

2016年度スーパークリエイータ認定者と実施プロジェクト概要及び評価

(所属は 2017 年 5 月現在)

- 1 石井 翔 氏 (東京理科大学 理学部第一部 応用数学科)
- 2 秋澤 一史 氏 (東京理科大学 理工学部 物理学科)
- 3 大谷 拓海 氏 (東京工業大学 工学院 情報通信系 情報通信コース)
- 4 城山 賢人 氏 (東京理科大学 理学部第一部 応用数学科)

Web 開発のための WebGL フレームワーク「Grimoire.js (グリモア.ジェーエス)」の開発 (正式プロジェクト名 : Web サービス開発フローと親和する Web3D ライブラリ)

Web 制作の現場で高度な映像表現を効率的に開発できるフレームワーク「Grimoire.js」を開発した。近年登場した「WebGL」という仕様に基づいた技術を用いると、Web 上で 3D 映像などの高度な映像表現が可能となっている。しかし一般的な WebGL ライブラリはゲーム開発などの文化に寄った仕様となっているため、Web 制作に携わる技術者がそれを習得するにはギャップが大きいと、結果的に WebGL の利用を断念せざるを得なくなるなどの問題があった。

本プロジェクトでは WebGL を簡便に利用するために、Web 開発者が慣れ親しんだプログラム手法で WebGL をコントロールできるフレームワーク Grimoire.js を構築した。一般的な WebGL ライブラリでは膨大なプログラムを記述していた表現を、Grimoire.js では数行のマークアップで記述することができる。Web 制作に Grimoire.js を用いれば、これまで Web 開発者が慣れ親しんだ記法で、かつ最小限の記述により、自然に WebGL の強力な表現を活用できる。

この成果により、今後制作される Web コンテンツは、3D などのリッチな表現が多数取り入れられることになるかもしれない。

評価ポイント (担当プロジェクトマネージャー 首藤 一幸 氏) :

彼らは妥協をしない。ソフトウェアとしての発展性を確保すべくコンポーネントシステムから開発した。ビルド環境は 5 回ほど作り直した。描画を担うレンダラはゼロから開発した。シェーダ記述言語 GLSL を拡張した。3D モデルのフォーマット glTF の仕様への対応具合は、各種ライブラリの中でトップクラスである。ウェブブラウザ上で動作するインスペクタ (データの木構造を見て変更するツール) も開発した。

当初は、複数人プロジェクトゆえの心配もしていた。大学の同じキャンパスだから出会ったという 4 人が、キャンパスとは関係のないプロジェクトという軸で心を合わせ続けられるのか? 4 人全員が意欲を維持できるか? 音楽性…もとい方向性の違いで仲違いしないか? など。しかし杞憂であり、4 人とも最初から最後まで高い意欲を保って貢献し続けた。初夏の頃にコンポーネントシステムをしっかりと作ったことで、それ以降の分業をスムーズに進められたともいう。

ソフトは出来た。世界を獲って欲しい。

5 片岡 秀公 氏（立命館大学 大学院 情報理工学研究科 情報理工学専攻）

iPhone で手軽に農作物を鮮度測定できる未来を創るソフトウェア「VEGFR（ベジフル）」の開発

（正式プロジェクト名：音を用いた農作物の鮮度計測）

消費者が家庭で手軽に農作物の鮮度計測ができる「VEGFR」を開発した。

消費者が店頭や家庭で収穫済みの農作物の鮮度を知るには、見た目やさわった具合など、消費者自身の不確かな経験を元にするしかない。一方、高度な鮮度計測をする機械は、農作物と物理的接触を伴うもの、赤外線を使うもの、農作物から放出されるガス量を測るものなど多数あるが、どれも高額な特殊機器で、中には測定時に農作物をいためるものもあり、消費者が手軽に利用できる機器は存在しなかった。

「VEGFR」は、鮮度測定に農作物にあてた微弱な音の反射を利用する。使用する機材は、小型のスピーカと、農作物にあてた音を集音するためのマイクロフォンが大人の拳ほどの大きさでワンセットになった箱型のデバイスだ。このデバイスで収集したデータをコンピュータ経由で Web サーバにあげ、サーバ側で解析した結果をコンピュータのブラウザで表示する。鮮度の識別には、機械学習という人工知能の技術を用いている。未踏期間中、開発者である片岡氏（片岡家）が膨大な量のトマトを栽培・収穫し、それらをサンプルデータとして利用した。その結果、トマトの鮮度計測では、平均推定誤差が 0.5 日未満という高い精度を達成した。

本開発により、一般消費者が農作物の「おいしさ」と「鮮度」を正しく、気軽に紐づけて理解することができる社会がくるかもしれない。

評価ポイント（担当プロジェクトマネージャー 首藤 一幸 氏）：

正直に告白すると、トマトの鮮度ってそんなに測りたいものかなあ（測定専用デバイスを使うのも面倒だし）、と思っていた。一方、トマト農家に育った片岡君はあまりに本気だった。100 個を超えるトマトやみかんのデータを毎日採り、新しいデバイスと手法を編み出すたび（4 回）にまたデータを採り、いいデータを採るために専用ビニールハウスを建てた。使い易さのためには、経験のなかったウェブアプリケーションの開発においても妥協することはなかった。本人曰く「信じられないくらいひどい」ユーザインタフェースからスタートし、周囲の教えを請いながら、モダンでわかりやすいものを仕上げた。専用デバイスが要るんでしょ？という私の心配に対しても、スマートフォンだけの測定が出来そうところまで示してくれた。今なら私も、自分の手で野菜の鮮度を測っている未来をイメージできる。

もちろん、0.5 日未満という平均推定誤差は驚異的であり、にわかには信じがたいくらいである。

この通り、片岡君は当初の期待や想定を大きく超えてくれた。また、絶対的に見ても驚きの成果を挙げた。信じて情熱を注ぐことの大切さを教えてもらった気がする。

6 木村 廉 氏（神戸大学 大学院 科学技術イノベーション研究科）

IoT 社会を加速させる、ハードウェア制御用プログラムのデバッグ、テスト、解析プラットフォーム「Kernel Analysis Platform（カーネル アナリシス プラットフォーム）」
（正式プロジェクト名：カーネルソフトウェア開発支援ツール）

デバイスドライバなど OS 内で動作するソフトウェア（以降「カーネルソフトウェア」と呼ぶ）の開発を支援するための開発プラットフォーム「Kernel Analysis Platform」を開発した。

IoT 社会の広がりにともない、制御、接続されるハードウェアは多様化、複雑化している。それらハードウェアは、カーネルソフトウェアを介して OS の中で制御されることで、開発者は容易にそれらを使用するプログラムを開発することができる。しかし、このカーネルソフトウェアにバグがあると、OS を巻き込んだ重大なクラッシュにつながる場合が多いため、カーネルソフトウェアの開発では入念なデバッグやテストが必要である。従来、ハードウェアベンダは自社専用のデバッグ、テストツールを開発し、利用してきたが、標準的かつ効率的に扱えるカーネルソフトウェアのデバッグ、テストツールは少なかった。また、カーネルソフトウェアはハードウェアの制御を伴うため、オフィスソフトウェアやゲームなど、ハードウェア制御を考慮する必要がない OS 上で動くソフトウェアのデバッグに用いられる Timeless Debugging などの、最新で柔軟なデバッグ手法を用いたデバッグツールの存在は皆無であった。

本プロジェクトでは、上記の Timeless Debugging などのデバッグ手法をカーネルソフトウェアの開発にも取り入れた。また、仮想化技術を駆使することで、OS に依存しない柔軟なデバッグ、テストツールを実現した。

この成果により、ハードウェアベンダにおけるデバイスドライバの開発コストが下がり、より多くの機器の登場が期待できる。また、既存のソフトウェアに対しても、本成果を利用することで、多くのバグが発見できる可能性もある。

評価ポイント（担当プロジェクトマネージャー 後藤 真孝 氏）：

デバイスドライバなどのカーネル空間で動作するソフトウェアの開発を支援するためのカーネルソフトウェア開発支援ツール「Kernel Analysis Platform」を木村君は実現した。ソフトウェア開発においてデバッグは不可欠だが、カーネルソフトウェアのデバッグの場合には、通常のユーザ空間でのデバッグを効率化する動的解析ツールの利用が困難という問題があった。木村君は、この問題を解決するために、開発者が利用するユーザインタフェースであるフロントエンドと、カーネルソフトウェアの動作情報を記録する基盤技術であるバックエンドで構成される「Kernel Analysis Platform」を実現することに成功した。フロントエンドとしては、カーネルソフトウェア専用のデバッガ「KlareDbg」および自動テストツール「kvalgrind」の二つを開発した。特に KlareDbg では、従来はユーザ空間でのデバッグにしか用いられていなかった Timeless Debugging というデバッグ手法を実装することで、任意の命令に自在にカーソルを移動して実行時の状態を復元・確認可能にした点が優れている。さらにバックエンドでは、当初の計画では、仮想マシン（VM）で OS ごとエミュレートさせる「QEMU 改」のみの開発であったにも関わらず、実機を用いた効率的なデバッグも可能にした方が開発者の利便性が高いことから、ハイパーバイザにより実機を利用しつつ解析対象のカーネルソフトウェアのみエミュレート（インタプリタ実行）させる「K2E」も木村君は開発した。しかも K2E では、開発者が容易にテストを拡張でき

るようにプラグイン機構を提供し、そのプラグイン開発を円滑にできるように、C++の標準ライブラリまで自力で移植してしまうなど、非常に大規模なソフトウェア開発を木村君は成し遂げた。VMにはQEMU、ハイパーバイザにはBitVisorという既存の優れたソフトウェアをベースにしながら、木村君自身がそれらのソフトウェアを読解して深く理解することで、従来は実現されていなかった改造・拡張を可能にし、二つのバックエンドを実現したことは、当初の想定を大きく上回る特筆すべき成果である。既にGitHubにて一般公開中であり、フィードバックを得ながら開発を進めるなど、カーネルソフトウェア開発を的確に支援する素晴らしい成果をあげた。その木村君の才能と卓越した構想力、達成力、プレゼン力、情熱、開発実装力を、極めて高く評価する。

7 佐伯 学哉 氏（東京大学 大学院 情報理工学系研究科 コンピュータ科学専攻）

8 西脇 友一 氏（東京大学 大学院 情報理工学系研究科 コンピュータ科学専攻）

Linux 上で作成されたアプリケーションを、そのまま macOS で動作させるミドルウェア「Noah」の開発。Linux で作成されたソフトウェアが、主要 OS で動作する環境が整った（正式プロジェクト名：ハイパーバイザ技術を用いたクロス OS な Linux バイナリ互換プラットフォームの構築）

Linux 上で作成されたアプリケーションが、そのまま macOS で動作するバイナリエミュレータ「Noah」を開発した。

現在、多くのソフトウェア開発者は Linux を好んで使い開発を行っている。Linux 環境で開発されたソフトウェアは、ELF バイナリという Linux 専用のデータ形式（以降「Linux バイナリ」と呼ぶ）となる。

Linux、macOS、Windows、FreeBSD といった主要な OS で動作するソフトウェアは、それぞれの OS 上で開発されることが当然であり、Linux で作成された Linux バイナリは、通常はそのままでは Linux 以外の OS では動作しない。POSIX という標準規格に準拠して開発されたソフトウェアであれば、ソースコードレベルでの互換性は保証されるが、規格自体が古いこともあり、十分な互換性が実現できない場合が多い。ソースコードレベルではなく、ある OS 上で作成されたソフトウェアのバイナリを他の OS 上で直接動かすためには、エミュレータや仮想環境と呼ばれる中間処理用のミドルウェアが必要で、それらはいがい処理が低速になりがちである。これまで Linux バイナリを実行することができるミドルウェアが用意されている主要な OS は、Windows（Windows Subsystem for Linux による）と FreeBSD（Linuxulator による）に限られていた。

本プロジェクトで開発されたソフトウェア「Noah」は、Linux バイナリをもう一つの主要な OS である macOS 上で動作させるためのソフトウェアだ。Noah は、ハイパーバイザと呼ばれる最新の仮想化技術を使うことで、高い移植性のみならず、高速にそのソフトウェアを動作させることをも可能にした。

macOS ユーザは、自分の Mac PC に「Noah」をインストールしさえすれば、それ以降その PC では、Linux バイナリがそのまま動作するようになる。そのため、Linux を好んで使う開発者たちは、自身が作成したソフトウェアに「Noah」をバンドルし配布すれば、macOS ユーザにも利用してもらえらるソフトウェアを配布したことになる。

「Noah」により Linux のソフトウェアが主要な 4 つの OS（Linux、macOS、Windows、FreeBSD）で動作する環境が整った。これは、Linux が POSIX に代わる次世代の標準規格となったことを意味している。

本成果は、5 月 31 日～6 月 2 日に日本で開催する Linux の国際会議である LinuxCon Japan（今年度から Open Source Summit Japan に統合）で発表する。その後、技術的詳細をまとめた論文を国際学会に今年の夏に投稿予定である。

評価ポイント（担当プロジェクトマネージャー 竹迫 良範 氏）：

macOS 上で Linux バイナリを無修正で動かすという非常に技術的難易度の高いエミュレーション技術のコンセプトを本プロジェクト期間中に実装し、世界初の Proof of concept を実現することができた。

佐伯氏は 64bit VM の作成、fork + exec に関するプロセスモデルの作成、スレッド関連のレイヤの作成、ネットワーク関連のシステムコール変換層の作成、シグナル機構の構築、

カーネル側のメモリ管理機構の初期実装と高速化、メタ strace などの本システム自体のデバッグ機能実装など幅広く担当し、実装量も多く、熱意と短期間の集中作業が必要で、未踏という枠組みがないと実現できなかった試みであり、このチャンスをうまく掴み利用した。

西脇氏は ELF ロードの作成、仮想ファイルシステムレイヤの構築、ロック関連のシステムコール変換層の作成、時刻関連のシステムコール変換層の作成、仮想メモリ関連の初期実装、共有メモリアロケータの作成、CI サーバの管理と運用、Homebrew のインストールスクリプトや VM イメージの作成・管理など、本プロダクトに関連するすべての周辺作業を巻き取り、サーヴァント型リーダーシップを発揮した。元々過去に言語処理系の開発などに携わっていたことがあり、プロダクトを実用レベルまで仕上げ、地道に課題をつぶし、品質を向上させていく部分でプロジェクトに大きく貢献した。

9 神野 響一 氏（ソナス株式会社 共同創業者）

高品質なセンサシステムを容易に構築可能にするプラットフォーム「United Mesh Network（ユナイテッドメッシュネットワーク）」の開発 （正式プロジェクト名：高品質なセンサシステムを容易に構築可能にするプラットフォーム）

高品質なセンサシステムを、センシングデバイス、親機、ゲートウェイ、サーバというコンビネーションで実現した総合プラットフォーム「United Mesh Network」を開発した。土木、農業、医療等の現場では、①精度の高いセンサによる計測と、②計測結果の解析システムが必要である。たとえば、橋梁の過重にともなう負荷のモニタリングでは、負荷をかけた瞬間に橋梁のどの部分にどれだけの負荷がかかるか、といった同時計測性と、計測数値が欠損なく正確であることが必須である（①精度の高いセンサによる計測）。また、それらの計測が十分に実施された後、その計測データを瞬時にサーバへ送り、計測結果を可視化することが求められる（②計測結果の解析システム）。

IoT 社会の到来に伴い、多くのベンダから様々なセンサシステムのプラットフォームがリリースされているが、複数のセンシングデバイス間と、データを集約する親機との高精度の時刻同期通信が必須要件となるため、サービス利用者はその通信手段を確立するために、回線設計、無線通信、ネットワーク技術といった組み込みに関する専門的な知識が必要であった。そのような背景から、土木、農業、医療等の研究者は、それらのプラットフォームを利用できない場合も多かった。

本プロジェクトの成果である「United Mesh Network」では、「同時送信フラッドイング」という最新のセンサネットワーク技術を用いることで、複数のセンシングデバイス間、及びそれらから親機へデータを送信する際の通信を、プラットフォームに組み込むことに成功した。この開発により、無線通信の専門知識がない利用者でも、このプラットフォームの利用が可能だ。

また、センサシステムの要諦である「精度の高いセンサによる計測」を担保するため、センシングデバイスを無線通信部とセンシング部にハードウェア的に分離することで、高品質な測定システムが実現できた。

本成果により、多くのユーザが簡易にセンサシステムを利用できるようになるため、各種分野の研究が加速すると同時に、ライブエンタテインメント分野など、これまで想定もしなかったようなセンサネットワークの用途が生み出される可能性がある。

評価ポイント（担当プロジェクトマネージャー 石黒 浩 氏）：

1 対 1 の無線通信を可能にする従来の通信方式とは異なり、クリエイターが提案する通信方式は、1 対多の通信を可能にするとともに、多数の無線デバイス間で高速に同期を取れるというメリットを持つ。

クリエイターはこのアイデアをもとに、実際に動作する小型の無線デバイスを作り上げると共に、広い場所で多数の LED を正確に同期させる実用的なアプリケーションを開発した。その実装能力は高く評価できる。またその他にも実用的な応用について様々考察し、開発した無線デバイスの可能性を探究することができ、今後の実用化が大いに期待できる。

10 中村 優文 氏（早稲田大学 大学院 先進理工学研究科 物理学及応用物理学専攻）

11 山口 周悟 氏（早稲田大学 大学院 先進理工学研究科 物理学及応用物理学専攻）

手書き文字を、紙に美しく書くためのソフトウェア「てふみ」の開発

（正式プロジェクト名：手書き文字を美しく書くためのソフトウェア）

手書き文字を美しく書くための総合支援アプリケーション「てふみ」を開発した。文字情報の主要な伝達手段がモバイルやコンピュータになった現代にあっても、冠婚葬祭での署名、履歴書、恋人への手紙など、「美しい」手書き文字を書きたいと思うシーンは多い。コンピュータ上で手書き文字の上達を支援する研究やアプリケーションはいくつか存在しているが、封筒、はがき、祝儀袋、半紙等の「実際に書く対象となる紙」に美しく文字を書くことを支援するものは、これまで存在しなかった。「てふみ」は実際に手書きが行われる場面を想定し、紙に書く文字を美しくするために開発された、唯一のアプリケーションだ。

「てふみ」を利用するには、「てふみ」がインストールされた PC、Web カメラ、小型プロジェクトが必要だ。

「てふみ」の利用開始時、ユーザは祝儀袋、はがき、半紙等の「書く対象」をアプリケーション上で選択する。それから、書きたい文章を、アプリケーション上のテキストボックスに記入し、お手本としたいフォントを選ぶ。このアプリケーションがユニークなのは、お手本となるフォントを任意のレベルでぼかすことができる点だ。こうすることで、ユーザはお手本のフォントの上をなぞるのではなく、「ぼけた」フォントを目安にすることで、自分の文字の個性を保持しつつ、理想となる文字を書くことができる。

アプリケーションでのセットが完了したら、Web カメラで「書く対象」を認識した PC が、「書く対象」に対して最適なレイアウトを解析し、小型プロジェクトで「ぼけた」お手本の文字を表示させる。ユーザは、レイアウト、文字の大きさ、文字のフォーム等、本来手書きをする際に感じるストレスを感じることなく、美しい文字を書くことができる。

「てふみ」には「清書編」と「練習編」の二つのモードがある。使用方法の基本は先述のとおりだが、たとえば書く対象が「祝儀袋」のような綺麗に清書をしたい場合には、「清書編」が用いられる。

「練習編」は、筆や筆ペンを用いた文字の書き方を練習する際に使う。「練習編」で実装したのは「アノテーションフォント」と名付けられたフォントセットと、筆遣い、筆圧の強弱、書き順等が直感的にわかる練習プラットフォームだ。「アノテーションフォント」とは、書道の有段者である開発者が自身で筆書きした常用漢字 2,136 文字と、ひらがな、カタカナのすべてだ。このフォントを使うことで、利用者は「とめ」や「はらい」といった筆の動き等、本来指導者の元で練習する一連の「書道的な動き」を、半紙などの実際の紙を使って容易に学習できるようになった。

このアプリケーションの普及により、手書き文字を書くことの敷居が下がり、よりたくさんの人が「手書き文字」の良さや楽しさを再発見することになるだろう。

評価ポイント（担当プロジェクトマネージャー 石黒 浩 氏）：

手書き文字が下手になっていることは現代人の大きな問題である。パソコンが普及したと言えども、いまだ手書き文字が必要な場面は数多くある。そのような問題を解決するために、クリエイターは、本人の字の特徴を損なわない範囲で、綺麗な字が書けるシステムを提

案し、実装した。その実装能力と粘り強さは高く評価できる。また、さらに字を綺麗に書くための特殊なフォントを常用漢字全てについて開発したことも高く評価できる。

12 怒田 晟也 氏 (筑波大学 情報学群 情報科学類)

「動く IoT デバイス」を手軽に作ることができる開発プラットフォーム「MakeStack (メイスタック)」の開発 (正式プロジェクト名: システムソフトウェア開発プラットフォーム)

簡単に低コストな IoT アプリケーションを開発するための開発プラットフォーム「MakeStack」を開発した。

IoT が社会に根付いたことで、インターネットに接続する機能をもった安価なマイクロコントローラ (以降「マイコン」と呼ぶ) が活発に利用されている。安価で初心者でも簡単にプログラミングできるマイコンや、3D プリンタの登場により、手軽にオリジナルな IoT ハードウェア・ソフトウェア (まとめて「IoT アプリケーション」と呼ぶ) を作ることができるようになった。

一方で、IoT アプリケーションには、ソフトウェアのリモートアップデートやハードウェアの死活監視など、必須で組み込む必要のある制御系機能を求められるが、それらを個別に実装するのはコストが大きい。それらの機能をパッケージ化して提供するサービスは多数存在するが、それらサービスは Raspberry Pi などの一昔前の PC 並の高い CPU 処理性能やスマートフォン並の電力資源を必要とする機器を対象としているため、安価で手軽に手に入るマイコンには利用できないという問題があった。

本成果では、500 円程度で購入できる低性能、低消費電力のマイコンでも IoT アプリケーションの開発ができる開発プラットフォームを実現した。また、そのようなマイコンにインストールできるコンパクトで高性能な OS も実現した。さらに、ハードウェアを動かすための API の仕様も、直感的でわかりやすいものにしたため、ハードウェア開発経験のないソフトウェア開発者にも気軽に「動く IoT デバイス」を作ることができるようになる。本成果により、乾電池駆動で使えるような楽しいデバイスが、たくさん生み出される社会になるかもしれない。

評価ポイント (担当プロジェクトマネージャー 藤井 彰人 氏) :

IoT PaaS である MakeStack は、IoT アプリケーションを簡単安価に開発配備できることがその特徴であるが、それを具現化できた理由はデバイス側の MakeStack HyperVisor, MakeStack OS にある。怒田氏は、プロジェクトリーダーとして全体の開発方針を決めるだけでなく、自身の開発した独自 OS の Reasea を発展改良し、これを物理的制約もある安価な ESP8266 上に実装している。

バックエンドのサービスを中心に、既存技術を組み合わせる IoT PaaS を実現しているサービスが多い中、デバイスからサーバ側まで幅広く知識を有し、最適な形で IoT PaaS を実現したその能力は、高く評価をすべきである。

13 橋本 論 氏 (株式会社 PTP)

次世代ネットワークデータ解析ソフトウェア「Dripcap (ドリップキャップ)」の開発 (正式プロジェクト名 : Web 技術を利用したモダンなパケットアナライザの開発)

インターネット等のネットワークから流れてくるデータ、あるいはネットワークへ流すデータを解析するためのソフトウェア「Dripcap」を開発した。

PC、モバイルデバイス、IoT 機器などは、様々な用途で日々大量のデータを通信している。その際通信されるデータは、経路、使用機器、目的、データサイズ等によって、エンジニアが最適と判断した通信手段を使い、通信手順の設計が行われている。

エンジニアが通信手順を設計する際、やりとりされるデータが滞りなく、最適な方法で流れているかを解析する必要がある。解析のために通信データをモニタリング、可視化するソフトウェアはパケットアナライザと呼ばれる。

現在、パケットアナライザに類するソフトウェアは様々なものがリリースされているが、ユーザが無料で利用できるオープンソースソフトウェアは、実質的に Wireshark のみである。Wireshark は多数のユーザの利用実績があり、多くの通信手段に対応しているが、ソフトウェアの設計が古く、現在の OS やハードウェア環境を十分に活かすことができない。具体的には①機能間のつながりが強いため、特定の機能だけを無効化したり、コアの仕様に変更を加えずに新しい機能を追加したりすることが困難、②ハイパフォーマンスな環境で利用しても、設計上データ処理の並列化ができないため高速化に限界がある、などの問題があった。

本成果は、Wireshark 同様オープンソースソフトウェアとして公開されている。また、機能ごとにモジュール化を徹底しているため、機能の有効化、無効化を容易に実施できるのみならず、機能別カスタマイズや個別開発すら可能な構造になっている。そのため、他のエンジニアが作成した機能モジュールを簡単に利用することができるという利点もある。さらに、現在大部分のエンジニアが利用しているハイパフォーマンスな開発環境で、効率的に動作する設計となっている。

公開して間もないにも関わらず、2017 年 4 月の時点で、GitHub 上で 1400 以上のスターを獲得するなど、急速に多くのエンジニアに利用されはじめた本成果は、パケットアナライザの次のスタンダードになる可能性を秘めている。

評価ポイント (担当プロジェクトマネージャー 竹迫 良範 氏) :

提案当初、JavaScript による高速化実装に大きな課題があり、処理速度の改善数値目標を達成できるかどうか不明であったが、フルスクラッチで大幅なコード書き換えを何度も試すことによって、マルチコア環境で複数スレッドを協調するアーキテクチャを確立し、期待以上のパフォーマンス改善を実施することができた。macOS, Windows, Linux のそれぞれの OS 固有の環境問題に対処し、インストールが容易となる配布パッケージを作成し、完成度の高いプロダクト品質を確立している。クラウド CI 環境を用いて継続的インテグレーションでビルドを自動化するなど、開発環境もモダンな手法を採択している。既にすべての実装をオープンソースで公開し、Electron を用いた新しいパケットアナライザという珍しさと完成度の高さもあり、GitHub の Star も 1400 以上獲得し、英語圏を中心に世界レベルで評価されている。

14 藤坂 祐史 氏（筑波大学 大学院 図書館情報メディア研究科 図書館情報メディア専攻）

授業、講演、取材などの音声記録の振り返りがスマートにできるモバイル端末向けアプリケーション「Recoco（レココ）」の開発

（正式プロジェクト名：音声に視覚的特徴を加えて振り返りを支援するシステム）

音声記録を効率的に「学習・振り返り」するための iOS アプリケーション「Recoco」を開発した。

授業、講演、取材などで、ボイスレコーダーを使って音声記録を残す場合がある。音声記録は、手書きのメモ等と比べて、正確で網羅的な記録が容易にできる。一方、すべての音声を聞き返すのは時間がかかる上、内容を一覧できないという欠点がある。従来のボイスレコーダーは、聞き返すことに煩わしさを感じるため、活用しづらいという問題があった。

本プロジェクトでは、音声記録の学習・振り返りでの用途にフォーカスし、ユーザが振り返りたい内容を含んだ音声のみを効率的に聞き返すことができるように、音声の一覧性・検索性を高めるシステムを開発した。

「Recoco」の機能は、①記録、②再生の大きく2場面に分類できる。ユーザは、音声記録を残したい場で、「Recoco」を起動した iPhone や iPad を机の上に置いて使用する。

「Recoco」は録音しながら、並行して音声認識により発言内容をテキスト化していき、時刻情報によって音声に紐付けて保存する。発言の切れ目を検出し、テキスト化された発言がタイムライン上に区切られて並んでいく。ユーザは話を聞きながら、大切なポイントだと判断した時に、その時点で瞬時にタグ付けを行える。

ユーザは、タイムライン状に並んだ「文字起し」や「タグ」を頼りにして、音声記録の中から特に必要な箇所を見つけ出し、タップすることでその時点から音声を再生できる。長時間の音声では「文字起し」や「タグ」の数が多くなり、把握しづらい場合もある。その時には、タグの種類やキーワード検索で表示を絞り込むことができる。また、タグにはチェックボックスがあり、ユーザは、解決済みの課題や、十分理解できている箇所等のタグにチェックを入れることで、より聞き返すべき「未解決の部分」を把握でき、再生しやすくなる。

このような「振り返りを効率化する工夫」は、従来のツールにはなかった。

「Recoco」は、学習・振り返りの用途にフォーカスし、音声記録をスマートフォンで手軽に活用できることを重視した、まったく新しいアプリケーションといえる。

音声認識の言語は、日本語、英語、中国語に対応しており、ユーザは設定でそれらを切り替えることができる。

「Recoco」は、学習内容の復習に役立つことが期待できる。また、インタビュー取材の記録や、プレゼンの発表練習等、学習以外での活用も期待できる。

評価ポイント（担当プロジェクトマネージャー 藤井 彰人 氏）：

ボイスレコーダーのスマートフォン実装である Recoco は、U-22 プログラミング・コンテスト作品 Recture の単純な機能改善版ではない。プロジェクト期間中には、音声での振り返りが有効なのはどのような場面か、既存ボイスレコーダーはなぜ有効に利用されないのか、といった背景にある活用例を丁寧に一から考え直し、顧客開発型で必要な機能を選び直し再構築している。Recture が持っていた単純なタグ付けだけでなく、文字書き起こし

や、自動区分、タグ候補提示、全体マップ拡縮、検索、カレンダー・クラウド連携などが実装されている。実際に利用してみれば、そのわかりやすさが理解されるだろう。

多くのユーザに受け入れられる Recoco をたったひとりで作り上げたその開発力だけでなく、デザイン志向で実装機能を選別し、UX を設計していく能力も高く評価したい。

15 和田 夏実氏（慶應義塾大学 大学院 政策・メディア研究科）

顔と手の動きにイラストを重ねた視覚情報でコミュニケーションするためのアプリケーション「Visual Creole（ビジュアル クレオール）」の開発

（正式プロジェクト名：手の動きに特化したコミュニケーションツールの開発）

音声言語は、発話と聴覚を利用した意思疎通のツールであり、手話は顔や身体の動きと視覚を利用したものだ。ろう者の使用言語とされる手話は動きや形、位置、量など豊かな視覚情報を含むが、聴者が手話を習得しているケースは少ない。そのため、これまで、ろう者と聴者がリアルタイムにコミュニケーションをとる際、音声言語を基にした筆談や読話などの方法が主に用いられてきた。

「Visual Creole」は、コンピュータ上で動くアプリケーションで、そのアプリケーション上で手の動きを利用した新しい表現を可能にすることで、視覚情報によるコミュニケーションをエンハンスする。ユーザは、「Visual Creole」上で、自分が表現したい物や感情等をイラストやアイコン等の絵で描き、その描画表現とことばと紐づけて登録する。また登録された絵は、自分に重ねて動くよう設定できる。絵を描くことが苦手なユーザは、他者が作った絵のリストから自分のイメージに合う絵を選択することもできる。ことばとイメージが紐付いた状態で登録された視覚言語はそれぞれのユーザが思い描く動きや形等の情報をもつ。この単語を送受信しあうことで感情やイメージを共有することができる。リアルタイムにコミュニケーションをする際には、Web カメラ等でコンピュータの画面に映った自分の姿の上に、登録した線画、イラスト、アイコン等をドラッグアンドドロップで引き出しながらいりとりをする。現在、この開発をベースとして、MP4 やアニメーション GIF のフォーマットで保存したイメージを SNS で共有する機能や、Google ハングアウト等の遠隔コミュニケーションツールの拡張機能としての実装を進めている。

「Visual Creole」の利用が広がることにより、音声に依ることなく、視覚的に世界中の人々とコミュニケーションをとること、イメージを共有することが可能となる日が来るかもしれない。

評価ポイント（担当プロジェクトマネージャー 五十嵐 悠紀氏）：

和田氏は両親が耳が聞こえない環境下で手話を第一言語として育ったという背景があり、「手話」を題材に、ろう者（耳が聞こえず手話を用いて会話をする人）のためだけでなく、ろう者と聴者のコミュニケーション手段、聴者同士のコミュニケーション手段、として成り立たせるシステムの実現に挑戦した。

プロジェクトタイトルに「手話」が入っていないことから分かるように、和田氏は手話を「音を使わず目で理解し手で伝える表現」である「メディア」ととらえており、視覚言語かつ身体言語として音声言語と異なる性質をもち多くの可能性があるという信念が最初から最後まであった。本来、ろう者のコミュニケーション手段であった「手話」を「手の動きに特化したコミュニケーションツール」であると位置づけ、本プロジェクト期間中に作成したシステムをもって、ろう者だけでなく、健聴者や外国人との国を超えたコミュニケーションに発展する可能性を見せた。これは、これまでのコミュニケーションのあり方を変える可能性を秘めており、スーパークリエイターとしての評価基準を満たしていると考えられる。