17行目ではfizzbuzz関数を定義している．図-2の13行目では3の剰余と5の剰余がゼロかゼロ以外かの組合せでパターンマッチさせ，図-2の14行目のように両方ゼロのときはFizzBuzzを，図-2の15行目のように3の剰余のみがゼロのときはFizzを，図-2の16行目のように5の剰余のみがゼロの場合はBuzzを，両方とも非ゼロの場合はその数字を文字列に変換したものを返すようにしている．

このようにCottonはシンプルな構文で自分自身を拡張できるように設計されている．

最後に，ユニオン型の便利な側面としてエラーを表すときに有用であり，Cottonでエラーハンドリングを行うコードは**図-3**のように書ける．

```
1  data Error1
2  data Error2
3  data Error3
4
5  may_fail_1 : () -> Result[(), Error1] = | () => Err(Error1)
6  may_fail_2 : () -> Result[(), Error2] = | () => Err(Error2)
7  may_fail_3 : () -> Result[(), Error3] = | () => Err(Error3)
8
9  do_1_2_3 : () -> Result[(), Error1 | Error2 | Error3] =
10     | () => do
11         ().may_fail_1?
12         ().may_fail_2?
13         ().may_fail_3?
14         ().Ok
```

図-3　エラーハンドリングするコード

図-3の11〜13行目にある？はエラーだった場合にアーリーリターンをする記法で，この例ではエラー時の型

がそれぞれ違う3つの関数に対するエラーハンドリングを行っている．図-3の9行目で定義しているdo_1_2_3の型ではユニオン型によって発生する可能性のあるエラーを具体的に示していて，このエラーに対して網羅性チェック付きのパターンマッチをすることも可能だ．Rustにも同じような記法があるが，Rustは直和型を採用しているためこのような場合にはenumやトレイトなどを使って3つのエラーの型が同一になるように調整する必要があり，エラーハンドリングが煩雑になってしまう．Cottonではこのようにユニオン型を使ってシンプルに書き表すことができる．

CottonはGithub上で開発が続けられている．VSCode向けの拡張機能も実装されており，リアルタイムのシンタックスハイライトも可能で，マウスオーバーすると内部の型表現を出力する機能もある．Gitpod上で数クリックでPlaygroundとして実行できるオンライン環境も整備されているので，興味を持った方はぜひ触ってみていただきたい．Cottonの成果を広く公開することで，静的型付け関数型言語にユニオン型を採用する有用性を多くの人に知ってもらい，その後開発されるプログラミング言語の開発に影響を与えることを期待している．

（担当PM・執筆：竹迫 良範）

[関連URL]
https://github.com/nanikamado/cotton （開発中）

[統括PM追記] あるプログラミング言語がユーザコミュニティや開発コミュニティを巻き込んで「文化」にまで成長するには相当な時間を要するし，はっきり言ってしんどい．しかし言語設計に挑戦する，というよりは楽しむという「プログラミングの文化」を維持することは，プログラミングに関する柔らかい発想のためにとても重要である．伊藤さんは明確なアイディアをベースにきっちりと言語環境まで作り込んでいることからも分かるように，余裕を感じさせる開発を進めた．実際，それは今期のクリエータの中で，他プロジェクトに対する発言の多さで最も目立ったことにもつながった．

（2023年7月3日受付）
（2023年9月15日note公開）

レイアウトの自由度とキー操作性を両立したノートテイキング「鍵記」

情報処理学会・学会誌「情報処理」
2023年9月15日 08:07

稲葉 皓信（いなば　てるのぶ）

ノートテイキングアプリは，アイディアの整理や情報の共有，プロジェクト管理など，リアルタイムで情報を整理する便利なツールとして欠かせない存在である．しかし従来のノートテイキングアプリはマウス操作が主流であり，キーボード操作を主とした高度な自由度を持つノートアプリは珍しく，手を離さずに情報を入力することができないという問題があった．

そこで稲葉さんは，ブラウザ上で動作し，キーボードのみで操作可能なノートアプリ「鍵記（kenki）」を開発した．このシステムを使うことで，Markdown言語やキーボードショートカットを駆使して効率的にノートを制作することが可能となる．

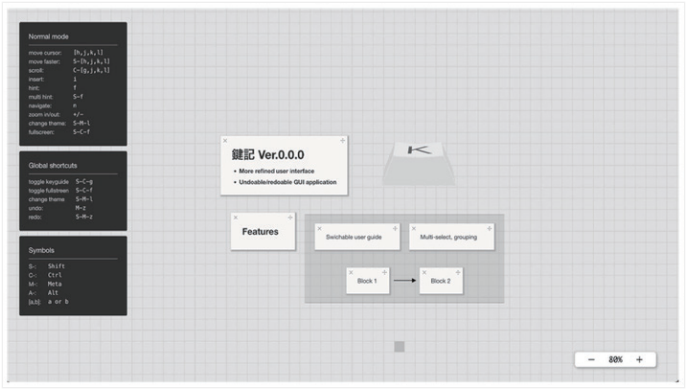

図 -1　「鍵記」アプリのスクリーンショット

鍵記（kenki）は，マウスを使わずキーボードのみで操作が可能な新しい操作体験を提供する．ユーザがスムーズにノートを作成・編集できるよう，Markdown言語やキーボードショートカットなどの機能を実装している．それによって，ユーザがキーボードから手を離さずに情報を入力することが可能となっており，情報の整理やアイディアの記録がより効率的に行える．Markdown言語を用いてビジュアルノートを編集するというツールはこれまでにもあったが，既存の手法では，自由なレイアウトを実現するのは難しいという制限があった．そこで，本プロジェクトでは，Markdown言語とキーボードショートカットを組み合わせて，文字を入力したりオブジェクトを自由にレイアウトしたりすることを可能にしている．

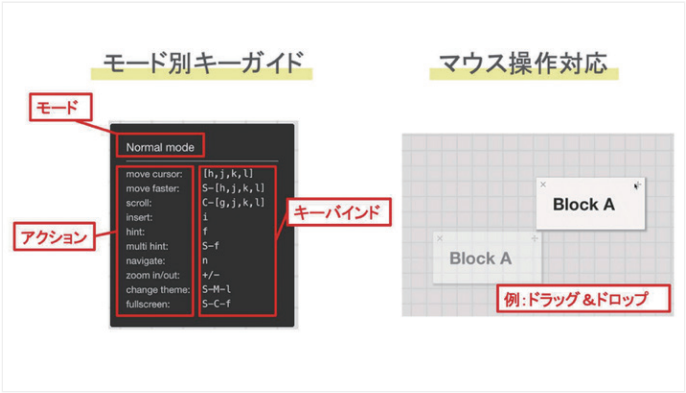

図-2　キーボードのみで操作が可能なUI・キー操作への対応

また，アプリ内で利用できるショートカット一覧や操作ガイド，操作動画の作成なども行われ，ユーザが「鍵記」を最大限に活用するためのサポートも充実している．

キーボードの操作のみで，ビジュアルなオブジェクトも扱うことができるノートアプリを作成したい，というクリエータの稲葉さんの情熱に支えられ実現したという面が非常に大きい．納得のいく操作性を実現するために何度も試行錯誤と繰り返しながら開発に取り組み，キーボード操作に特化したビジュアルノートアプリという新たな可能性を示した点で，高く評価できる．彼の創造力と技術力の賜物である．

プロジェクトの開始当初，稲葉さんはアルバイトでの開発や既存システムをベースとした開発の経験はあったものの，プロジェクトの発案から要件定義，実装，そしてフィードバックを基にした改善までの一連の開発プロセスを1人で行った経験はなかった．しかし，プロジェクトが進むにつれて，彼自身が開発を推進していく自走力を見つけ，それを十分に活かすことができた．また，彼の情熱と技術力，そしてユーザへの配慮が見事に結び

ついた本プロジェクトを通じて，さまざまな挑戦が可能であることが示された．未踏プロジェクトを通じて，技術力だけでなく，プロジェクトマネジメント能力などを大きく向上させるクリエータは多い．稲葉氏も未踏を通じて大きく成長したひとりである．

そして，成果発表会では，ほかの発表を聞きながら実際に「鍵記」を使って作成したノートが稲葉氏によって披露された．キーボードのみを駆使して多様なオブジェクトを使ったノートの作成過程は圧巻であった．今後も機能拡張に取り組んでいくということで，より多くのユーザに使われるアプリに育て上げることが期待される．

（担当PM・執筆：岡 瑞起）

[関連URL]
https://kenki-editor.vercel.app/

[統括PM追記] エディタについては，vi派，emacs派の宗教論争がいまだに続いているような気がするが，かなり筋金入りのvim遣いである稲葉さんは，マウスを使わないでテキストボックスを移動したり，サイズを変更したり，線で結んだりといったマウスが得意とする作業もすべてvimコマンドの自然な拡張という形で実現した．そういう意味で哲学はバッチリで，一貫性のあるUIデザインとなっている．かく言う私は結構頑固なemacs派なので，鍵記と同じスタイルのマウス不要ノートアプリが欲しくてうずうずしてきた．

（2023年7月3日受付）
（2023年9月15日note公開）

サッカー試合映像の検索・分析システムTASC

情報処理学会・学会誌「情報処理」
2023年9月15日 08:09

内田 郁真（うちだ いくま）
スコット アトム

内田さんとスコットさんは部活動やクラブチームに所属するようなアマチュアサッカー選手を対象として，サッカーの試合映像を対象にビデオトラッキングベースの自動分析ツール「TASC（Tracking AI for Soccer Coaching）」を開発した．効率の良い映像視聴を支援する作業を自動化し，撮影負荷が低く，かつ技術向上に必要となるプレー映像を素早くかつ簡単にアクセスできるような仕組みを開発した．これにより，アマチュアサッカー選手が自身のプレーの撮影を手軽に行い，振り返りや次戦の対策など，サッカーの技術向上のために必要不可欠な試合映像の振り返りのための映像選定を支援することができるようになった（**図-1**）．

図-1
TASCの映像認識とデータ解析により，技術向上に必要なプレー映像を手軽にアクセス可能に

内田さんとスコットさんは筑波大学体育会サッカー部に所属していた経験がある．また，コーチングスタッフや J1チームでのアナリスト経験もあり，サッカーにかける強い情熱がある．複数の国際学会 CVPR Sports'22，ACM MM Sports' 21，ICAART 2022において主著論文もあり，コンピュータビジョンの研究者としての高い研究遂行能力・知見もある人材だ．本プロジェクトを一緒に行ったこの2人はサッカー部での選手やアナリストをともにした相棒であり，CVPR, MM の論文では Co-first authorship（共同第一著者）でもあり，良いパートナーである．そんな2人がサッカーの現場で感じた課題感の解決を密にコミュニケーションをとりながらフラットな関係で開発を進めていった．

内田さんは主に，どの映像をハイライトにするか，プレーの良し悪しといった判断など，「プレー評価指標」について技術開発を担当した．また，検索機能の面では，幾何学的な類似度や深層学習ベースの類似度などの技術を担当した．スコットさんは技術面では主にトラッキングやピッチの認識，パスイベントの検出といった画像処理を担当した．また，LINE や YouTube API 連携など，ユーザインタフェース周りも担当した．「主に」と書いたが，2人がそれぞれ完全に独立で開発を進めてきたわけではなく，お互いがお互いを補いあって開発していった．未踏事業はグループでの応募の場合，システムには代表者1名を記入するような仕様になっているが，このプロジェクトでは内田さん・スコットさん2人がCo-firstである．このプロジェクトが成功につながった強さの秘訣はここにもある．

2人はプロジェクトを進めていくにあたり，多くの現場関係者へのヒアリングを実施した上で必要な機能を取捨選択して実装していった．プロジェクト開始からしばらくの間はプロサッカー選手を対象に検討していたが，対象をアマチュアサッカー選手に絞り，スマートフォン1台で手軽に無人で撮影できるようなシステムに仕上げることにした．また，ユーザインタフェースについても誰でも手軽に使えるLINEとYouTubeを利用することで，導入の敷居を下げることに成功した．学術的にも新たな手法やアルゴリズムなどを考案・開発し，システムを作り上げた．

2人は未踏期間終了後，大学院博士課程に進学し，TASCの要素技術の研究に焦点を当てている．ビデオトラッキングや映像検索は学術領域でもまだ困難が多いタスクでもある．これらの課題に直接取り組める環境に身を置きながら，プロダクトの発展を目指している．

（担当PM・執筆：五十嵐 悠紀）

[関連URL]
https://github.com/AtomScott/SoccerTrack

[統括PM追記] 本文中にも記載されているとおり，当初，プロサッカーチームでも使えるシステムも目指していたが，国際的にお金に糸目をつけない競合が多い中，アマチュアチームに焦点を絞ったのは，コスト制約が強く，むしろ挑戦的で，正解だったと思う．スマホ1台にこだわったのはその顕れである．2人とも博士課程に進んで，TASCの完成度をさらに上げようとしていることに，強い意気を感じる．でも，そろそろカメラ1台の制約を外して，コストをあまり上げずに，さらに精度の高い有用な情報を得られるようにする選択もあっていいのではなかろうか．

（2023年7月3日受付）
（2023年9月15日note公開）

麻雀プロのためのAI牌譜解析ツール 極

情報処理学会・学会誌「情報処理」
2023年9月15日 08:10

大神 卓也（おおがみ たくや）
奈良 亮耶（なら りょうや）
天野 克敏（あまの かつとし）
今宿 祐希（いまじゅく ゆうき）

プロ雀士が研究や対局，また，牌譜の解説に活用し得る麻雀AIとツール 極（ごく）を開発したプロジェクトである．

プロ雀士の強さには至らなかったものの，ネット麻雀 天鳳にて七段，つまりプレイヤの上位1.2%という強さに到達した（**図-1**）．極は，従来の麻雀AIとは異なり，1位の獲得が重要なプロ麻雀向けの判断と，4位回避が重要なネット麻雀向けの両方を行うことができる（**図-2**）．ツールとしては，牌譜解析ツールを開発し，期間限定で一般公開した．有用なツールとするため，手牌推測にも取り組み，従来のシステムより高い精度を達成した．手牌推測の良さを評価する新指標もいくつか提案した．

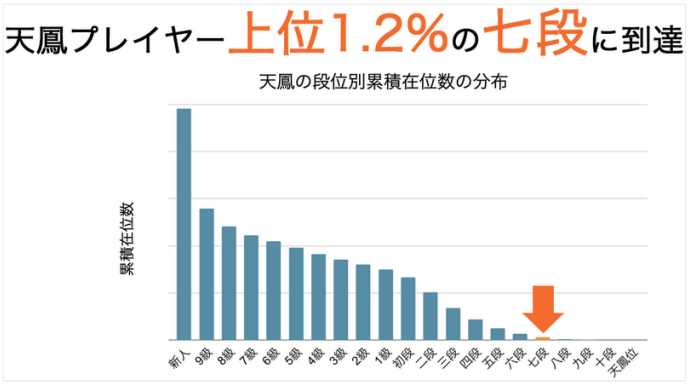

図-1　天鳳7段を達成

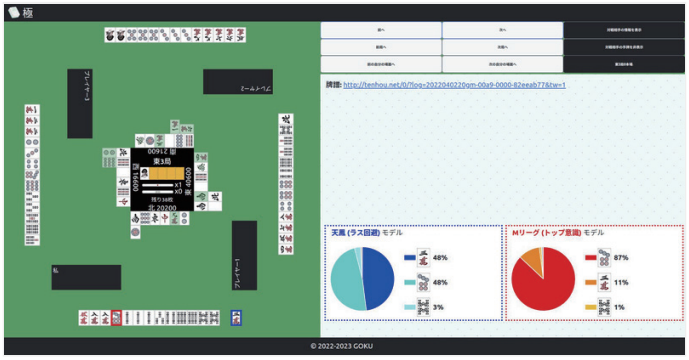

図-2　ネット麻雀とプロ麻雀への対応

これらの取り組みを，プロジェクト初期より，プロ雀士や強豪プレイヤからの協力を得て進めた．たとえば，手牌推測の結果にコメントをいただいたことが新指標に結びつき，1月にはプロ雀士との公開対局を実現した．

麻雀AI Suphxという先達に習えるところがあったとはいえ，1年もかからず人間の上級者レベルに到達し，ネット麻雀への参戦を果たし，上位1.2%という天鳳七段に到達した．それでも，強い麻雀AIの開発はあくまでもプロジェクトの基礎であり，大目標はプロによる麻雀AIの活用である．その大目標に向け，牌譜解析ツールを開発した．手牌推測も，従来になかったレベル，つまりは世界最高の精度を達成した（**図-3**）．

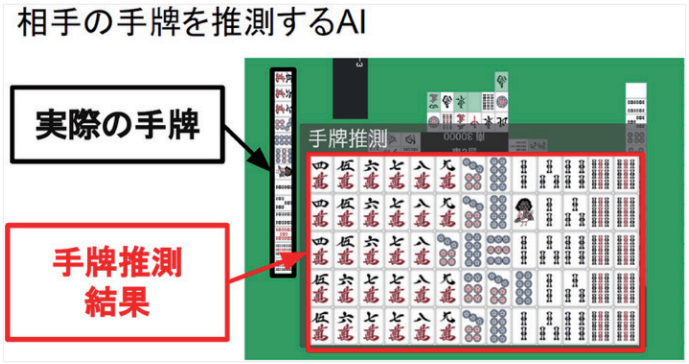

図-3　手牌推測

こうした華々しい成果の裏には，機械学習に使う麻雀シミュレータの開発，手法やシステムの工夫による強化学習の100倍以上の高速化といった，目立たないが欠かせない成果も数多くあった．

4人いてこそ到達し得た成果ではあるが，4人いればできたかというと，そうではない．4人それぞれが，互いを補いつつ高度な仕事を行い，そうした4人の成果がしっかりと噛み合ってこその到達点である．4人の誰を欠いても，この到達点はなかった．大神さんは特に，天野さんと同様，教師あり学習，順位点対応に取り組んだ．加えて，手牌推測は彼1人の仕事である．奈良さんは特に，対局解析Webアプリケーションに取り組んだ．加えて，ネット麻雀自動対戦は奈良さん1人の仕事である．また，開発環境・サイクルを整備して4人でのチーム開発を支えたのも奈良さんである．天野さんは特に，大神さんと同様，教師あり学習，順位点対応に取り組んだ．加えて，強化学習，および，雀風の模倣は天野さん1人の仕事である．今宿さんは，教師あり学習のデータ作成に共同で取り組んだほか，特に，対局解析Webアプリケーションに取り組んだ．深層学習の知識や開発経験がまだ薄かった状況から，類似研究の先端に迫り，一部先端を追い抜くところまで至った4人の成長も著しかった．

（担当PM・執筆：首藤 一幸）

[統括PM追記] 首藤PMはプロ最高レベルに到達しなかったことを悔しがっておられるが，着手選択（麻雀の場合は捨て牌の選択）のベースとして相手の手牌推測をきちんと行っているので，深層学習AIの説明可能性の問題をかなり解決している．これは従来の麻雀AIでは成し遂げられなかったことだと思う．4人ともかなりの麻雀好きとのことだが，もうこの極システムにはまったく勝てないという話を成果報告会で聞いた．開発者を超えるAIがどんどんできる時代になったものだと，私のようなロートルはただただ感銘するばかりである．

（2023年7月3日受付）
（2023年9月15日note公開）

VRと電動トレーニング機器を用いた筋力トレーニングシステム TRAVE

情報処理学会・学会誌「情報処理」
2023年9月15日 08:13

栗本 知輝（くりもと ともき）
黒木 琢央（くろき たくお）
松田 響生（まつだ ひびき）

デジタル技術を用いたトレーニングシステムは過去の未踏事業でも数多く提案されていますが，TRAVEと名付けられた本システムは，独自の電動トレーニング機器とVRを用いることで，ゲームをするように楽しく筋トレの効果を向上させようとしている点に新規性があります．また，VRでの触力覚提示デバイスは振動や数N（ニュートン）程度の力をユーザに伝えるものが多いですが，こちらの装置は数百N以上の力を提示可能である点も特長となっています．

TRAVEの全体像を**図-1**に示します．TRAVEは，自作したデバイス，それをUnityから操作するためのAPI，これらを用いたVRアプリケーション（以降，VRアプリ）から構成されています．

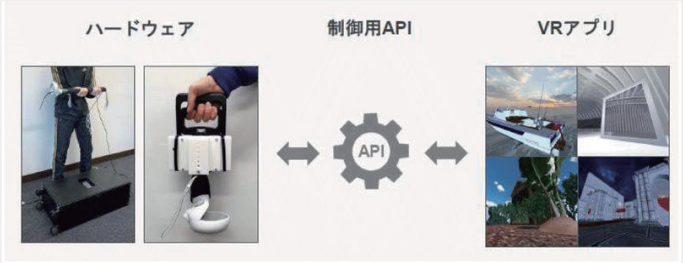

図-1　システムの全体像のイメージ図

本プロジェクトでは，TRAVEを活用して筋トレの視覚・聴覚・力覚体験を拡張し，筋トレの課題を解決するようなデモアプリを，3つ作成しています．

1つ目は，釣りをモチーフにしたデモアプリで（**図-2**），VR空間での大きな力覚演出を筋トレに取り入れることで，負荷そのものに楽しみを加えて，筋トレの単調さの解決に寄与しています．実際の運動ではユーザがハンドルを上げて腕の筋肉を鍛えていますが，VRの世界では，それが釣りユーザが大型魚を引き上げる体験に変換されます．

このアプリケーションには，魚が釣り針に引っ掛かる感覚，魚が暴れる感じ，大型魚が竿を強く引っ張る演出など，多数の力覚演出が組み込まれています．さらに，ユーザの筋力を自動で測定し，適切な負荷を設定する機能や，運動中に自動的に負荷を調節する機能など，電動トレーニング機器の特性を活かした筋力トレーニング支援機能も導入しています．

図-2　釣りをモチーフにしたデモアプリのプレイ画面

2つ目は，VR空間で生起される錯覚効果を利用して筋トレ効果を向上させるデモアプリです（**図-3**）．本アプリでは，挙上動作をガイドする移動壁が右方向から左方向に向かって流れていくため，その移動壁に当たらないように所持していくスティックを操作します．本アプリには2つの錯覚効果を用いており，1つ目は，プロテウス効果を利用し筋肉質なアバターを生成して自身の体と感じてもらうことで，挙上回数の向上を狙っています．2つ目は，Pseudo-Hapticsを利用し，VR空間での腕の動きを実際よりも大きくすることで，疑似力覚が生じて重量を軽く感じさせることができます．

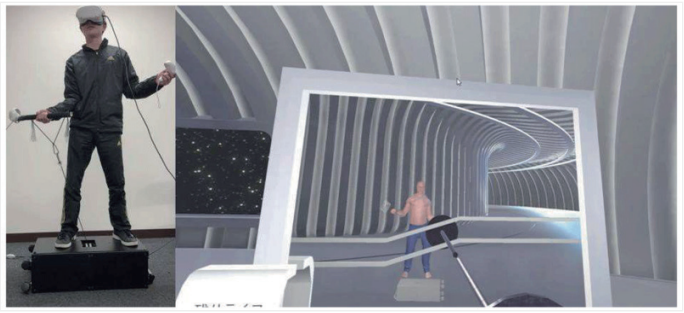

図-3　VR空間での錯覚効果を用いたデモアプリのプレイ画面

3つ目は，異なる種目・筋力・場所同士での合同トレーニングを実現するアプリです（**図-4**）．今回は握力系と筋トレデバイスを通信させることで，異なる種目であるにもかかわらず合同トレーニングを可能としています．握力計側のユーザは，現実空間では握力計を握っていて，VR空間では魔法の杖を握り，力に応じてビームを出します．一方で，筋トレデバイス側のユーザは，大きなカブを引っ張っているように見えます．これらデバイスの出力の大きさを，ユーザ間の筋力差に応じて調整することで，異なる筋力間の協調を実現しています．そして，遠隔通信を行い異なる場所間での協調を実現することに成功しました．

図-4　異なる種目・筋力・場所同士での合同トレーニングを実現するデモアプリのプレイ画面

本プロジェクトで開発したTRAVEを利用することで，家で筋トレを行う人とジムで筋トレを行う人の両方が，筋トレの楽しさ，成果，モチベーション等を向上させることができるようになることでしょう．デモアプリが示す力覚的楽しさ，VRの錯覚効果による支援，異なる種目・筋力・場所間での合同トレーニングに加え，視覚・聴覚・力覚に連携して介入できるさまざまな効用をトレーニングに提供できます．将来的には，筋トレだけでなく，VRと高出力の力覚連携システムのプラットフォームとして，VR領域全体に対する研究・開発に大きく貢献できると期待しています．

（担当PM・執筆：稲見 昌彦）

[関連URL]
https://home.trave-project.com/

[統括PM追記] まず，VRによるある種の錯覚を利用した筋トレのアイディアが素晴らしい．大きな力覚を生むための少々重い装置などが必要だが，VRゴーグルをかけるのでそれはもう目に見えない．成果報告会で示されたデモはどれも素晴らしい説得力があった．実際に大型魚を釣り上げる体験をした人はほとんどいないだろうから，筋トレ目的でなくても，魚種によって変化する釣りの醍醐味を仮想的に味わえる装置として，釣った魚のリアルっぽいVR映像も用意すれば，遊技場やテーマパークでも使

われるようになるのではなかろうか．これに限らず，TRAVEは筋トレ以外の展開が大いにありそうだ．

（2023年7月3日受付）
（2023年9月15日note公開）

UVプリンタを用いたラインストーン造形システム

情報処理学会・学会誌「情報処理」
2023年9月15日 08:14

島元 諒（しまもと　りょう）

ラインストーンは，装飾やアクセサリーに用いられる小さな装飾用の石で，さまざまなパターンやデザインが可能である．従来，ラインストーン作品は手作業で作成されていたが，手間がかかるため一般のユーザにとっては敷居が高いという課題があった．

そこで島元さんは，UVプリンタを活用し，デザインから印刷まで手軽に行えるシステムを開発した．このシステムを使うことで，子どもや細かい作業が苦手なユーザでもWebアプリケーション上でデザインを手軽に行え，さらにUVプリンタで短時間に制作することが可能となった．

UVプリンタは，紫外線硬化インクを使用して多種多様な質感を出力できるため，自由な表現を持つラインストーンの印刷が可能となる．UVプリンタを用いてラインストーンを印刷するという研究はこれまでにも存在していたが，既存の印刷手法では，半球形状の造形のみが可能であった．そこで，本プロジェクトでは，多様な形状のストーンを造形する新たな手法を開発した．また，より美しい輝きを持つラインストーンを目指し，ミラー素材を用いたり，底面パターンを用いたりといった工夫を重ねた．その結果，**図-1**に示すような，リアリティのある強い輝きを持ち，さまざまな色の反射の変化などをもったラインストーンの印刷が可能となった．

図-1　ラインストーンの作例

ラインストーンの印刷には，レンズ造形手法を応用し，透明インクを重ねて印刷することで造形する．具体的には，上面が平らになるように一定の比率で図形を縮小して重ねて印刷し，最後に表面の積層痕を埋めて滑らかにするために光沢印刷を行う．光沢印刷は一定量の透明インクを噴霧し，しばらく時間を置いてから紫外線を照射してインクを硬化させ，表面に滑らかな光沢を出すUVプリンタの印刷方法である．その結果，形状自体はきれいに造形することができ，星の形状では鋭角のエッジを表現することが可能である（**図-2**）．

図-2　ストーン形状の一例

また，誰でも手軽にラインストーンを使ったデザインを可能にするアプリケーションの開発も行われた．アプ

リでは，事前に用意されたさまざまな形状のラインストーンを画面においていくことで手軽に好きなデザインを作ることができる．また，3Dプレビュー機能により，実際に印刷されるラインストーンの見え方を事前に確認することもできるようになっている．

子どもから大人まで幅広い年齢層のユーザに実際にアプリを使ってもらい，ラインストーンのデザインを体験してもらうワークショップも開催した．その結果，人や動物などの絵を大きく表現した作品や，文字を表現した作品，模様を表現した作品など，とても多彩な作品が制作された．印象的な作品には，直径7mm以上の大きなラインストーンを利用して車のライトやタイヤを表現したり，大きな異なるカラフルなラインストーンを規則的に並べることで模様を表現したりといったデザインもあった．また，最も多くラインストーンを使った作品には，1.0mmという非常に小さいライントーンを1,000個以上使った，市販のラインストーンを用いた手作業の制作では困難な作品もあり，本プロジェクトのシステムならではの表現が生まれた．

本プロジェクトは，岡が採択基準としている人々の創造性を広げる技術に合致し，クリエータの島元さんのラインストーンに対する情熱とセンスが注がれたプロジェクトであった．

成果発表会では，ワークショップ参加者が作成した作品に加えて，島元さんがデザインしたさまざまなラインストーン作品も披露され，その独自の創造性と多様性を通じて，新たな表現手法の可能性が広がったと感じている．

（担当PM・執筆：岡 瑞起）

[統括PM追記] ラインストーンのデコでスマホを飾っている人はたくさんいると思うが，これを作るのは実はかなり大変な作業ということを初めて知った．UVプリンタは高額だし，そもそも利用料も高いから，失敗する前にアプリだけで簡単にデザインできるこのシステム（まだ名前はないようだ）は，喜ばれそうだ．私は山梨県の名産である印伝が好きで，財布などはそれにしているが，印伝は鹿革に漆をあたかもラインストーンのようにあしらった伝統技芸である．島元さんは印伝の現代版を手軽にデザインできるようにしたと言える．

（2023年7月3日受付）
（2023年9月15日note公開）

スマートフォン向けにカスタマイズが可能なサイレントスピーチインタフェース

情報処理学会・学会誌「情報処理」
2023年9月15日 08:17

蘇 子雄 （そ しゆう）
方 詩涛（ほう しとう）

Amazon AlexaやApple Siri など，誰もが簡単に使える音声インタフェースは，特別な訓練や知識を必要としないため，スマートデバイスだけでなく，テレビリモコンからカーナビ，家電とさまざまなものに広く活用されている．ただし，この音声インタフェースにも弱点がある．自らが発話でき，騒音がなく，また他者の音声を拾わない環境条件を必要とする．

たとえば，電車の車内，静粛な図書館，または掃除機をかけながらでの音声入力を想像すれば，その制約を理解してもらえるだろう．このような環境下では音声入力は活用できない．

蘇さん，方さんは，この問題を解決する，つまり，音声入力の自然さを保ったまま無発声で活用できる，サイレントスピーチインタフェースを，スマートフォンなどの携帯端末で利用できるシステムを開発した．例えるならば，カメラを活用した「口パク」での端末操作を可能にする，読唇術インタフェースである．

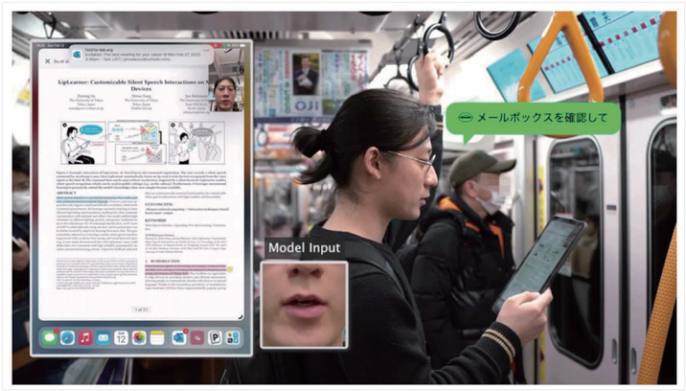

図-1　iPadでのサイレントスピーチ操作例

サイレントスピーチインタフェースの実現のためには，口唇の動きを認識するリップリーディングのAI技術が必要になるが，すでにLipNetを始め，Lip-Interactなどさまざまな取り組みが行われている．ただし，スマートデバイス上で簡単に利用できるようにするのが大変難しい．コマンド数や事前の学習に手間がかかってしまうのである．実際，Lip-Interact では 44 個のコマンドを区別するために，それぞれ 28 回，合計 1,232 回の発話による学習が必要になってしまう．これでは現実には使いものにならない．

蘇さん，方さんは，これを学習済みのモデルを特徴抽出器として，入力された口唇映像をエンコードし，ワンショット転移学習（One-shot Transfer Learning）を加えることで，既存のリップリーディングシステムの課題を解決した． ユーザは，1 回分の発話データを提供すれば，使いたいコマンドを登録でき，誰もがいつでも簡単に利用できる，「口パク」操作をスマートフォン iOS 上で実現したのである．

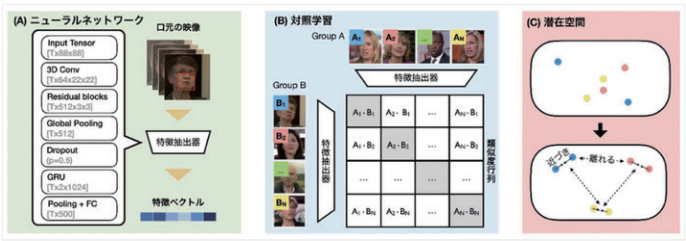

図-2　対照学習パイプライン（自己教師あり学習で有用な表現を抽出）

サイレントスピーチインタフェースは，多数の登録済みコマンドへの無音声UIによる操作を実現しており，カメラ読唇，リップリーディングによる不特定多数に対するあらゆる発話の文字起こしではないことは，誤解のないようお願いしたい．

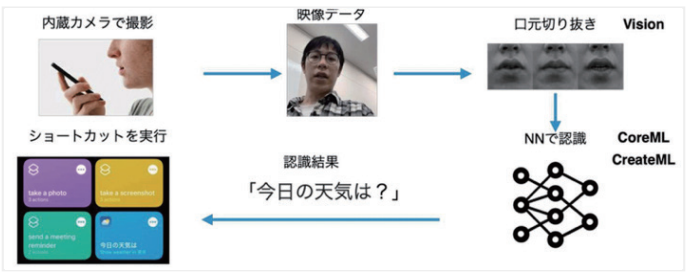

図-3　iOS アプリの実装

サイレントスピーチインタフェースは，従来の仕組みでは難しかった課題をわずか1サンプルのデータで認識可能にし，5つのサンプルでは，98.75%の精度を達成している．決められたコマンドの音声認識でなく，ユーザ自身で登録したいコマンドをiOS上で簡単に作成することが可能になっている．語彙数の制約もなく，さらには複数の言語にも対応できていることは，想定以上の素晴らしい成果である．

成果報告会では，日本語，英語，中国語でのデモに加えて，iPadでのスライド作成のデモを行ってくれたが．第3の手のように，カメラと口をうまくつかって，ショートカットを効率的に利用する姿は，新たな可能性を感じさせてくれた．

図-4　音声によるコマンド登録のパイプライン

最後に，本プロジェクトが，中国出身クリエータ2人のみからなるプロジェクトであることにも触れておきたい．蘇さんは，河北省出身．日本のアニメ・ゲーム好きから，千葉大への交換留学を経て，2019年に東大大学院へ．方さんは，浙江省出身．日本映画好きで，早稲田大学への交換留学を経て，2020年に東大大学院へ入学している．2人が，未踏OBも多い暦本研で知り合ったことが，中国籍2名での未踏へのチャレンジへと繋がっている．

未踏プロジェクトでは，会議などコミュニケーションは日本語が中心．2人は流暢な日本語で中国での事例などさまざまな情報を共有してくれたのだが，やはりそんな2人であっても，日本語ネイティブ同士を前提とした会議環境には，とても苦労したとのことである．

PMとしてもっと配慮すべきだったと大いに反省しているが，プロジェクト終了後に，彼らの口から，米国での会議はもう少し Inclusive な雰囲気がありますよねとのコメントを聞き，はっとして言葉に詰まってしまった．日本流のコミュニケーションが，礼儀正しすぎるからかもしれないが，2人のように，日本好きで日本で活躍したいと考えてくれる，海外からの優秀なクリエータに対して，ダイバーシティとイノベーションを志向する

未踏コミュニティこそ，どこにも負けないInclusive な環境を提供すべきであろうと思う．
蘇さん，方さんだけでなく，みなさんが海外からの留学メンバなどと出会う機会があれば，サイレントでなく，ぜひ積極的に話しかけてほしいものである．

（担当PM・執筆：藤井 彰人）

[統括PM追記] 藤井PMがお書きになっているように，2人は日本文化をこよなく愛している．未踏がこういう想いをしっかり受け止められる器になったと感慨深い．実際，蘇さんは，アニメ好きが昂じて，2023年度の未踏アドバンスト事業に，アニメ制作現場でのデータ資産を有効利用できるAIというテーマで共同イノベータとして採択された．

（2023年7月3日受付）
（2023年9月15日note公開）

翻訳IME「Konjac」とInput Method抽象化レイヤ

情報処理学会・学会誌「情報処理」
2023年9月15日 08:18

竹村 太希（たけむら ひろき）

竹村さんは大学のゼミのSlackで英語話者とリアルタイムにコミュニケーションをする必要に迫られ，キーボードから日本語をひらがなで入力し，適切な漢字かな交じり文を選択すると，翻訳された英文を入力する翻訳IME「Konjac」を開発した．機械翻訳をIME（Input Method Editor）というインタフェースで提供することで，ユーザはどんなアプリケーションでも煩わしさを感じることなく自然に外国語を入力できるようになる．この翻訳IMEの開発の過程で，各プラットフォームにそれぞれ存在するInput Methodのインタフェースを共通化するIM抽象化レイヤ「IMPlane」と通信プロトコル「IMProtocol」も開発した．これにより，アプリケーション開発者が変換ロジックの実装のみにフォーカスして，簡単にマルチプラットフォーム対応の独自IMEを実装できる．IMとIMPlaneとIMProtocolとIMEの関係を**図-1**に示す．

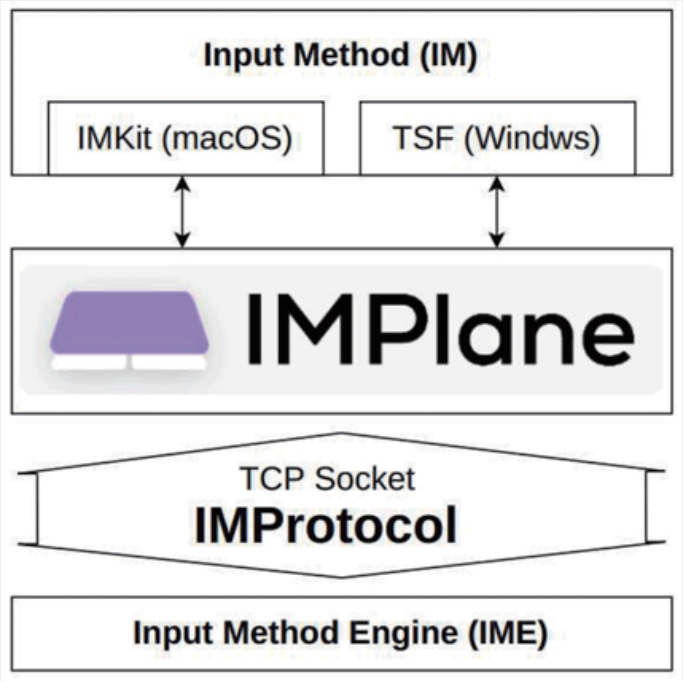

図-1　IMとIMPlaneとIMProtocolとIMEの関係

IMPlaneはデーモンとして起動し，OSのIMに接続してIMEとして振る舞う．その後，IMPlaneとIMProtocolに対応した各IME間でTCPコネクションを確立する．日本語IMEの通信フローを**図-2**に示す．

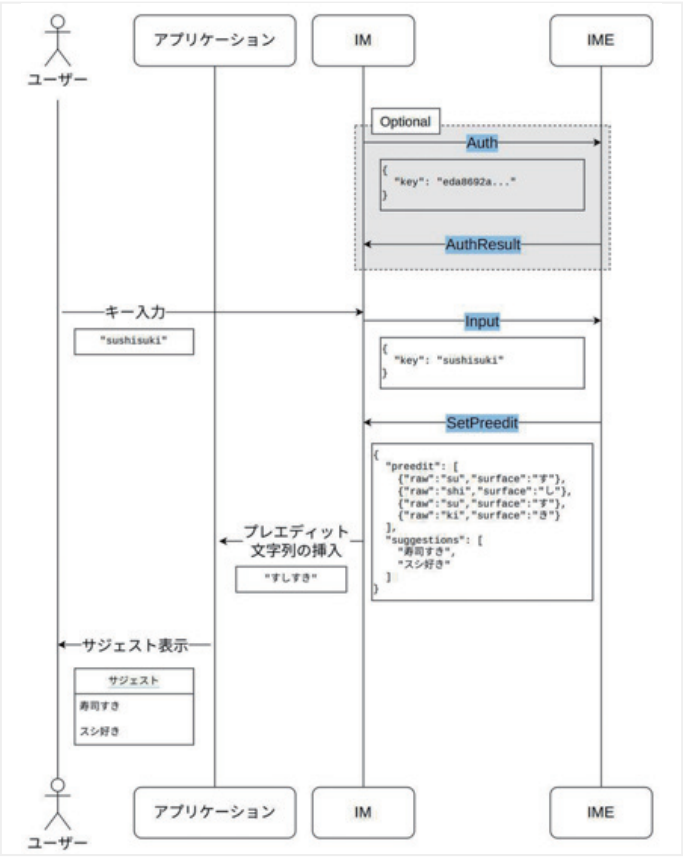

図-2　日本語IMEの通信フロー

ユーザからのキー入力が発生したタイミングで，IMがIMPlaneにInputリクエストを送信し入力を通知する．ユーザが変換キーを押すとConvertリクエストを送信する．ユーザが変換候補を選択した際にはSelectedリクエストを送信する．また，IMEからSetPreeditレスポンスを受け取ると，アプリケーションウィンドウ上にプリエディット文字列を描画する． SetCandidatesレスポンスを受け取ると，変換候補を描画し，ユーザに選択を促す．Doneレスポンスを受け取ると，確定文字列をアプリケーションへ渡し，状態をリセットする．

この仕組みを応用して，日本語のかな漢字変換の部分を機械翻訳のDeepL APIに置き換えたりすることで，独自の翻訳IMEを開発することができる．もちろん日本語・英語以外の言語にも対応できる．竹村さんは自分自身で発表されたばかりのChatGPT APIとIMEを連携できるモックアップも開発し，デモでは大変好評だった．Web APIや特定のアプリケーションのプラグインではなく，IMEのレイヤで実装されているため，SlackやDiscord以外でもオンラインゲームのチャットにも使うことができる．IMEの開発はOSごとに独自に実装しないといけない部分が多く，ハードルが高かったが，IMPlaneを利用することで，関西弁変換や，自動おじさん構文変換など，気軽に独自のIMEを開発することができるようになる．私自身も独自のIMEを作りたくなった．

現在のIMPlaneはmacOS向けの実装が完了している状態であるが，開発途中のWindows，Linuxへの移植作業や，独自IME開発者向けのIMPlane，IMProtocolのドキュメント整備など，今後の開発の動向をウォッチしたい．

（担当PM・執筆：竹迫 良範）

[関連URL]
https://implane.kekeho.net/

[統括PM追記] Konjacは，ドラえもんの翻訳こんにゃくにちなんで，「こんにゃく」と読む．入力したら，ユーザが途中の処理を意識することなく，目的とする外国語（のみならずオジサン言語）に翻訳した出力を出す，そのために異種OSの差を吸収するIMPlaneとIMProtocolという抽象化レイヤを設けるという発想はいかにも自然だが，実装はかなり大変だったようだ．恰好いいのだが，レイヤ間のやりとりが増えて遅くなるからだ．そこで投機的実行などのテクの登場である．こういった表に出ない苦労がスーパークリエータの格好良さでもある．

（2023年7月3日受付）
（2023年9月15日note公開）

抜かない型の設計支援ツールKatalystによるものづくりの自在化

情報処理学会・学会誌「情報処理」
2023年9月15日 08:20

皆川 達也（みながわ　たつや）

型成型技術は大量・安価に成形品の生産が可能であり，材料選択の自由度も高いため，日常的に使うプラスチックや金属，シリコンなどでできた数多くの製品の製造に用いられている．しかし，成形品は型から抜けなければならないという強い製造上の制約を持つ．皆川さんは，これまで型成形では難しいとされていたカタチを成形できる型を生成するための設計支援ツール「Katalyst」を開発し，実際に Katalyst が生成した型で型成形を行った．この設計ツールの名前には，型（Kata）を使ったものづくりを自在化するための触媒（Catalyst）という意味が込められている．Katalystは，CADソフトウェアの1つであるRhinoceros 3D上で動作する．

型成形は3D プリンタと比較して多くの材料を活用できる利点があるが，作ることが難しいカタチとそのための型設計の手間が存在する．本プロジェクトでは2つの要素によりそれらを解決した．1つ目は「作れないカタチをなくす抜かない型（中子）」である．完成品から最後に溶かして型を除去することで，成形品のカタチの制約を緩和することができる．2つ目は「型設計の手間をなくす型設計支援ツール」である．作りたい形状の3D データを入れると，作りたい形状を作る型の3D データが半自動で出力され型設計の手間が軽減する．未踏期間では Katalyst の開発を行い，それを駆使することで従来の型成形では難しいとされていた「入り口と出口のある空洞」「入り口の狭い空洞の形状」「パイプが自己交差する形状」「複数の要素が繋がった構造」を型成形によって作ることが可能となった（**図-1**）．

図-1　抜かない型とKatalystを使った制作物
（左上）醤油差し （右上）うさぎ入りのフラスコ （左下）クラインの壺 （右下）鎖

皆川さんは未踏採択時，ものづくり歴7年であり，ありとあらゆる工法や素材を使ったものづくりに興味を持ち，レーザカッターや CNC を利用したコンピュータでのものづくりはもちろん，フライス盤や旋盤などを使った機械工作の経験もあった．そんな中，皆川さんが次に注目したのが型成型だった．従来の型成形は「型を用意して，材料を入れて固め，その型から取り出さないといけない」という前提があったが，皆川さんはその前提を覆し，抜かない型を前提とした型設計支援ツール「Katalyst」を完成させた．これにより，型成形による自在なものづくりができる社会の実現可能性を見せた．また，材料も型成形の利点を活かし，チョコレート，アクリル，エポキシ，ポリウレタンなど，複数の材料を使って試していった．形状も完成している物体にさらに中子を取り付け，型成形を行うなど，斬新なアイディアで次々と成果物を増やしていった（**図-2**）．

図-2　成果報告会でのネームストラップは抜かない型を繰り返し使って作成した

未踏期間終了後も，茨城県つくば市のものづくり系のイベントに参加し，企業の方とも交流しながら，どういった製品に使えたらよいかを議論したり，作れたら嬉しいものについて意見をもらったりと，検討を進めてきた．また，これまで行ってきた型成形を活用したセンサの製品化も進めているところであり，今後の展開にも期待している．

（担当PM・執筆：五十嵐 悠紀）

[関連URL]
https://sites.google.com/view/katalyst-fab

[統括PM追記] 少しイメージが掴みにくいかもしれないので補足しよう．通常の型成形では，型を壊さないために，成形品をうまく抜けるようにするために作れる形に制約があった．図-1にあるようなものを作りたい場合はいくつかの部品を作ってそれを貼り合わせていたわけである．しかし，たとえば図-1の醤油差しの場合，内側の空洞部分を，ロウなど溶ける中子で作成して型の中で中空に浮かぶように設置する（このためには，中子自体を型成形し，中子を入れた型自身を複数個作って貼り合わせる必要がある）．こうして材料を入れて固めてから，お湯などで中子を溶かすわけである．溶かした中子は再利用可能である．なお，南部鉄器などでは砂と粘土で作った中子を壊して取り除いている．なお，溶ける型でも精度は十分に出るとのことだった．産業に使えるようになることを期待したい．

（2023年7月3日受付）
（2023年9月15日note公開）

ラップバトル対話システム

情報処理学会・学会誌「情報処理」
2023年9月15日 08:23

三林 亮太（みばやし りょうた）

ラップバトルとは，ラップミュージックのスタイルの1つであり，複数のラッパーが相互に技術や表現力を競い合う音楽的な対決の形式である．通常，複数の参加者が交互にラップを披露し，その内容やフロー，リズム感などが審査や観客の評価の対象となる．ラップバトルでは即興性が求められ，参加者はリアルタイムで自身のラップを構築し，相手に対して効果的なフロー（リズム）とライム（韻）を踏んだアンサー（返答）を用いて応戦する．競技としてのラップバトルは，審査員やオーディエンスの前で行われ，勝敗が決定されることもある．また，ラップバトルはエンタテインメントの要素も含まれており，ラッパー同士の熱いバトルやパフォーマンスが観客を魅了する．

本プロジェクトの提案者である三林さんは，溢れんばかりのラップバトルへの情熱をもとに，自身の計算機科学の能力を活かして計算機をラッパーとして参戦させるという挑戦的な試みを行った．このプロジェクトは，従来人間によって行われてきたエンタテインメント領域において，計算機による新たな可能性を開拓する意欲的な取り組みとして注目される．特に人間同士が当たり前であったラップバトルの世界において，対戦相手が計算機になったり，両方が計算機になったりというのは，体験価値としても新しい．

ラップバトルでは，ライムの質やアンサーのバリエーション，そしてフローの印象などが重要視される．これらの要素を計算機が考慮しながら，相手の即興ラップ（バース）に応じた戦略的な反応を返すことは困難な課題と言える．しかし，ライムの分解における母音の判定や，バースの選択における適切なボキャブラリーの探索といった点では，計算機の得意分野が活かされる．その実現のため，本プロジェクトにおいては，実戦デモシステム，バース生成システム，ラップ音声合成システムの作成の，大きく分けて3つのシステムを実装した．

最初にバース生成システムを解説するが，これは入力バースに対して，「ライム」と「アンサー」を考慮した返答バースを生成するテキストベースのシステムで，深層学習モデルを用いてラップバトルコーパスからラップ生成を行う手法を提案した．

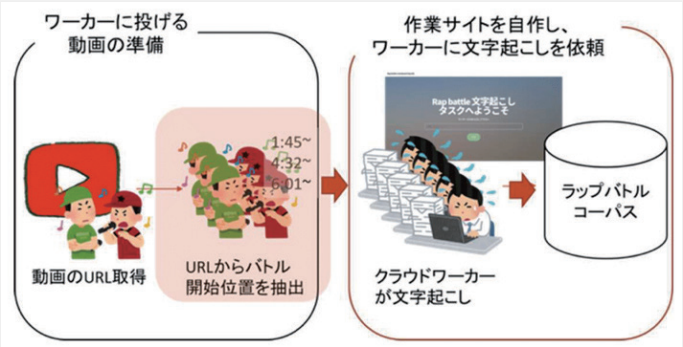

図-1

ラップバトルコーパスは，ラップバトル動画内のバースを文字起こしして構築した．このため，クラウドソーシングを活用し，ラップバトル動画の文字起こしを行った．ラップバトルコーパスは約1,780件であり，バース単位で計算すると9,749バースを含んでいる．

ライムを考慮したラップ生成手法では，逆向き生成による方法を採用した．Transformer Encoder-Decoderを用いて，逆方向からの文生成を行うことで，文末にライムを考慮したラップを生成する．また，アンサーを考慮したバース生成システムについても，アンサーの対応関係を表した学習データを作成し，独自の評価関数によるフィルタを加えて開発した．これにより，相手の発言をふまえた返答をすることができるようになり，ラップバトルの楽しさをより一層高めることができると考えられる．

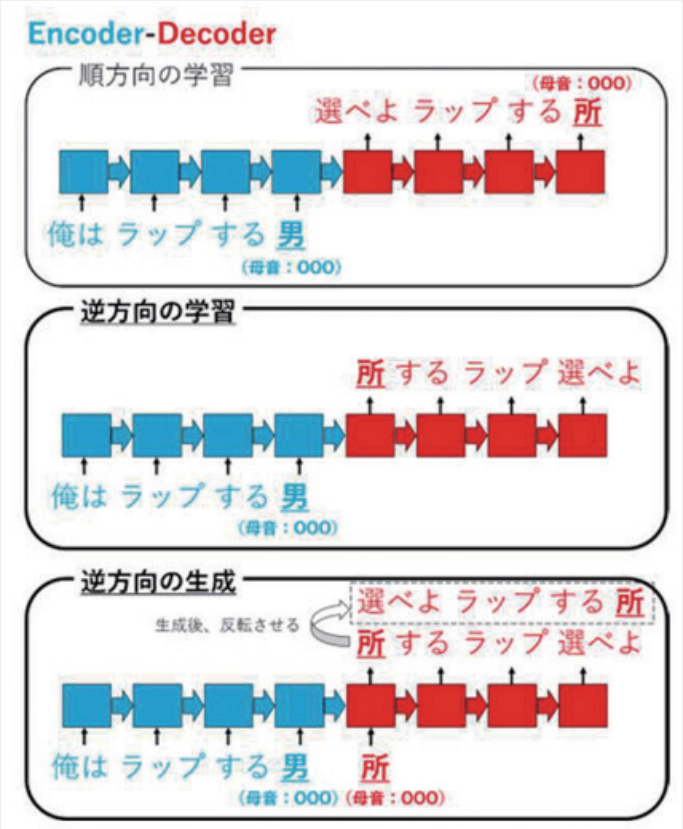

図-2

次に，音声合成システムであるが，これは生成された返答バースをラップ音声に変換するシステムである．Tacotron2とGriffin-Limアルゴリズムを用いて実装し，ラップ調の音声合成が可能となった．また，ラップバトルに合わせた高速な音声合成や，生成音声の強制的な推論切り上げなど，即興に対応した実装も行った．

成果報告会では，対応が間に合わず，CoeFontのラップ音声合成APIを一部利用したが，その後ラップ音声コーパスを自作し，完全に自作のラップ調音声合成を開発した．このラップ音声コーパスは，324文のテキストに対してラップ調に読んだ音声データの集合である．ラップに特化した音声コーパスは存在しないため，自作する必要があった．本コーパスは，三林さん自身がラップ調に読んだオリジナルのコーパスである．

最後に，実戦デモシステムであるが，これはバース生成システムとラップ音声合成システムを用いた，実空間でラップバトルが体験できるデモシステムである．このシステムは，ハードウェアとソフトウェアの両方で構成され，ハードウェアにはPC，ミキサー，スピーカー，有線マイクを搭載した折りたたみ式台車が使用され，ソフトウェアにはラップバトルが可能なWebアプリケーションが使用されている．

ハードウェアは，持ち運びや人間とのインタラクションを考慮した設計となっており，台車には予備バッテリーを搭載し，周囲に荷物が残らないようにカバーを設置した．また，ホイールをエラストマー素材に差し替えることで騒音を軽減し，新幹線や電車等の公共交通機関を利用する場合の持ち運びにも配慮した．

本プロジェクトにおいては，PMが採択基準としている大きな情熱と溢れるクリエイティビティにピッタリと合致し，クリエータ自身が楽しんでプロジェクトを遂行できたことが素晴らしい．

成果報告会の発表前日には，実戦でもシステムを片手に別のクリエータと共に秋葉原の街中でラッパーと実際に対決し，相手も計算機との初対戦に終始新鮮な反応を示していたことが印象に残る．

成果報告会においても，発表者である三林さん自身が本システムとの対決を行い，絶妙なやりとりを発表することができ，とても充実したプロジェクトであったと感じている．

（担当PM・執筆：田中 邦裕）

[統括PM追記] 三林さんを見ていると，本当にラップ大好きということがよく分かる．実は彼は2018年，2021年にも未踏に応募していて，2021年もラップバトル絡みの提案だった．分厚い準備を進めてこそ達成できた，まさに3度目の正直でのスーパークリエータである．田中PMの紹介にもあるように，成果報告会の前夜に装置を転がして秋葉原に繰り出すという勝負心も素晴らしい．

（2023年7月3日受付）
（2023年9月15日note公開）

フィギュアスケートの練習を支援するSkate Jump Board

情報処理学会・学会誌「情報処理」
2023年9月15日 08:29

山形 昌弘（やまがた まさひろ）
麻 大輔（あさ だいすけ）

本プロジェクトのクリエータらは，学生時代にフィギュアスケートの競技者として，練習や後輩への指導に向き合ってきました．彼らが練習や指導に取り組む中で，特にジャンプの習得に苦労したようです．まず，ジャンプは一瞬なので，頭で考えている暇がない．次に，着氷は後ろ向きにしかできないので，徐々に回転を増やしていく練習ができない．最後に，作用反作用の法則により，空中では，各部位の動作を独立にコントロールできず，たとえば「右足以外はそのままで，右足の動きだけを変える」ことはできないわけです．これらの課題により，指導の言葉で直すべきポイントを理解するだけでは，実際には直せないという壁に何度も直面してきたようです．

そこで本プロジェクトでは，スマホで練習を撮影した定点映像から競技者を追跡し，ジャンプを検出して切り抜き，そのジャンプを加工した映像もしくは画像を生成して見せることで，ジャンプの感覚的な理解を促進させるシステムを当初の目的としていました．

しかし，プロジェクト期間中に映像生成機能を開発したところ，それによって生成された映像の踏切時の動作に不自然な点があることが課題となりました．そこで新たな方針として，ジャンプの一種「ループジャンプ」の踏切動作を，ゆっくり，足の動きをガイドして，踏切動作単体で練習できる装置，Skate Jump Board

（**図-1**）の開発にチャレンジしました．この装置は，スケート靴および氷の特徴と，ジャンプ中に身体に働く慣性力の一部を再現しているため，実際のジャンプに近い動作が可能になっています．また，自分の動きとお手本の動きを見比べるための，お手本表示機能を有しています．

図-1　Skate Jump Boardの可動台（左）とSkate Jump Boardを使った練習の様子（右）

このシステムは以下の3つの要素により構成されています．

1. 傾斜したレール上の可動台
   ループジャンプの踏切では，右足にほとんどの体重を乗せて急カーブを描き，最後に右足の爪先で跳び上がります．この右足の動きを再現するために，傾斜したレール上の可動台に右足を乗せた構造を用いています．また，ジャンプの踏切動作を，ほぼ静止から実際のジャンプに近い速度まで連続的に調整しながら練習することができます．
2. スロープ
   ジャンプ中にはブレーキがかかるので，進行方向に投げ出される慣性力が働きます．これを再現するために可動台のレールを傾けています．

図-2　お手本表示機能の概要

3. お手本表示機能（**図-2**）

この装置には「ジャンプをゆっくり練習できるため，全身の動きを細かく意識できる」という特徴があり，動作中にお手本の動きと自分の動きを見比べる機能により，自分の動作に連動したお手本映像と，自分を後ろから撮ったライブ映像を，目の前のiPadに表示することで自分の動作とお手本の動作との違いを細かくチェックすることができます．

さらに定点カメラで撮影した映像からユーザを自動追跡し，ジャンプ動作を自動で検出して一覧表示するAndroid/iOSアプリを開発しています（**図-3**）．

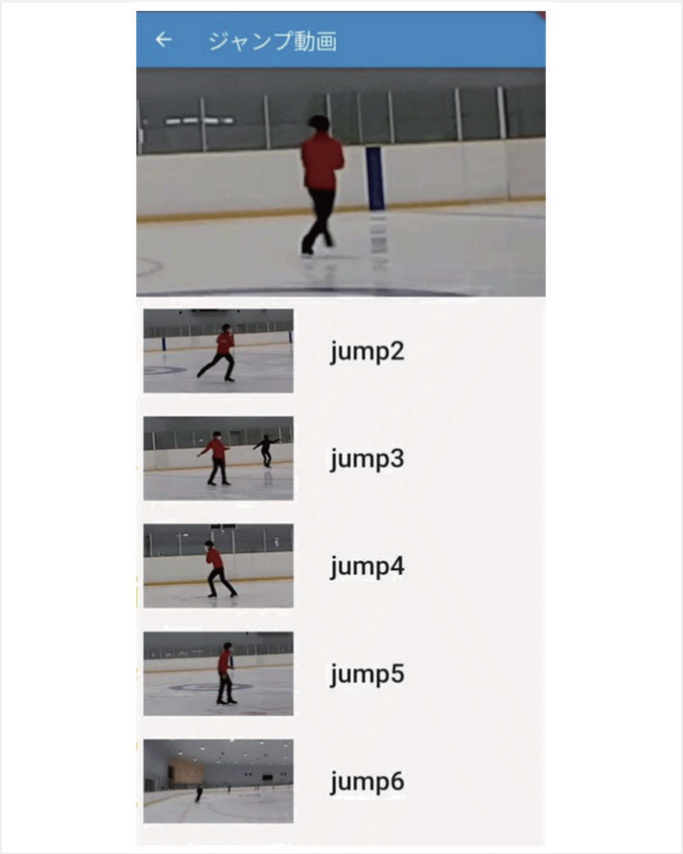

図-3　映像による振り返りを支援するスマホアプリの画面

本プロジェクトのクリエータの1人麻大輔さんは，自身がスケート部に所属し何年かけても成功できなかったダブルループジャンプを自ら開発したシステムで練習し見事成功しました．つまり，未踏クリエータとして貢献したのみならず，自らのスケーターとしての長年の夢を未踏での活動を通し実現したという点で驚くべき，そして喝采すべき成果と言えます．

（担当PM・執筆：稲見 昌彦）

[関連URL]
https://twitter.com/skate_jumpboard

[統括PM追記] 映像解析システムの開発のはずだったが，途中でハードウェアの開発にほぼ重点が移ったという，過去に例のないようなピボットの見事さに感銘した．稲見PMが書かれているようにフィギュアスケートは本当に瞬間芸なので，いくら言葉や映像で説明されても効果が薄く，結局身体，つまり人体ハードウェアで覚えるしかない．説明もぜひ成果報告会動画を見ていただければと思う．なお，Skate Jumping Boardに可動部分はあるが，動力装置がない．まさに，巨人の星ギブス並みの，どこでも練習ツールだ．きっと売れると思う．ならば，ほかのスポーツにも類

似のアイディアがあるのではないかという気がしてきたのは私だけではないだろう.

（2023年7月3日受付）
（2023年9月15日note公開）