報告

未踏の第 24 期 スーパークリエータたち

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．この認定は 2018 年で 24 回目となる．突出した才能を持つスーパークリエータを広く産業界や学界に知っていただきたい，というのがこの年次報告の狙いである．

第 24 期の未踏クリエータは計 28 名（20 プロジェクト）で，そのうちの 14 名がスーパークリエータとして認定された．2017 年に引き続き，認定率が 50%となった．

昨年（第 23 期）は，特異的に「低レイヤ」案件が多かったが，今年もある程度それを引き継いだ．また，ハードウェア試作を伴うプロジェクトが相変わらず多い．

例によってクリエータ（代表者）を 50 音順に紹介する．なお，2018 年 2 月 10 ～ 11 日の 2 日間に開催された成果報告会（Demo Day）のすべての動画は
https://www.youtube.com/user/ipajp/
から見ることができる．

門本 淳一郎（かどもと じゅんいちろう），中川 修哉（なかがわ しゅうや），丹羽 直也（にわ なおや），髙橋 光輝（たかはし こうき）

DragonASIC：Web ベースの ASIC 設計プラットフォーム☆1

IoT のために，Raspberry Pi などを用いたプロトタイピングボードを開発したり，汎用 IC または FPGA を中心にした組合せ基板を作ったりというのでは，消費電力，小型化，量産性，デバイスコストで大きな問題がある．やはり，IoT に必要となる小型かつ低電力消費のハードウェア製造には ASIC（特定用途向け集積回路）がベストなのだが，大量生産を前提としているため開発コストが高すぎて，スタートアップ企業などには使えなかった．しかし，最近 3D プリンタの半導体版ともいえる，ミニマルファブ構想が国家プロジェクトとして進められており，サービスインも間近である．これが実現すれば，イニシャルコスト数十万円で，短納期の ASIC が製造可能になる．

ところが，実際に ASIC を開発しようとすると，2 つの壁に当たる．1 つは 1 億円以上のライセンス料が必要な開発ツール，もう 1 つは広範な専門知識が必要なチップ設計の難しさである．

これを解決しようというのが，門本君たちの「チップ設計 as a Service」DragonASIC である．DragonASIC は Web アプリであり，その GUI で必要なコンポーネントを組み合わせるだけで，サーバ側で ASIC の設計が行われてしまう．回路や IC に関する高度な内容は隠蔽されており，ハードウェアの知識を持たないソフトウェアエンジニアでもアプリ指向の ASIC 設計が可能になる．もちろん，高額のライセンス料は不要．これで，数億円と言われる開発費が一気に数十万円と，劇的に下がるというわけである．大量にチップを作れば，1 個あたりのコストは 1,000 円以下だろう．

☆1 https://dragon-asic.jp/

具体的な開発手順は以下の通り．ユーザは（1）使用するセンサを選定する，(2) IoT デバイスの動作内容となるアプリケーションを C 言語で記述する，(3) 書いたコードをブラウザを介して動作させ，記述した IoT デバイスをシミュレーションならびにデバッグする，(4) 完成したコードを確定させ，CPU などの配線データと組み合わせて ASIC の製造データを出力する．この製造データを工場に送れば，ユーザが必要とする最小限のモジュールのみを搭載した ASIC を手に入れることができる．

DragonASIC は Web アプリなので全 OS 対応で，インストール不要なのは当然だが，彼らが独自で開発した汎用 8 ビット CPU TRSQ-8 のデータとそのソフトウェア開発環境が用意されているので，CPU から開発する必要はない．

こんなうまい話，本当にできたの？という気になるが，彼らは実際に，汎用 I/O，温度センサ，照度センサを内蔵した ASIC（図 -1）を製造し，消費電力 0.1mW，12mm 角の農業向け IoT センサモジュールに埋め込んだ（図 -2）．これを 7 個実際に野菜畑にばらまいて，畑の温度と照度を 1 日測定した．デバイスからは BLE（低エネルギーの Bluetooth）で中継機となるタブレットにデータ送信した．ちなみに，従来の農業向けセンサモジュールは 20cm 弱四方の大きな箱である．

4 人のプロジェクトだったが，全員が持分をきちんとやり遂げ，現場導入実験に持ち込めた．まさに期待を超えた成果だった．ドラゴンをあしらったロゴもインパクトがある．ぜひ，Web ページを見ていただきたい．

（藤井彰人 PM 担当）

■神原（かんばら） 志帆（し ほ）

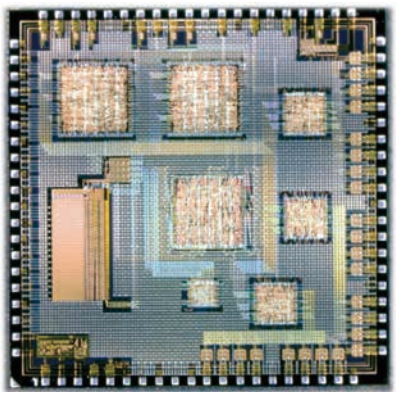
Waybe：Web 閲覧履歴を自動でコンテンツ化する自己表現☆2

神原さんは，文学部思想文化学科で，宗教観の差異の可視化を研究している人だ．だから，人と人とのつながりについての思索が深い．同じ価値観を持った者のみがいる SNS に籠る「島宇宙化」現象を，個々人が閲覧した情報を本人が見た順序や注目度を反映して可視化するのが Waybe である．SNS と違って，自分のものの見方を（受動的に）自動で可視化する新手の自己表現あるいは思考共有ツールだ（表 -1）．

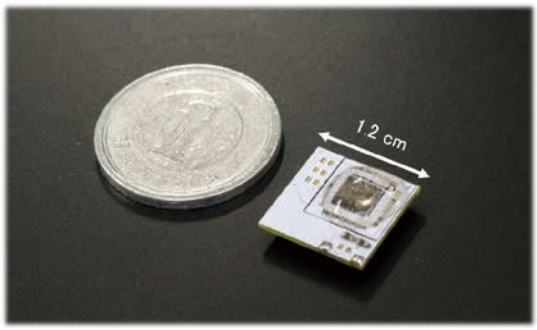
よく知られている SNS と真逆なので，イメージが掴みにくいかもしれない．ユーザがネットサーフィンをすると，ブラウザのアドオンと連携して自動でコンテンツを生成する．システムは，滞在時間，スクロール速度から注目度を算出する．その注目度に応じて，リモートアプリのユーザウォールにユーザが閲覧したコンテンツが，コンテクストごとに表示される．つまり，あの人は最近どんなページを見ているのかが，整理された形で友だちから見える．図 -3 は化学系ユーザの Waybe 画面である．

自分が閲覧したページがみんな透け透けになってしまうという心配はない．単語フィルタ，ログインが必要なページをフィルタするなど，デフォルトの自動フィルタリングの機能が充実しているほか，特定の閲覧や，特

☆2 https://waybe.me/



■図 -1
センサを内蔵した ASIC，3.6mm 角



■図 -2　農業向け超小型センサモジュール

定のドメインの閲覧を公開しないことがブラウザ上のコントロールパネルでいつでも指定できる．面倒そうだが，これでも SNS で自分から積極的に発信をする手間に比べれば圧倒的に楽だ．

このほかに，コンテクストを保持して共有するためのストーリーの検出機能，リアルタイムで閲覧の流れが見えるタイムライン機能，ユーザプロフィール表示機能，人が見たページをブックマークする機能，「実は自分も興味がある」というユーザを発信元に可視化する通知機能，ユーザレコメンデーション機能，あるユーザからの「あまりにも近しい存在である彼女に自分のリアルタイムのアイテムを見られたくない」というリクエストによって実装されたユーザのブロック機能（食べログが彼女の妄想をかきたてて特に危険だったとか）などなど，この小さな紙幅には書き切れない，実に豊富な機能が用意されている．通知機能は発信元に「不思議な快感がある」など，とても評判が良かったようだ．

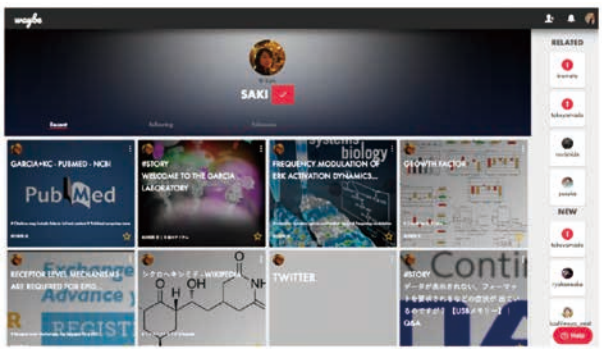
このシステムは 2018 年 1 月から限定公開されたが，1 カ月間で累計 100 名のユーザ，総計 10 万個以上のコンテンツが作成された．ユーザの声は「これは新体験だ，すごい」とか「他人のフィードバックを気にせずに，自分でフィードバックを回せるのが素晴らしい．日記を書くのに似ている．SNS 疲れした人のための空間というのに共感した」といった賛辞が目立つ．

このプロジェクトの最初のタイトルは，視線検知を念頭においた「生体情報を用いてパーソナライズされた要約の自動作成システム」だった．早めにピボットして，より現実的でサービス性の高いシステムに仕上った．未踏はこれがあるから面白い．

Waybe は，ビジネス向けに部署内での情報共有ツールとしてマネタイズすることを含め，これから本気で事業展開していくようである．（石黒浩 PM 担当）

■表-1　Waybeの立ち位置

	アクションの必要性	コンテンツの生成者	情報選択態度
既存の SNS	書いたり押したりする	自分が書いたり撮ったりする	シェアしたいものを選ぶ
Waybe	なにもしない	つくらない	シェアしたくないものを選ぶ



■図-3　Waybe画面の例

■北野 和哉（きたの かずや）

Photon Camera：千億分の一秒の光の拡がりを見る

光が飛んだ距離を測る ToF（Time of Flight）カメラとして有名なのがマイクロソフトの Kinect だが，2017 年末に製造中止になってしまった．Kinect をはるかに超えるピコ秒単位の超短時間に発生する光の現象（ちなみに光は 10 ピコ秒で 3mm 進む）を撮影することが可能なデバイスには大きな応用可能性がある．

サブピコ秒で光を追えるフェムト秒レーザとストリークカメラという組合せがあるが，数千万円から数億円という価格で，測定に 10 時間以上要する．

北野君は，コンピュテーショナルフォトグラフィという研究分野において，これを手軽なものにする技術の開発に挑戦した．目標は時間解像度が 100 ピコ秒以下，可視光から近赤外の 8 ～ 16 波長での計測，640 × 360 画素である．そんなカメラは存在しないので，フォトダイオードとコンピュータの演算によって実現する．

所属研究室での経験があるとはいえ，茨の道が待っていた．フォトン 1 個に超高速で反応する光セン

サ SiPM と，超短時間計測のための TDC（Time to Digital Converter）の組合せをうまく実装すればいいのだが，TDC を自前で FPGA 実装しようとして壁にぶつかった．ミリメートル単位の配線のずれでがらりとタイミングが変わる世界なので，精度が出なかったのだ．結局，性能の良い市販の TDC を使ったが，それと光センサをつなぐ基板の作成が，インピーダンス，基板の材質・厚さなど，アナログ精度の徹底的な追求だった．ノイズ対策も大変だった．

これらに時間を消費してしまい，結果は，なんと 1 画素，光の波長は赤と青の 2 種類のみになってしまった．しかし，目標としていた 100 ピコ秒の 1/10 の 10 ピコ秒の時間解像度が得られたのである．波長数を増やすのはそんなに難しくないし，画素数はマイクロミラーを駆動して増やすという方策を考えている．やはり，このプロジェクトの最大のポイントは時間解像度の追求であった．ともかく，コストパフォーマンスの良い装置のプロトタイプが実現した（図 -4）．

何かとんでもないものができたらしいことがお分かりいただけたかと思うが，これくらいの時間解像度で光が追えると何が可能になるのか．なんと，図 -5 のように見た目ではまったく区別のつかない中空と中身の詰まった同じプラスチックが区別できる．また，白・黒・クリアの混合率が異なる 6 種類の材質を（見た目でも分かるが）きちんと区別できた．これらには数分しか要しない．このほかにも，人体や食品の中の異物なども検出できる．霧を通して向こうが見えるという応用もあるという．

可視光程度のものを使うので，被曝といった心配がない．だから，PET，位相差 CT，MRI などに代わる，人体の非接触・非侵襲計測への応用が期待される．北野君はこのあとも開発を続行していくのでさらなる成果が楽しみだ．（首藤一幸 PM 担当）

■今 悠気（こん ゆうき）

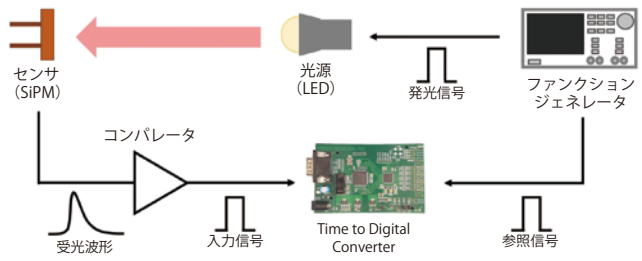
HangerOVER：ハンガー反射を用いた触力覚提示内蔵型 HMD

ハンガー反射という現象をご存知だろうか？　安物のワイヤハンガーを，下辺を前に少し右前とか左前にして頭にかぶせると，その方向に自然に顔が向いてしまうのである．すでに，今君はハンガー反射の権威といっていい．詳しく調べると，左右の首振り（ヨー軸）だけではなく，前後という並進軸，顔を上下に向けるピッチ軸，首を傾けるロール軸もハンガーのかけ方でコントロールできることが分かった．

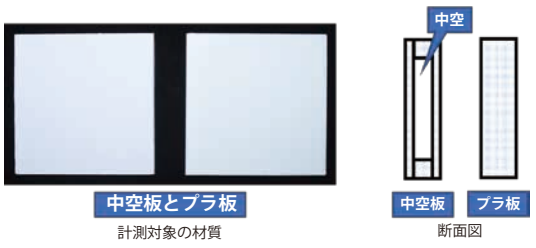
要するに，頭の特定の部位に圧力をかけると，その部位に応じて押されたような触力覚や運動提示ができる．ちなみに，腰周りでも同じような現象がある．実際，今君はこれを利用して歩いている人を操縦できた．

プロジェクトでは，ハンガー反射を応用して，視聴覚に加えて，外界から触力覚・運動提示を与えられるようなヘッドマウントディスプレイ（HMD）を開発した．これで今までにない楽しいバーチャルリアリティ（VR）経験が可能になる．

まず，ハンガー反射をどう実現するかだが，今君は既存の HMD の内側につける鉢巻のようなデバイスの



■図 -4　システムの構成図



■図 -5　見た目にはまったく区別がつかない 2 つの物体

内側 4 カ所に小さなバルーンをつけ，その空気圧を個別に制御するようにした(図 -6)．これにより，ヨー，ピッチ，ロールの「制御」ができる．

これを VR 体験の装置にするために，ゲームエンジン Unity と結合し，ハンガー反射を Unity のタイムラインエディタから制御できるようにした．

こうして装置を開発しても，首が回るという瞬間芸以上のコンテンツを作らないと，なかなか多くの人の関心を呼ばない．プロジェクトの最重要課題は，ハンガー反射を最も効果的に味わえるデモコンテンツの開発だった．今君はいくつかのコンテンツを開発したが，特に良かったのは，VR で味わうジェットコースター（Hanger Coaster）だろう．

単なる HMD をかぶってジェットコースターに乗っている映像を見ると，違和感を覚えるはずである．そう，加速度感がないのだ．大がかりなシミュレータだと，乗っている台を傾けたり，ゆすったりして加速度を実際に与えるのだが，今君の Hanger Coaster は HMD に内蔵されたハンガー反射を使って手軽に錯覚させる．視覚情報によって移動しているような感覚が起こされるベクション（視覚誘導性自己運動感覚）と呼ばれる現象をハンガー反射で増幅するのである．

これは実際に体験しないと分からないと思うが，図 -7 のモニタ画面に写っているジェットコースターに VR で乗っていると，大きく曲がるところでいかにもそれらしい体の動きが起こってしまう．被験者は声を出して楽しんでいる．

もっとも，ジェットコースターの乗り方に応じて，ハンガー反射の与え方が逆になるという知見も得られた．ジェットコースターを楽しいと思う人は，コースターが上に向いたら上を向きたいし，恐いと思う人は，頭を保持したいがため逆に下向きに力が入るらしい．

世の中にはハンガー反射のない人もいるが，デモの前に「自分の心に素直な人は必ずハンガー反射が起こる」と吹き込むと素直に首が振れる．ともかく，人間というのは面白いものだと教えてくれたプロジェクトだった．動画でないと分かりにくいので，ぜひ成果報告会の映像を見ていただきたい[☆3]．（石黒浩 PM 担当）

■坂元 律矛（さかもと りづむ）

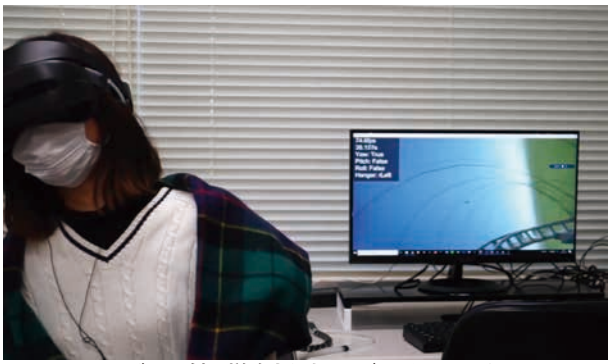
論理的思考を学ぶ迷路作成キット[☆4]

プログラミング教育が小学校で義務化される．その具体的な方法の是非はともかく，STEM 教育（理系教育），つまり理系教育に対する関心が高まり，教材開発が盛んになってきている．坂元君が開発した迷路作成キット Amazeing!! maker は，小学生向けの「考えさせる教材」として秀逸だと思う．

坂元君は 20 歳になったばかりだが，その前の夏休みにこのキットの前身となる迷路を作って子供たちに遊ばせた．すると「自分たちでも作りたい」と言った子が半数いたという．盤を傾けながら金属の球を壁に沿って転がしていくのだが，金属センサがあちこちにあり，

☆3 https://www.youtube.com/watch?v=PMIY0ssUI40
☆4 http://amazeingmaker.com

■図-6　空気圧でバルーンを制御する鉢巻



■図-7　ハンガー反射で増強された VR ジェットコースター

球がそこに乗ると違う場所の扉が開いたり閉まったり，回転ドアが動くなどのギミックがいくつか仕組まれている．迷路のサイズはそれほど大きくないのだが，これで一挙に複雑さが増す．子供たちは，あそこに行けば，こちらが開く，だからえーと，そっちから通って，と「先読み」の論理的思考をしなければならない．

この迷路を子供たちが自ら作れる安価なキットにするのが，坂元君の提案だった．3 種類の小さな板で「直線，カーブ，T 字路，十字路，行き止まり」の 5 種類の基本ブロックをまず作る．通電する特殊ブロックには「ドア，回転床，スイッチ」がある．この特殊ブロックと迷路盤の下にあるマイコンボードのコネクタを結線することによって，マイコンがギミックの動作を制御できるようになる．図 -8 に作成途中の迷路を示す．ブロックをはめこんでいっている様子が分かり，未接続の線も見える．

コネクタにはプリセットされた動作を表現するものもあるが，Google Blockly に基づいたビジュアルプログラミング言語で制御を変更することもできる（図 -9）．LED を点灯したり，ある時間だけ動きを止めたり，といったこともできる．

迷路を解くではなく，迷路を作ることは立派なプログラミングである．迂闊に作るとどん詰まりになって解けない迷路になるし，簡単すぎてつまらない迷路にもなる．「設計」をして，組み立てとデバッグのサイクルを回さないといけない．2 人の子供で作らせると，まさにペアプログラミングの様相になるという．ギミックの使い方，迷路の作り方の約束事など自明ではないが，ガイドブックは分かりやすく作られている．

この迷路キット，小学生に遊ばせる（勉強させる?）だけではもったいない．実際，ワークショップでは，親のほうがしびれを切らせて遊び始め，はまってしまったこともあった．迷路の外枠を大きくすれば，中学生以上でも十分楽しめるようになる．

坂元君はこれをなんとしても製品化したいと考えている．木材ではなく，カラフルなプラスチックの試作品も作っている．ぜひ，どなたか支援していただきたいと思う．このプロジェクトは成果報告会のビデオを見ていただいたほうが，動画もあり，坂元君の元気の良さが一目瞭然である．（五十嵐悠紀 PM 担当）

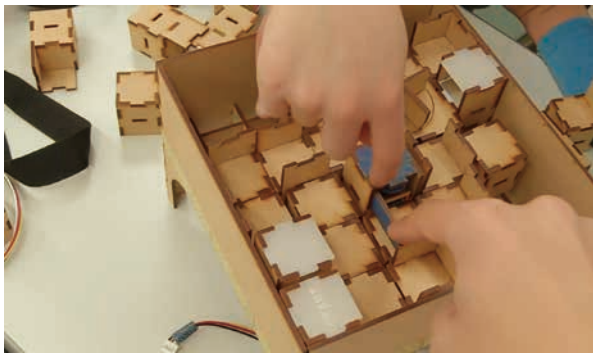
■ 城倉（しろくら） 弘樹（ひろき）

susanow：自動最適化する高性能通信基盤☆5

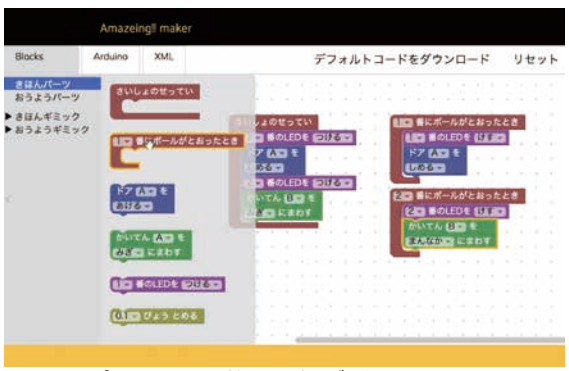
汎用 PC の高性能化に伴い，従来，専用ハードウェアを用いて実現されていたルータやファイヤウォールなどのネットワーク機能が汎用機上のソフトウェアや仮想化技術で実現されるようになった（NFV, Network Function Virtualization）．こういった機能を VNF（Virtual Network Function）と呼ぶ．

城倉君の提案は，10/40GbE 以上の高性能な VNF の開発を支援し，VNF を自動最適化する仮想ネットワークの実現である．達人に頼らないネットワーク構築

☆5 https://github.com/susanow/susanow



■図-8 作成途中の迷路



■図-9 プリセットコードとコードエディタ

だ．こうしてVNF開発において高性能化と機能実装が分離される．

城倉君は学部4年生，しかも，高校3年までは，ずっとプロテニスプレイヤを目指して，坊主頭で頑張っていた．彼曰く，ラケットからパケットに転向したという．

彼が開発したシステム全体をsusanow(スサノヲ)と呼ぶ．まず，VNFの開発を支援するフレームワークがあり，通信基盤はこれで開発されたVNFを適宜監視し，インターネットの環境に対して自動的にチューニングを最適化する．図-10にsusanowのアーキテクチャを示す．D2エンジンはsusanowの通信基盤の重要なコンポーネントである．

パケット処理のバックエンドには，汎用PC上で10/40GbEの通信を可能とするフレームワークDPDKを用い，開発の難易度が上がる部分をフレームワークによってできる限り隠蔽した．それにより，10/40GbE以上のVNFの開発サイクルを高めるだけでなく，VNFの性能チューニングをリモートから行えるようになった．

DPDKを用いても必要になる高度な技術は上述のD2の中に隠蔽したので，VNFの開発コストが劇的に下がった．またD2で開発したVNFは，使用するコアの数をネットワーク負荷の状況に応じて秒単位で調節できる．単にDPDKを使っただけだとこれは不可能である．

また，高性能にパケット処理を行う部分はC/C++を用いて開発し，susanowの機能を制御する部分はPythonで開発した(ssnpy)．ssnpyをラップしたssnctlコマンドを用いると，プログラムを書かずにsusanowの制御ができる．ssnpyはsusanow上に実装されたREST APIをラップしたものなので，Rubyなどのユーザフレンドリーな言語に移植可能である．susanowはオープンソフトウェアである．

実際にsusanowでNFVの性能がどう上がったかの実験結果を図-11に示す．最上段が2つの接続のネットワークトラフィック量，中段がD2エンジンをオフにした状況，下段がD2エンジンをオンにした状況である．小さくて分かりにくいかもしれないが，中段のような固定割り当てでは青の接続も赤の接続も無駄にコアが割り当てられているところがある．しかし，下段のように動的にコアを割り当てると，平均で15%の性能向上が見られた．彼はこれからもパケットを愛し続けるようだ．

(竹迫良範PM担当)

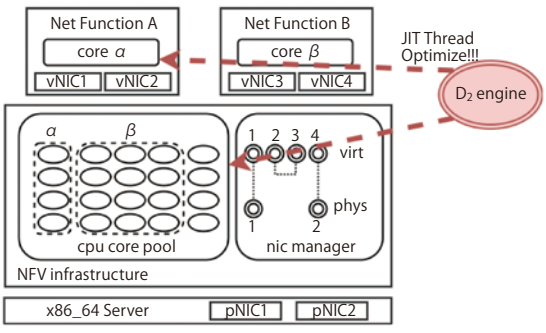
■諏訪 敬之（すわ たかし）

型による静的検証能力の高い組版システム☆6

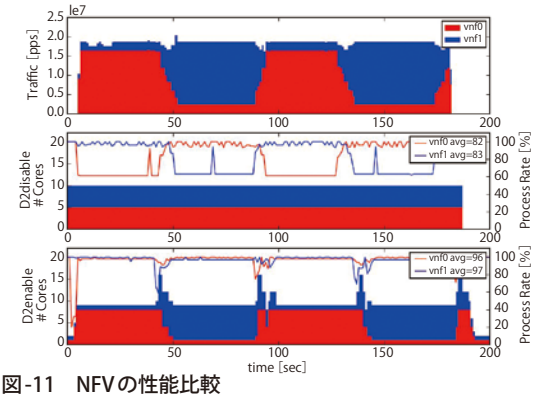
今時はMS Wordのような WYSIWYG エディタで文書を作っている人のほうが多いと思うが，技術者にはこだわりをもってTeX/LaTeXのようなマークアップ言語を使って文書を作る人が多い．私もその1人だ．

LaTeXを使っている人の最大の悩みは，コマンドを間違ったときのエラーメッセージがあまりにも分かりにくいことである．LaTeXはTeXをラップしているので，下位のTeXのエラーメッセージが生で出てくるからだ．

☆6 https://github.com/gfngfn/SATySFi



■図-10 susanowのアーキテクチャ



■図-11 NFVの性能比較

その TeX を「伏魔殿」だという人はかなり多い.

諏訪君は Yet Another LaTeX といえる SATySFi を開発した. その特徴は, OCaml という関数型言語に近い形でコマンドが定義できることと, それによって可能になった強力な型検査によるエラーチェックで, エラーメッセージの品質が格段に上がったことである.

たとえば,「13 行目 26 文字目から block-text 型の引数が与えられているが, 本当は length 型の引数, つまり何らかの長さの指定が期待される」といった内容のメッセージが出てくる. たしかにこれなら分かりやすい.

また, マークアップ言語としての可読性, 書きやすさが高まったことにより, 組版のいろいろなライブラリを作りたい人にはとても嬉しいシステムとなった.

SATySFi の内部構造を図 -12 に示す. これを見て分かるように, 型検査器が重要なコンポーネントになっている. さらにページ分割処理のあと, いきなり PDF 出力になっている. dvi や ps といった余分な中間ファイルを必要とする TeX と違ってすっきりしている.

Knuth は TeX を作るときに, 印刷術に関する膨大な文献を調べた. その結果, TeX で印刷できない印刷書式はほぼないというくらいになっている. SATySFi も文字が連接する合字, 文字の間の微調整を行うカーニング, 自動ハイフネーション, 数式, 英文と和文の混植, 画像の挿入など, 必要な機能はほぼ揃っている. 組版例を図示してもいいが, ふつうにきれいな PDF なので逆にインパクトがない. なお, これだけのものがそう簡単にできるわけではなく, 諏訪君が未踏以前に作っていた Macrodown という言語の実績があったからだと思う.

これからも改良と機能増強が続くはずであるが, 諏訪君の報告書自体がすでに SATySFi で書かれているし, コミュニティ形成が進んでいて, SATySFi で出版すると申し出た人も現れた. それくらい完成度が高い. 私は個人的に, 脚註機能を早く実現してほしいと思っているが, 予想外に難しくて, 少し時間がかかるそうだ. さらに完成度が高まれば, 諏訪君が言っていた「打倒 LaTeX」もとい「better LaTeX」の地位は確実に掴めると思う.

ところで, 諏訪君は Knuth と同様フォントにもこだわりがあり, いくつか自分でフォントデザインをしている. SATySFi のロゴも然り, 諏訪スタイルともいうべき独特さがあるので, それを図 -13 に紹介しておこう.

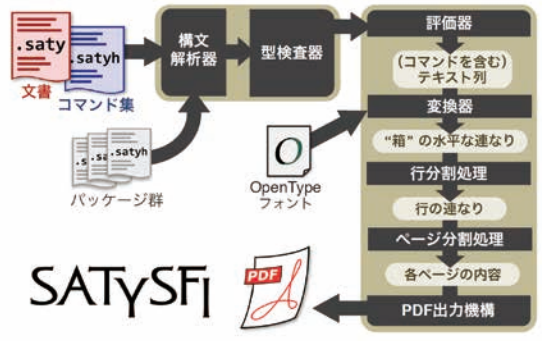
(竹迫良範 PM 担当)

■高田 峻介 (たかだ りょうすけ)

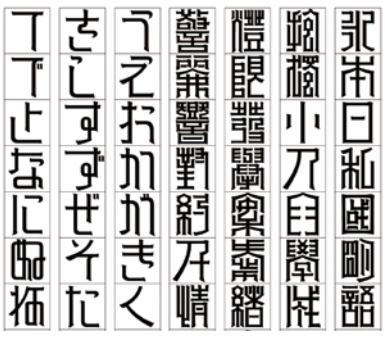
Grov：安価かつ軽量なデータグローブ☆7

ゲーム, HMD を用いた没入型 VR コンテンツ, 仮想キャラクタによる動画配信のためのモーションキャプチャなどにおいて, 手の形状をセンスする需要が高まっている. これをいろいろなセンサを仕込んだ手袋で実現するのがデータグローブである. カメラで手を撮影するより精度が高い. 1990 年にはニンテンドーのファミコンのコントローラとして 2 万円ほどで発売されている.

☆7 http://grov.fit/



■図-12 SATySFi の内部構造



■図-13 諏訪君がデザインしたフォント

つまり，歴史は十分古い．

その割に大きな技術革新がなかったと主張するのが髙田君である．なにしろ高い．高いのは1万ドル，一番安いので180ドル．その割には重くて，信頼性が低く，なによりも洗濯できない．手に汗握るゲームを続けたら臭くて使えなくなりそうだ．

髙田君は，そもそもデータグローブにセンサをつけるから重くなる，だから，センサレスの手袋にすべきと発想した．こうしてできたGrovは，当然，軽い（20gというのはデータグローブの常識を破る軽さ），柔らかい，洗濯できる．それを図-14に示す．その卓抜な秘訣は指の内側のほうに編み込まれた導電繊維を利用することである．手袋自体は普通の手袋とほとんど見分けがつかないが，根元のほうに電極がついている．これを腕に装着した計測回路につなぐ．指を曲げると，導電繊維が短絡し，抵抗値が曲げ具合に応じて下がるのである．この曲げ測定の精度が意外に高い．生のセンサ値としては400段階以上（1度以下）の解像度があるが，指1本あたり3段階の精度に抑えている．

指と指の接触の検知は，指ごとに周波数の異なる交流電圧をかけることで実現した．指と指が触れると周波数合成された交流が流れる．それを周波数解析すればどの指同士が触れたかが正確に分かる．同様に，特定の対象物を掴んだかどうかは，その対象物に固有の周波数の信号を発するマーカを貼り付ければよい．

駆動時間は約10時間，製造原価は片方30～40ドル（販売価格は100ドル程度？）なので，類似製品の戦意を一挙に喪失させるくらいのインパクトがある．本稿の最初に紹介したDragonASICを使えば，10mm四方にFFTやモジュールが入ってしまう！

Grovを使って開発をしたい人のためのSDKが公開されているし，自作者用に回路図やArduinoソースが公開されている．

ここまで出来がいいと，いろいろな応用が思いつく．遠隔地のマニピュレータの操作，手によるジェスチャ表現のキャプチャ，あるいは外国で流行っているデータグローブを使ったサバイバルゲーム（引金を引くくらいだから実は簡単そう）など．サバイバルゲームは実際にやってみたそうだ（図-15）．

髙田君はGrovの普及，ビジネスに強い意志がある．実際，複数の企業，大学，高専などからコンタクトがあり，いまも折衝を継続している．（藤井彰人PM担当）

■照屋（てるや） 大地（だいち）

Mulvery：Rubyで書けるCPU＋FPGAコデザイン環境

「コデザイン環境」と言われてもピンとこないかもしれないが，このプロジェクトの狙いは一言でいうと「Rubyのプログラムをどえらく速くする」ことである．

プロジェクトのきっかけは大量のマイクを使ったシステムを作ろうとしたとき，マイコンでは能力不足，FPGAでは開発がとても面倒．だったら，Ruby（実はmRuby）でプログラムを書いただけで，自動的にFPGAにハード化する部分と，マイコンの上にソフトウェア実装する部分に分けてくれるシステム，つまりコデザイン環境を作

■図-14　Grov，計測回路を除くとただの毛糸の手袋にしか見えない

■図-15　Grovを使ったサバイバルゲームの実験

ればいいと発想したことである．これが Mulvery である（図 -16）．Ruby でシステムを書きたい人が FPGA の細かい技術を知っていることは想定しにくいので，FPGA 合成のチューニング負荷をかけないことが重要である．

Reactive Programming というプログラミングパラダイムの具体的な実装の1つである Reactive Extensions（Rx）で記述された mRuby プログラムからは，容易にデータフローグラフが抽出できる．Mulvery はインタフェースクラスを用意することで，プログラム記述をソフト／ハード分割に適した構造に誘導する．インタフェースクラスを用いた Rx 記述とそれ以外の記述は互いに疎な結合であるため異なるメソッドに簡単に分離できる．つまり，ユーザにとって自然な書き方でソフト／ハード分割できるようしたわけである．

こうしておいた上で，Rx オペレータと等価な振舞いをする回路を（ハードウェア記述言語で書かれた）テンプレートとして用意しておき，Rx オペレータが呼び出された順に回路を接続していく．実際にはラムダ抽象周りで，いろいろな難題があったが，ある程度現実的な解法で乗り切った．

こうしてできたシステムを実際に試してみたのが，128 × 128 の画像に対するラプラシアンフィルタ適用である．実際にハードは作らず，シミュレーションをしただけであったが，Ruby でソフト実行（Core-i3@3.4GHz）したら 500ms のものが，FPGA Zynq-7000@100MHz では 0.16ms という結果が得られた．

実際にハードウェアとして作ったのは，32 × 32 の大きさのフルカラー LED マトリクスの制御システムである．24 × 1024 ビットの信号を 0.4/0.8 μs のパルスで連続出力することで，1 本の信号線ですべての LED を光らせることができた．マイコン側は Ruby の Web サーバフレームワーク Sinatra を用いた（図 -17）．当然だが，マイコンだけではその速度で信号を送ることはできない．

プロジェクト期間中，「どうして Ruby なの？ いまどきは Python のほうがユーザが付くのでは？」という質問がよく寄せられたが，照屋君の答えは「Python はメソッドチェインがしにくいし，ブロックが渡せないのがイマイチ」だった．しかし，最終段階では「開発して分かったけれど，Mulvery を移植すれば人工知能が得意な Python にも適用できるはず」と述べている．

照屋君は FPGA ベンダの Xilinx 社を訪問して，Mulvery の適用可能性について議論したりもした．そこでは多品種少量の IoT を低コストで開発したい工場の現場に使えそうという感触を得た．たしかにそれこそが Mulvery の狙いである．（首藤一幸 PM 担当）

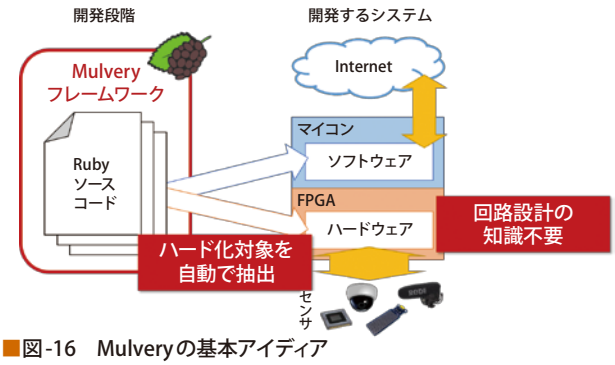
■師尾（もろお） 彬（あきら）

yabits：もう 1 つの UEFI 実装[☆8]

PC を立ち上げるとき，ほとんどのユーザは気にすることはないが，BIOS のお世話になっている．BIOS は，PC の起動時に，ハードウェアの初期化や OS へのサービス提供などを行うために最初に実行されるソフトウェアである．

一般的な x86 系の PC では，1981 年の IBM PC 由

☆8 https://github.com/yabits/uefi



■図-16 Mulveryの基本アイディア

■図-17 Mulveryで開発したWeb画像表示LEDパネル

来のレガシー BIOS が長い間使われていたが，標準化が進んでいないとか，柔軟性が足りないとかの問題があり，近年 UEFI Forum が標準化した UEFI（Unified Extensible Firmware Interface）が主流になってきている．現在は Intel 主導でオープンソース開発が進められている TianoCore と，それをベースとして各 BIOS ベンダが独自の機能を追加して売っている UEFI 実装の 2 種類がある．後者はソースが公開されていないので問題が起こったときに対処しにくいため，オープンソース実装の動きが進んでいる．

その代表格が coreboot である．coreboot はマシン依存の coreboot 本体と，さまざまなソフトウェアを載せる payload からなる．TianoCore も payload に載るソフトウェアである．

前置きが長くなったが，師尾君はなんとも裏方的なこの世界で，高速，軽量，オープンソースで，TianoCore とコードを共有しない新しい UEFI，yabits を coreboot の payload として実装した．これは，QEMU などの仮想マシンを含む，coreboot がサポートするマシンすべてで動作する．すでに GRUB2，Linux，OpenBSD の起動を確認している．図 -18 に yabits から起動された GRUB2 の画面を示す．実際，成果報告会では，yabits で立ち上げた仮想マシンの上でそのままプレゼンテーションを続行するという技を見せてくれた．

yabits は TianoCore に比べてはるかに小さいフットプリントを持ち，OS 起動のみにフォーカスしている．ドライバの読込みを必要最小限にするなどして高速化も達成している．TianoCore と比べると，フットプリントが 4.1MB から 0.4MB と約 10 分の 1，起動時間が 7.15 秒から 3.15 秒と半分以下となった．

表 -2 に yabits とほかの UEFI 実装の比較をまとめる．なお，この中の NERF は開発期間中に Google が発表したデータセンタ用のファームウェアである．

機能面ではほかの UEFI と大差はないが，軽量・高速ということで，組込みデバイスや，ベアメタルクラウドにおいてインスタンスを起動した際の起動時間短縮やコストカットが見込まれる．

師尾君はとても地味な雰囲気の人だが，デバッグ環境がほとんどない状況でやるべきことをきちんとやり遂げた．今後 x64 対応など必要な改良を行いつつ，yabits コミュニティの形成に努め，BIOS ベンダに積極的にアプローチしてデフォルトの BIOS として採用されるように活動する予定である．　（竹迫良範 PM 担当）

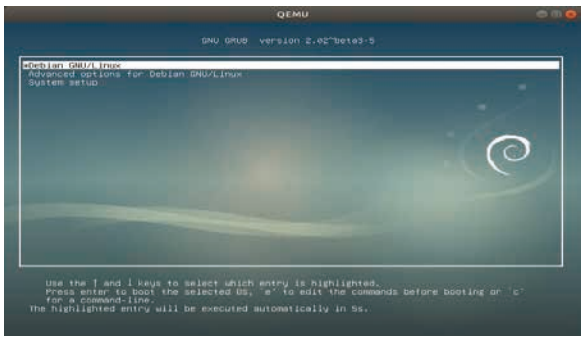
■ 横山 稔之（よこやま としゆき）

MoMIG：グラフゲノムブラウザ☆9

DNA の読み取り技術はこの 12 年間で約 100 億倍進歩した．こうして読み取ったゲノムを，種の基準配列と比較しながら見る，あるいは日本人といった集団のゲノムをまとめて見るなど，ゲノムの可視化ツールは研究者にとって重要なツールである．

ゲノム配列間の欠失・挿入・逆位（逆方向になること）・重複・転座（他の染色体に移ること）といった「構造多型」

☆9 https://github.com/MoMI-G/MoMI-G



■図 -18　yabits から起動された GRUB2

■表 -2　yabits とほかの UEFI 実装の比較

	起動速度	フットプリント	オープン性	TianoCore とのコード非共有
yabits/uefi	○	○	○	○
既存の UEFI	×	×	×	×
TinaoCore	×	×	○	×
NERF	○	○	○	×

は，遺伝子の重複・欠損・融合を引き起こし，遺伝性疾患やがんの原因となる，だから，構造多型をゲノムブラウザで見られるようにしなければならない．従来からもゲノム配列を見るためのゲノムブラウザはあったが，構造多型を可視化するには，配列間の差をグラフ構造として捉えるべきである．これがグラフゲノムブラウザである．

横山君は真に使いやすいグラフゲノムブラウザを目指して開発を進めた．その成果が Web アプリケーション MoMIG（Modular Multi-scale Integrated Genome Graph Browser，モミジと読む）である．従来からあったゲノムブラウザの技術を援用しつつ，独自のグラフィクスをいくつも用意した．「いくつも」というのは，ゲノムをどの詳細度，どの様相で見たいかは，得たい情報によって異なるからである．だからデフォルトの MoMIG の画面は巨大である．

図 -19 はトップレベルの Circos Plot の全体図である．染色体が円環上に並んでいて，染色体をまたぐ構造多型があるとその間に線が引かれる．この図の右側には詳細情報が表になって示される．従来からある手法だが，色や表示方法に新しい工夫がある．

MoMIG の売りは SequenceTubeMap というグラフ表現である．図 -20 の上段は黒で示された参照配列に対して，オレンジ色の配列ではある部分の配列が逆位していることが分かる．下段は集団のゲノムの違いを表現している．逆位などの極端な構造多型はないが，このような可視化が研究者にとって有用なことが理解できよう．

横山君は開発期間中，実に多くの研究機関等を訪問して，MoMIG の改良のための意見聴取を行った．そして数十個もあったコメントをどんどんシステムに反映していった．これは並大抵のことではない．MoMIG はオープンソフトなので，仲間が加わって，さらに改良が進むことを期待したい．

MoMIG の活用によって，構造多型に相関があることが分かっている遺伝病，肺腺がんや大腸がんなどの疾患に対して，遺伝子的な治療が行えるようになる日が来ることを期待しよう．　（首藤一幸 PM 担当）

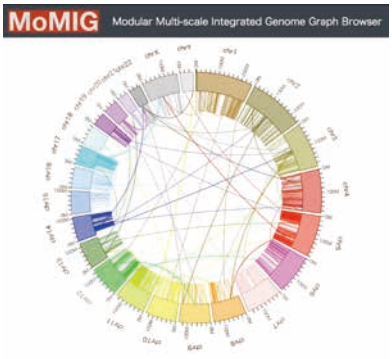
＊　　＊　　＊　　＊

2017 年度，起業・事業化に重点をおいた年齢無制限の「未踏アドバンスト事業」のトライアルが行われた．5 つのチームがプロジェクトを実行した．この事業は，人材発掘・育成を趣旨とする未踏とは異なるので，採択された人々を「イノベータ」と呼ぶ．2018 年から始まる「未踏ターゲット」と，2016 年から始まっている 17 歳以下を対象とする「未踏ジュニア」（一般社団法人未踏が担当）も含めて，「未踏ファミリー」と括られることになった．しかし，25 歳未満を対象としたこの未踏が，若さゆえの溌剌さと，アカデミアにも興味を持ってもらえる内容があって面白いと思うのは私だけではあるまい．

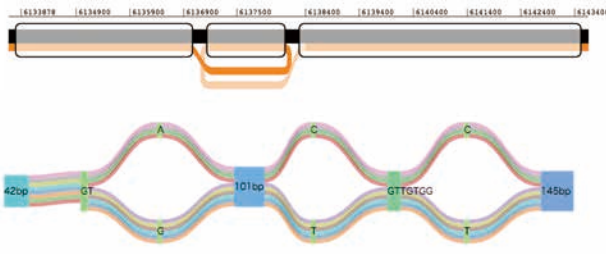
（2018 年 5 月 22 日受付）

竹内郁雄（正会員）■ nue@nue.org

1971 年東京大学大学院修了．以降，NTT 研究所，電気通信大学，東京大学，早稲田大学を経て現在，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，一般社団法人未踏代表理事．（株）Givery 技術顧問，東京大学名誉教授．本会フェロー．



■図-19　Overall View：Circos Plot



■ 図-20　Graph View：SequenceTubeMap