

未踏の第25期 スーパークリエイターたち



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々を未踏スーパークリエイターと呼ぶ。この認定は2019年で25回目となる。突出した才能を持つスーパークリエイター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25期の未踏クリエイターは計27名(21プロジェクト)で、そのうちの16名(12プロジェクト)がスーパークリエイターとして認定された。2017～2018年の認定率50%をさらに超えた59.3%となった。嬉しい。しかも、今期は2名の高校生が認定された。

今期は低レイヤが比較的少なかったが、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。

例によってクリエイターを代表者の50音順で紹介する。タイトルは正式なものではなく、「名は体を表す」キャッチに変えてある。なお、2019年2月16～17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。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、この記事は単なる導入紹介だとしてご理解いただきたい。

おがわ ひろみ やまな たくと

■ 小川 広水, 山名 琢翔

あつという間にロボットを歩かせる
MonoMotion^{☆1}

世の中にはホビーとしてロボットを作ってみたいと思

☆1 <https://monomotion.netlify.com/>

う人がたくさんいるに違いない。しかし、2つのハードルがある。1つ目は、設計して、部品を集めてちゃんと組み立てること、2つ目は組み立てたロボットを歩行させるなど、うまく制御することである。

小川君と山名君は高校2年生だったが、以前から1つ目の課題については、Webを通じていろいろなノウハウを公開していた。しかし、2つ目については、自分たちも散々苦労してきた。Web上にも有益な情報がない。それならば自分たちで分かりやすい支援システムを作って、広く情報を公開したいというのがこのプロジェクトの目的である。

実際、脚註にあるURLにアクセスしていただければ、ここでの短い紹介よりもはるかに分かりやすいシステムの構造の説明が得られるし、GitHubを見れば、すべてのソースが公開されている。

それでもあえて紹介すると、ロボットを組み立てた人は、まずロボットの物理モデル、つまりジョイント間の各部分の重量を、組み立てたまま半自動的に測定する(Reficere, 図-1)。このために普通の高校生は解かないような物理の問題を解くことになった。2分間ほど各関節が自動で動き、荷重配分の測定によって、各部品の重量が求まる。

また、3DCGソフトBlenderへのアドオンとして制作したBlendMotionを使って、歩くモーションなどをGUIでさくさくと作成する(図-2)。これと測定したロボットの物理モデルを組み合わせて、物理シミュレーションを行う。所望のモーションになるように機械学習

によってモーションを調整する (trainer). この結果を実機を動かすためのソフト flom-runner に渡すのである.

この一連の流れを経て、いまのところまだヨチヨチではあるが、組み立てたロボットが歩きだす. 未踏以前には、組み立てたロボットを倒れないで歩かせるために彼らは9カ月を要したという. しかし, MonoMotion を使うことによって、後輩の中学1年生の未経験の子が、使い方の習得に10分、所望のモーションを作るのに30分しかかからなかった! また, trainer による学習により、歩行継続時間が20秒から3分以上に伸びた.

実はここまでできるまですべてが順調だったというわけではない. 当初、深層学習を使って歩行を実現しようとして数々の奇妙な「歩行?」が出てきて、見ていた我々に「大丈夫か、これで?」と思わしめた. しかし深層学習を諦めて、遺伝的学習のような方法に切り替えるピボットを行った. 新規のヒューマノイドロボットや腰から下だけの多関節ロボットを4台作って、特定の機種に限定されない工夫もした. 多数のプログラミング言語やツールを実によく使いこなした.

2人は毎日授業のある高校生なのに、よくここまでやれたと思う. しかも、夏休みには Maker Faire Tokyo 2018, 9月には World Maker Faire New York 2018 に出展までしている. 小川君が主にソフト, 山名君が主にハードという分担だったが、全体として見事な合作だった.

MonoMotion をまとめると、サーボモーターで構成されたほとんどすべてのロボットに対して、何も知識がないユーザでも動作を作成して、実機で動かせるように

するオープンソースのソフトウェア群、ということなのだが、その背後にあった彼らの哲学「創作欲のままに創れる世界へ」は多くの人々の心に響くものがある.

(首藤一幸 PM (プロジェクトマネージャ) 担当)

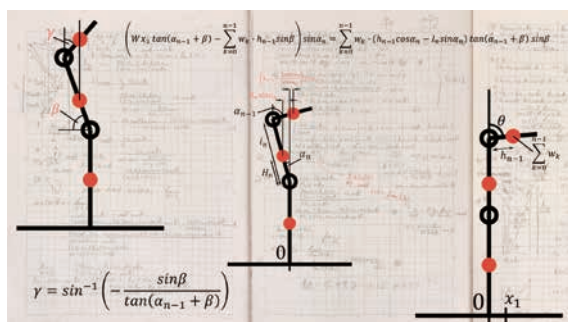
■ ^{かたくら しょうへい} 片倉 翔平

動きと機能の再生装置としての3Dプリンタ Functgraph ^{☆2}

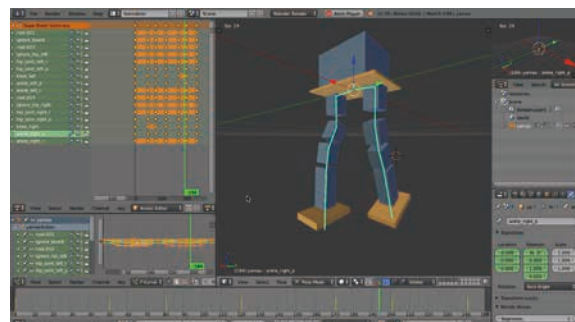
3Dプリンタの普及が目覚ましく、一挙にパーソナリ化が進んでいる. 多くの3DプリンタはFDM(熱融解積層)方式で、エクストルーダからプラスチックのフィラメントを融かしながら、 x, y, z 座標で指定した位置に、いわば直交ロボットから吐き出して台の上にプラスチックを積み上げていく. そんなことは先刻承知なのだが、片倉君は3Dプリンタを単なる印刷以外にも使えるのではないかと考えた. エクストルーダをうまく制御すれば、人形を動かすことはできる. しかし、それではつまらない、3Dプリンタで印刷したものがそのまま動くところまで、人間の介入を経ずにできなくては! とんでもないことを考えたものだ.

開発初期のFunctgraphのコンセプトデモの1つが、コーヒーミルを印刷して、そのままエクストルーダがハンドルを回して、コーヒー豆を挽くというもの(ただし、人がコーヒー豆を引出しに入れる). もちろん金属の歯がないので、まともに豆が挽けるはずはないが、なんだか心なむ作品だ.

^{☆2} <http://katakurashohei.me>



■ 図-1 分解しないで部品の重量を測る



■ 図-2 歩くモーションの作成

成果報告会で彼が見せた最大の作品はベベルギヤを持つカラクリ人形仕掛けである。写真1はプリント直後の全景である。新しい手法の数々が総合的に織り込まれている。

写真1にはいろいろなものが写っているが、右側のフォークのようなものは仕掛けを組み立てるための治具(エンドエフェクタ)である。つまり、エンドエフェクタも自分で印刷し、それをエクストルーダが掴みあげて(台から引き剥がして)使うのである。斜めになっている棒は中空印刷のための補助つかい棒で、終わったらエンドエフェクタでポキッと折る(折る場所と折る力を最適化するのに後述のCADを利用する)。また、使い終わったエンドエフェクタを外す左端の爪の仕掛けも自分で印刷した。

奥にある2枚の歯車は、そのうちの1枚をエンドエフェクタで持ち上げ、噛み合わせてベベルギヤにする! 手前のボックスの中の円は向こう端が四角になっている円柱で、立ち上がった歯車の中心に刺さる。ボックスの中では円柱がカムの形になっていて回転運動を上下運動に変える。こうしてボックスの上に乗せた人形が楽しそうに動く。しかし、このような言葉を並べるより、成果報告会の動画のほうが分かりやすい。

水平方向の歯車を回す? 台の上に印刷したものをそんな簡単に剥がせたり、動かしたりできるの? これは、まずフィラメントを細くぐちゃぐちゃと1.5~2.3mm厚に垂らして、緩い緩衝材のように敷き詰めてから中空プリントするという技術を使う。こうすると歯車と垂

れたフィラメントが外れて回転できる。

これも含めてたくさんの面白い技法を発明したのだが、すべては紹介しきれない。写真2のようなリンク機構で実現したマジックハンドは「橋渡し」という技法で可動リンクを作る。この腕は、写真の左にある空き缶を掴んで、ポイッと投げ捨てる愛敬がある。

このほかにも面白い作品をいくつか作ってくれた。ししおどしのような形で植物に水をやる装置や、扇いでくれる団扇など。もっともこの団扇、ゆっくりすぎて全然涼しくなさそう。もっばら見て愛敬の作品だろう。

このように動きと機能を持つものをデザインするソフトウェアをGrasshopperのコンポーネントとして作成した。これもとてもよくできている。

それにしても、機構学の勉強から始めたのだから伸びもすごい。片倉君はこの成果をベースに、ハプティクス技術やパーソナルファブリケーションを専門とする、ポツダム大学のPatrick Baudisch教授のところに学位取得のため数年間留学することになった。

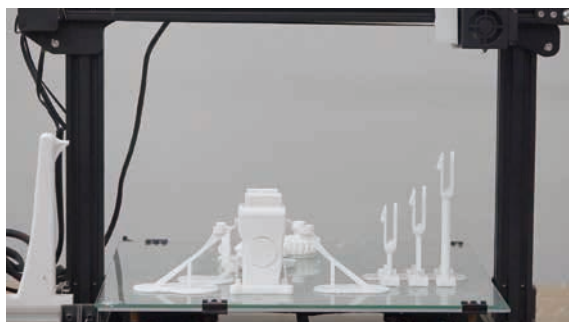
(稲見昌彦 PM 担当)

■ 桂 大地

ボルダリングコース作成支援アプリ RedPoint^{☆3}

東京オリンピックの正式種目にもなったボルダリングがいま熱い。日本の競技人口はすでに柔道の競技人口に並んだらしい。国内のボルダリングジムは10年間

☆3 <https://red-point.jp>



■写真1 自己作成カラクリ人形仕掛けのプリント



■写真2
空き缶を掴む
マジックハンド

で5倍に増えた。

室内におけるボルダリングとは、ロープをつけずに登れる程度の高さの壁に、掴んだり足を引っかけることのできる突起（ホールドという）がたくさん並んでいて、それを利用してきっちり上まで登る競技である。ただし、使うべきホールドの選択肢（コース）が指定されているところがミソである。だから、単に登るよりも競技性が高い。

ボルダリングのコースはジムが定期的に変更して、飽きさせない努力をしなければならないが、かなり手間がかかるし、設計も難しい。このほかにも、同じ壁に初心者向けとか上級者向けのコースをたくさん並置したいが、分かりやすくそれを表示するのが難しい。そもそも、コースの難易度がいまだに曖昧で、各国のいろんな基準が乱立している。それに、身長によってコースの有利不利があるという問題もある。身長が高いと有利に思えるかもしれないが、高身長だと体を大きく折り曲げないといけない、つまり落ちやすいコースもある。

桂君は、先生に頼んで研究室内にボルダリングの壁を作らせてもらったほどのボルダリング好きである。彼は上記の4つの問題の解決を支援する、コース管理者およびコースに登るユーザ向けのタブレットアプリ RedPoint を開発した（バックエンドサービスとして Firebase を使用）。対象とするのは初心者および中級者向けのコースである。

使い方の主な流れは次の通り。ジム管理者がまず壁を撮影し、ホールドの登録、床面の設定、画面と実世界のサイズの対応付けを行う。アプリはホールドの難易度を算出して表示する。管理者は必要な修正を行う。

これでコース作成支援の準備が完了する。

管理者またはユーザは、ホールドの難易度情報と、身長による有利不利を解消するための身長入力を用い、コースの設計を行う。これは小学生でもできるくらい簡単で、コースの難易度を設定し、スタートのホールドから順次、難易度に応じた推選ホールドを選んでいくだけである。なお、ちょっと難しめとかちょっと易しめとかは選択肢の色で分かるようになっている。作ったコースはデータベースに保存できるので、ほかの人が再利用できる。壁にはコースが表示されないので、大丈夫かな?と思ったが、慣れると子供でも間違えないそうだ（写真3）。

登っている動画の共有サービスもある。要するに、「あいつ、あんなコースを登ったんだ、すごい!」のようなボルダリング仲間での情報共有がサポートされている。

しかし、なんと言っても特筆したいのは、曖昧な難易度基準へ挑戦である。身長を考慮に入れている時点で、世界のさまざまな基準を超えていると思うが、このように難易度によってコース作成支援をする具体的なシステムを作ったというだけでも、それ自体が工学的基準になり得る。桂君は、PM たちの推しもあり、これを Katsura 指標 (K-index) として売り出していくはずである。

桂君は北海道の学生なので、このシステムを北海道の多くのジムに試験導入してもらっている（写真4）。これからもこの実証実験を進めていくので、この調子で RedPoint が全国、さらには世界に展開していくことを期待したい。（藤井彰人 PM 担当）



■写真3 自分で作ったコースに登る子供



■写真4 イベントの宣伝

きよかわ たくや ともちか けいた
■ 清川 拓哉, 友近 圭汰

認識 AI を迅速に賢くするフレームワーク^{☆4}

近年、ある大手物流センターでは毎週 10 万の商品が入れ替わるという。その上、コンビニ等での人手不足問題もあり、多様な商品が溢れる現場での商品管理の自動化が喫緊の課題となっている。課題は 2 つに分かれる。1 つ目は「カメラによる商品の高精度な認識」であり、2 つ目は「商品の迅速な登録」である。前者はすでに深層学習によって見事に解決されている。しかし、2 つ目は学習データの準備に驚くべき人手がかかっている。両君が過去に経験した事例では、6 つの商品のさまざまな角度での画像を収集して、商品情報のデータを付加するアノテーションに 15 時間以上かかっていた。これでは流通業界は動かない。

アノテーションを行うために、商品をいろんな角度、いろんな照明条件で、人手を最小限にして撮影しなければならない。このためには、多視点画像撮影を行い、あらかじめ入力された商品の大きな形状(缶、箱、袋の大きな外形)に従い、商品の、いわばアイデンティティを学習データとして収集する。

多視点画像を撮るカメラは 1 台のみで、それを 6 自由度のロボットアームに取り付ける。そして商品を自動回転ステージに乗せる(写真 5)。回転ステージには 3 つの AR マーカがあり、画像中で検出できれば、その ID と位置姿勢を知ることができる。事前に AR マーカと商品に対応づけしておく、形状の概略も与えてお

^{☆4} <https://takuya-ki.github.io/>



■写真 5 多視点画像の撮影

くので、商品の位置姿勢と輪郭が推定でき、最後に AR マーカの ID と紐づけた商品ラベルを画像中のボックスに対して付与することで、学習データセットを生成していく。なお、この際、撮影した画像を取捨選択し、学習に有用なデータのみ蓄積していくところもミソである。このあたりは特許出願準備中である。

こう書くと簡単そうだが、商品を囲むボックスの中に AR マーカが写り込んでしまうと、認識 AI の学習を混乱させてしまう問題があった。AR マーカを商品の一部の情報としてしまうのである。これを回避するために、写り込んだマーカをノイズ画像で上書きするという工夫をしている。たとえば、図-3(a)に写り込んだマーカの領域に対して、ノイズ画像を付加する(図-3(b))。

これらの方法により、15 時間以上かかっていた学習データの準備を 4 分で済ませるようになった!

ここで開発した技術は、従来使われていた技術に対して、人手を介す部分が大幅に減少しているところが優れている。つまり、現実の画像に対して完全に自動化されたアノテーションが行え、かつ多視点の画像の効率的な自動収集が可能になった。

清川君たちは、開発を進めつつ、このシステムに需要があるかを調べるために、巨大な物流センターを持ち、ロボットの活用を進めているアスクル(株)などでヒアリングを行った。その結果、(1) 商品改廃が多くある現場では、学習データセット生成を迅速に行う技術や認識技術の進化に需要はある、(2) 季節限定の商品等のマルチパッケージの商品を同一の商品として認識するところに大きな課題がある、(3) 99.999% の認識率達成+迅速



■図-3 認識 AI の誤作動を防ぐマーカ除去

な商品登録の両立が課題になっていることが分かった。

このプロジェクトは特許出願の数がほかのプロジェクトに比べて突出していて、競争優位に立っていることは間違いないのだが、彼らには商品形状の概略の入力にまだ人手がかかっていることを気にしている。そのため、別の技術の利用を考慮中である。

このシステムが無人コンビニの本部などに導入され、単なる省人化ではなく、本当の無人化に寄与する日がやってくるのが楽しみである。（首藤一幸 PM 担当）

くろだ かずや
■ 黒田 和矢

ゲーム実況 AI：宮咲ふわら^{☆5}

近年ゲーム実況の市場が熱い。ゲーム実況のスタイルには、人間が実況を行うスタイルに加え、合成音声を用いたスタイルや、バーチャル YouTuber と呼ばれる 3DCG を実況キャラクタとするスタイルがある。しかし、実況を行うのは常に人間である。

黒田君はこれを AI にやらせようとした。対象にしたゲームは任天堂のマリオカート 8 デラックス（以下マリオカート）である。人間のプレイの映像から実況発話、つまり言語生成に至るまで、「見る」、「感じる」、「話す」の 3 つの段階を経る。

(1) 画像認識／動作認識／物体検出を用いた画面認識

マリオカートは一人称（よりちょっと後ろの）視点で画面が動く。その画面にはキャラクタの状態、路上のいろ

^{☆5} YouTube: <https://www.youtube.com/channel/UC3ggg0ryKhUMSFaiCxAwbw>,
ニコニコ動画: <https://www.nicovideo.jp/user/87155387/top>



■ 図-4 学習支援をする画面

いろなアイテムや敵キャラ、ラップ数、順位、全コースの俯瞰マップなどが表示されている。これらを CNN (Convolutional Neural Network) で学習して画像認識させる。

図-4 は画像認識用学習データ作成支援システムの実行画面である。動作は、フレームの連続画像からのキャラクタの動き検出が必要なので、LRCN (Long-term Recurrent Convolutional Network) を使う。コース上の物体検出は SSD (Single Shot Multibox Detector) を使う。

(2) 画面認識と生体情報を用いた感情表現

生体情報はプレイしている人間の心拍数である。これで快・不快、眠気・覚醒を検出し、黒田君が自分で描いた宮咲ふわらの目と口元で表現する。

(3) 画面認識と感情表現を用いた実況内容の生成

ここまで得られた結果に合わせて実況を行う。実況には、実況動画投稿サイトから収集したリアクションのデータと、それをもとに RNN (Recurrent Neural Network) を用いて学習して生成した（分析した実況動画中に存在しない）新しいリアクションを使用した。新しいリアクションは、使えるタイミングも自動分類した。

人間の名物実況では、プレイに直接関係しない雑談が多く含まれており、それが独特の面白みを出している。黒田君はかなり頑張って雑談生成を試みたが、さすがにこれは奥深く、デモシステムに入れるには至らなかった。

図-5 は、実況中継の画面である。右下の女性が宮咲ふわら。語尾が「ふわ」になっている。もちろん、音声つきである。髪飾りがパーセプトロンをモチーフにしたデザインというのが心憎い。脚註につけた YouTube あるいはニコニコ動画をぜひ見ていただきたい。



■ 図-5 AI実況の画面

このプロジェクトは最初、そんなこと可能なの?と心配したのだが、いろいろな学習モデルをしっかり使いこなしながら、きちんとした統合システムに仕上げた。これはマリオカートを選んだ慧眼のせいもあろう。たしかにこのゲームは学習モデルに乗りやすそうだ。しかし、黒田君はゲームの種類を増やすこと、途中までやった雑談の導入、さらには第三人称視点での、たとえばeスポーツの実況にも手を伸ばそうとしている。これだけのことを短期間に成し遂げたのだからきっとやってくれるだろう。

(五十嵐悠紀 PM 担当)

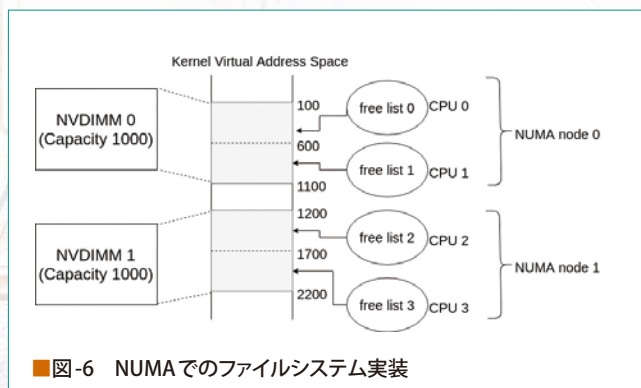
しげみつ ふみや
重光 史也

NVDIMM 向けファイルシステム AEON^{☆6}

最近の PC の外部記憶の主要部は HDD から SSD に変わりつつあるが、近い将来それが NVDIMM (Non-Volatile Dual In-line Memory Module) に変わっていくかもしれない。DIMM というように、CPU 基板に直付けになるが、DRAM のようにランダムアクセス可能で、速度も DRAM 並み、つまり SSD よりも 2 桁以上高速で、電源が落ちて記憶が失われないという特徴を持つ。

重光君は NVDIMM の最も良い導入法はこの上にファイルシステムを載せることだと考えた。ユーザ空間のアプリケーションの変更なしに、NVDIMM の性能を享受できるからである。速度性能はもちろん、まだ容量の少ない NVDIMM なのでファイルシステムのコンパクトさにこだわる。

^{☆6} <https://github.com/4ge32/aeon>



■図-6 NUMAでのファイルシステム実装

まず、高速なアドレス指定で使えるようにするために、NVDIMM 全体をカーネルアドレス空間にマッピングする。図-6は Non Uniform Memory Access (NUMA) アーキテクチャでの実装例である。実行中の CPU に近い NVDIMM を管理するフリーリストを選ぶことにより、より速くアクセスできる NVDIMM に書き込みを行う。メタデータ (inode) はフリーリストから取得したブロックによって生成されたキャッシュを使って割り当て、高速化のいろいろな工夫をした。

ファイルデータの読み書きは Direct Access for Files (DAX) を使って行う。DAX は、NVDIMM に対して最適な読み書きをするための技術である。ファイルの読み書き時に発行される read/write システムコールでは、カーネルのページキャッシュをバイパスして NVDIMM へ直接書き込む。こうすると書き込んだデータの永続性が高まる。

現代の CPU は CPU キャッシュメモリを使うことが必然なので、そこから直接 NVDIMM に書き込むことも必要である。しかし、システムダウンによるファイルシステムの破損は避けないとはいえない。AEON は Consistency Without Ordering (CWO) と呼ばれる、メタデータを保護する機構を NVDIMM 向けに再設計したものを備えている。ある種の二重化を行ってファイルシステム操作の原子性を確保するのである。いろいろなテストで、整合性破綻の検知・回復ができていることを確認済みである。このほかにも処理内容に応じたカーネルロック API の使い分け、他のカーネルプロセスから NVDIMM への誤った書き込みを防ぐ領域保護などの細かい気配りをしている。

NVDIMM の容量はまだ小さいので、AEON はバイトアクセス可能な NVDIMM の特徴を活かし、オーバヘッドの少ない透過圧縮を実現している。実際、透過圧縮をしても読み書きは 30% ほどしか遅くならない。

AEON と同じようなフルスクラッチで作られた NVDIMM ファイルシステムに UC San Diego の NOVA がある。まるで英会話教室の対決みたいだが、AEON には「永劫」という嬉しい意味がある。重光君が行ったいろいろなベンチマークでは AEON が

NOVA にほとんど勝っている。その中で最も顕著なのがメモリ使用効率のベンチマークである。図-7 は 16KB のファイルを 10,000 個生成した直後の NVDIMM 使用量の比較である。このような大差がついたのは、NOVA が正攻法でデータ保護を行っているためだと思われる。AEON のほうが現実解なのだろう。

AEON は現在でも、GitHub 上で改良が進んでいる。NVDIMM が本格普及したときに、はい、もうファイルシステムはありますよというのが重光君の狙いである。

(田中邦裕 PM 担当)

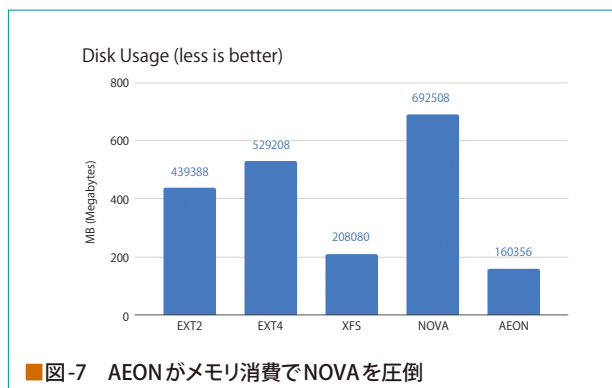
すとう かい たにみち こたろう
須藤 海, 谷道 鼓太郎

折紙プロダクト設計の統合環境 Crane^{☆7}

折紙は日本発の国際文化と言えるものであり、芸術としてだけではなく、建築、医療、産業技術、ロボティクス、宇宙工学などへの応用が始まっている。2000 年頃から現在にかけて、折紙関連の論文はなんと 10 倍近く増えている。

しかし、折紙技術はまだ身近にはなっていない。実社会からのさまざまな要求を満足する折紙のための製造技術はまだ確立していない。そもそも、折紙技術を用いた製品の図面を作る製造用ソフト (CAM) すらもほとんど存在しないのである。着ている背広も折紙風ブリーツという筋金入りの折紙マニアの須藤君と、コンピューショナルデザインが専門の谷道君のペアは、CAM

☆7 <https://www.facebook.com/Crane-design-tool-for-origami-products--272530303643926/>



を含む総合的な折紙プロダクト設計環境 Crane の開発に挑戦した。彼らの指導者であり、未踏での折紙技術の先輩である舘知宏氏 (2006 年のスーパークリエイター) が設計 (CAD) とシミュレーション (CAE) についてはすでに高いレベルのソフトを作成している。Crane は欠けていた最後のピースを補う統合環境である。

折紙の 3 つの本質とは、(1) 平面から立体に立ち上がること、(2) 折ると堅くなるようにできること、(3) 1 カ所を動かすと全体が連動するように自由度の制御ができること (この意味は後で紹介する彼らの作品の動画を見ると分かる) である。

Crane を使って、彼らは折紙で実際に座れる椅子 2 脚とテーブルを作った。どちらも厚い平らな木板を成果報告会のステージであつという間に折って即興の発表セットにしたのである (写真 6)。これは拍手喝采を受けた。

椅子のメイキングを見てみよう。まず、紙を折って椅子を気の向くまま何種類か作ってみる。折り上がったものの剛性は、紙でもある程度試すことができる。その中で丈夫そうなものを選び、折り目、つまり山折りは赤線、谷折りは青線、折紙の境界は緑線でマークしていく。これをスキャナで読み込んで、CAD データに変換する。ここから先の CAM は Crane 上での作業になる。

データを微調整しつつ、厚みがあってもちゃんと折れるかどうかなどのシミュレーションなどを行う (図-8 はその一部を示している)。こうして出力された図面をデジタル加工機に読み込ませて、木版をカットすれば折紙の各面ができあがる。これら切り出した木版をヒンジでつないで完成。あとは折れば椅子の仕上り



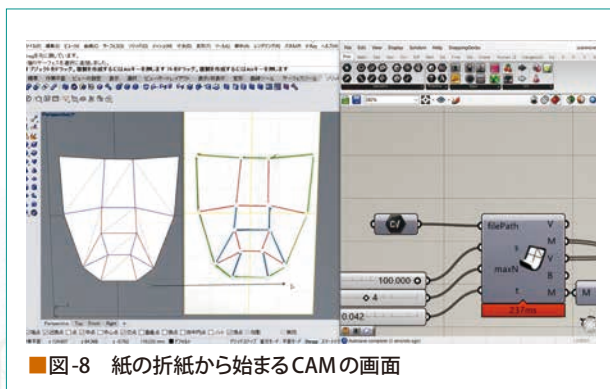
である。これらの流れは、成果報告会の動画を見ていただくほうがよい。この動画を見ると折紙の本質の3つ目(自由度の制御)の意味がよくお分かりになるだろう。1枚の板から立体のちゃんと座れる椅子を折り上げるのに10秒程度しかかからない。

このほかにも折紙で作ったスマホスタンド、デスクランプなどを見せてくれた。

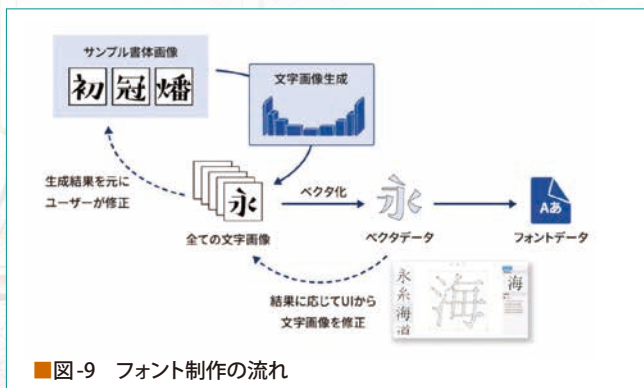
Craneは図-8を見てお分かりのように、Rhinoceros + Grasshopperのプラグインとして作られており、近々Food4RhinoというWebページで無償公開する予定である。これで、折紙の専門知識を持たない人々が、折紙パターンから厚みのある素材でのファブリケーションを行えるようになる。これで折紙に関心のあるデザイナーが面白い作品をどんどん作るようになるだろう。

企業との連携による量産品の生産も目指すそうだが、前述の館氏、計算的弾性体の大嶋泰介氏(2012年の未踏スーパークリエイター)と彼らは深い連携を保っている。そこから生じるシナジーも楽しみである。

(藤井彰人 PM 担当)



■図-8 紙の折紙から始まるCAMの画面



■図-9 フォント制作の流れ

たまだ あきひろ
■玉田 晃寛

Web フォント制作支援システム DeepGlyph ☆8

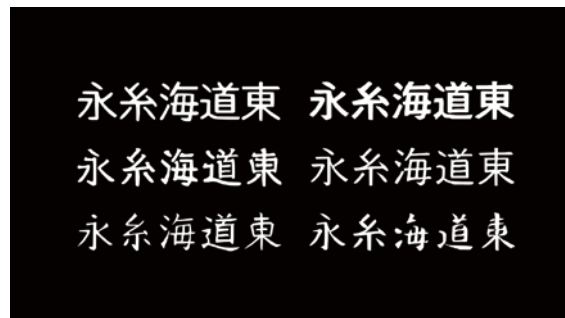
Webブラウザがインターネットを通してフォントデータをダウンロードするWebフォントの普及が進んでいる。デスクトップパブリッシング(DTP)に馴染みのないユーザも自己表現の方法としてフォントにかかわる機会が増えつつある。しかし、日本語のように文字数が多い言語ではフォントの開発は百人年オーダーのとんでもない時間がかかる。

玉田君はこれを解決するために、深層ニューラルネットワーク(DNN)を活用した画像スタイル変換による文字生成から、Web上でのフォント配信までを一貫して実現するDeepGlyphを開発した。図-9はフォント制作のざっとした流れを示す。

ユーザは左上にあるようにサンプルとなる書体を(ラスタ形式で)漢字、ひらがな、カタカナを問わずわずか10文字程度書く。そんなに丁寧に書かなくてよいところがミソである。するとDeepGlyphはほとんど待つことなく数千の文字を自動生成し、ベクタ化する。ユーザは気になる場所があったらラスタ形式でも、ベクタ形式でもGUIを使って簡単に修正できる。これだけのことでOpenType形式のフォントデータが完成する。この様子が一番よく分かるのが、この記事の冒頭で紹介したYouTubeの中の玉田君の講演の中の少し長いデモムービーである。百聞は一見にしかずなのでぜひご覧いただきたい。

■図-10は実際にユーザが生成したフォントのサンプル

☆8 <https://deepglyph.app/>



■図-10 ユーザが作成したフォント

である。これらが本当に手軽に作成できるのは驚きである。

仕組みをもう少し詳しく見よう。DeepGlyph は、書体生成ネットワークとして 2018 年に Y. Zhang らが提案した Encoder-Mixer-Decoder (EMD) ネットワークを用いる。文字を 1 枚の画像と見なして、ある 1 つの文字について異なる書体の文字画像のセット (Content reference) と、ある 1 つの書体について異なる文字の文字画像のセット (Style reference) を与えることで未知の文字画像を即座に生成できる。このネットワークをベースに DeepGlyph が実装した文字生成モデルは、GlyphWiki という Web ページにある文字データベースを Content reference とすることで、従来は実現できなかったほど多くの種類の漢字を Style reference へ数文字入力するだけで生成できるようになった。

文字を画像として変換する処理は軽くないので、一般的には Web ブラウザ上ではなく、インターネット上の高性能サーバで処理するのが普通だが、DeepGlyph はあらかじめ学習した生成モデルでの推論を WebD-NN で行う。これにより、Web ブラウザ単体でベクタ化したフォントにメタデータを付与するところまでが可能になった。OpenType フォントの作成だけはサーバサイドで行う。

上にも紹介したように DeepGlyph はラスタ形式の入力なので初心者でも簡単に使える。フォント作成の敷居を一挙に下げたと言っていいだろう。このシステムは脚註の Web ページから一般公開されるので(5月にベータ版を公開)、たくさんの Web フォントが作成され、それを利用するだけのユーザも含めて、大きなコミュニティが成長しそうだ。事前通知に対して「デザイナーをやっていますが、フォント制作はどれだけ手間がかかるんだとビビっておりました。こちらのサービス、非常にワクワクします」など多数のメッセージが寄せられている。玉田君は、プロ向けへの改良も考えているようだ。

(竹迫良範 PM 担当)

たまつ そうたろう おおつぼ しんべい
■ 玉津 宗太郎, 大坪 新平

タブ自動分類ブラウザ Leita^{☆9}

モバイルブラウザの使い方が変化している。検索したらそのままタブを残すのである。玉津君たちの調査によるといまどきの大学生は平均して 100 件以上のタブをモバイルブラウザに残しているという(私には信じられないが)。後で必要になるからという理由なのだが、ブックマークにしないのは、そうするのが単に面倒だからである。しかし、漫然と溜められたタグを見つけ出すのもまた面倒だ。

そこで「ブラウザを(ブックマークを使わずに)自動的に整理されるメモにする」という発想が生まれる。2人は未踏以前にもこの発想で Que というシステムを作成し、多くのユーザを集めた。未踏では、これを完全にリメイクした。

基本アイデアは、閲覧した Web ページをコンテンツから、映画、グルメ、本などいろいろなカテゴリに分類することである。このために機械学習した分類器を使う。ただし、ユーザごとに詳細なカテゴリや大雑把なカテゴリの基準が異なるので、カテゴリとタブの間をタグの多対多の関係で結びつける。こうして、どのような粒度のカテゴリでも容易に取り出せるようにした。

自動分類は BERT (Bidirectional Encoder Representations from Transformers) による分類モデル、単語ベクトルと文書ベクトルのコサイン類似度、キーワード抽出という 3 つの方法を組み合わせることにより、話題空間の大きさの異なるさまざまな単語に対する推論を可能にした。所要時間は 1 件あたり 200 ミリ秒以内なので、ユーザが開いたタブはすべて意識されることなくタグが付与される。

こうして、会話中に気になった言葉を聞いたとき、その場で検索してタブを残しておくことが可能になる。ブラウザのタブには情報が生のまま残っているので、再検索する必要もなく、関連する情報もシームレスに保存できる。

Leita は Search for Later とアイスランド語の search

^{☆9} <https://leita.app/>

を意味する単語にかけている。ロゴは食糧を蓄えるリスである。図-11はLeitaでたくさん保存されたタブの分類タグが表示されている画面の下半分を示している。右に入れたのがリスのロゴである。

実際に使ってみるとタブにはカテゴリ以外にも時間軸が必要なことが分かった。つまり、映画のタブが(捨てないので)そのうち数十から数百になることがある。そのため、過去(消費し終わったコンテンツ)、現在(いつでも消費できる未消化のコンテンツ)、未来(消費を希望するが現在アクションを取ることでできないコンテンツ)とさらに分けた。未来について、見たい日時に通知してくれる機能もつけた。図-12の上段の目覚ましなどが表示されているところがその指定ボタンである。

LeitaはiOSのアプリとして開発したが、Chrome Extensionも作成した。さらにWebアプリで、それらで保存したWebページをどの端末でも閲覧できるようにした。同期はしっかり行われる。検索したタブが少ない時点ではその使いよさが理解されにくいという問題はあがるが、まずは多くのユーザを惹き付ける土台が完成したと思う。

2人とも会社員としての仕事の合間にLeitaを開発した。Queの経験があったとはいえ、新しい技法をどんどん取り込んだことは素晴らしい。

(五十嵐悠紀 PM 担当)

まつい けん
■ 松井 健

C++ ユーザのためのパッケージマネージャ poac^{☆10}

松井君は自分を「マツケン」と呼ぶ熱烈なC++ loverである。当然C++の仲間をもっと増やしたい。C++で広く使われているConanというパッケージマネージャがあまりにも使いにくく、入門者の障壁になっているので、未踏でそれを打破しようとした。

C++に限らず、多くのプログラミング言語には汎用性の高い関数などをまとめたライブラリが存在し、言語のインストール時に付随する標準ライブラリと、外部からダウンロードする必要のある外部ライブラリの2種類に分類されることが多い。外部ライブラリを扱うには、ダウンロードするだけでなく、依存関係解決などの作業も必要になる。このような作業を担ってくれるのがパッケージマネージャである。

面接で紹介してくれたC++パッケージマネージャConanの使いにくさの説明は面白くて迫力があつた。松井君いわく「これではC++にはパッケージマネージャが存在しないに等しい!」

最初、システム名をPlus of Cのpocにしようとしたが、商標が大量に取られていたので、発音がそれほど変化しないようにaを入れてpoac(ポック)とした。システム名をそのままコンソールのコマンドとしたいので、打ちやすく片手だけに負担がかからないことにもこだわった。確かにpoacと打つとQwertyキーボード

☆10 <https://poac.pm>, <https://github.com/poacpm>



■図-11
見たいタブに行く
ためのタグ表示

■図-12 タブの時間分類

では流れるように打てる。さらに p の発音は日本語のオノマトペでは最も明るい印象なのだろう。本質的でない話にそれだが、彼はロゴデザインにもこだわった(図-13の右下に挿入)。

Conan と poac で同じ hello_world パッケージをインストールして走らせた比較を図-13に示す。短い右のほうが poac である。小さいので見にくいだが、poac はビルドシステムを内包しており、インストールからビルドまで管理することが可能になっている。つまり、最新の標準規格に準拠した C++ テンプレートメタプログラミングの技巧を駆使し、とてもモダンな C++ パッケージマネージャを実装することができた。スケルトン作成、パッケージ定義、依存関係の定義や図示、ビルド、実行、テスト、パッケージの登録など、パッケージマネージャの枠を超えた多くのコマンドを実装している。

詳細は説明できないが、poac のクライアントサイドのアーキテクチャを図-14に示す。クライアントサイドは C++ で書かれており、推論機構とコマンド、コマンドが使用するための各種コンポーネントによって実現される。大まかな流れとして、poac バイナリに引数を与えると推論機構でどのコマンドを実行するのか推論し、コマンド部分でコンポーネントを活用しながら各コマンドの機能を構成する。

Web アプリケーションの構築にもフロントエンドに Elm を利用し、AWS S3, Dynamo DB, Lambda, Elasticsearch, GCP k8s Engine をインテグレーションしてサービスインし、フルスタックの開発を1人で行った。poac の Web ページを見るとグラフィカルなデザインセン

スもいい。poac 公開後の反響もよく、挫折しないで済む C++ 入門者が増えることが期待できる。

(竹迫良範 PM 担当)

■ むらまつ なおや
村松 直哉

故障に強いロボット制御

路面状況が複雑な環境で移動できるのは、脚を持つロボットである。しかし、脚は可動部が多く、かつ車輪のように一方向に継続的に動かないので、アクチュエータに非常に負担がかかる。だから壊れやすい。

たとえば、6脚ロボットの1脚が故障してもちゃんと歩行し続けることができるようにするにはどうすればいいか？異なる形状のロボットを、基本が同じ制御プログラムで操作できればいいという発想が生まれる。こうすると2つの利点がある。1つ目はロボットの開発が簡単になること。統一されたプログラムにより、ロボットのマイナーチェンジや似た構成のロボットの開発が行いやすくなる。2つ目はロボットの故障に対して対応しやすくなることである。6脚歩行ロボットの脚が1本壊れても5脚ロボットを制御していると見ればよい。故障したその場で、制御対象のロボットが変わったものとし、再調節を自動的に行うことで、行動可能な状態にできる。これが村松君のアイデアであった。

開発は2つのステップに分かれる。まず、多脚ロボットをスムーズに前進させる学習、次に1~4本の脚を故障させて(脚を折り曲げて接地できなくする)、それを補償する学習である。

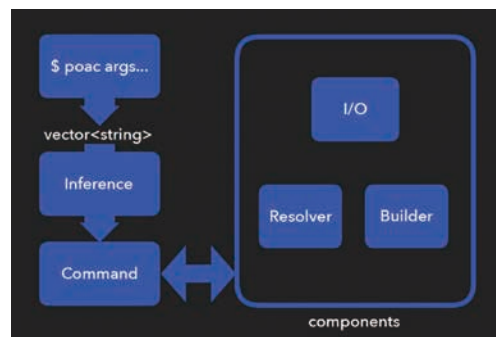
```

$ brew install cmake
$ brew install conan
$ mkdir hello
$ cd hello
$ run conanfile.txt
$ conan remote add conan-travis https://api.bintray.com/conan/conan:conan-travis
$ conan install --build
$ mv main.cpp
$ mv CMakeLists.txt
$ cmake
$ cmake --build
$ ./bin/greet
Hello World!

$ curl -fsSL https://sh.poa.cio | bash
$ poac new hello
$ cd hello
$ poac install hello_world
$ mv main.cpp
$ poac run
Hello, world!

```

■ 図-13 Conan と poac の比較とロゴ



■ 図-14 poac のクライアントアーキテクチャ

最初の学習は図-15のように、AR タグをつけたロボットを上からカメラで観察しながら行う強化学習である。どちらにもマイコンが搭載されており、ZigBee 通信で位置情報をロボットに伝える。

最終的な実験に用いたのは PhantomX という 6 脚ロボットである。常時 5 脚が接地している ripple, 4 脚が接地している amble, 3 脚が接地している tripod など、多種類の歩行方法があり、脚を故障させるたびにこれらの基本歩行方法を調節するように学習させた。なお、普通に強化学習するととんでもない歩き方(?)を学習することがあるので、脚先端以外の接地を認めないとか、サーボモータは中心からプラスマイナス 45° 以内でしか動かせないといった制約を加えた。強化学習のこういうノウハウは継承されていくべきものだろう。

次に故意に脚を「故障」させて、再学習させる。これに 10 ~ 20 分しか要しなかったというのがこのプロジェクトの最大の特徴である。通常、こんな短い時間で回復学習はできない。図-16 は右中脚と左後脚を故障させたときの様子である。真ん中の写真のように、重量負荷バランスが崩れ、よたよたと右へ曲がってってしまう。しかし、学習して改善すると、正常時に比べてそれほど劣らない直進性と速度を達成している。成果報告会ではさすがに実時間での学習をさせる余裕はなかったが、この写真と同じように動作した。

実はこのプロジェクト、当初の計画とかなり異なる道を歩んだ。素早い歩行学習のため、生物が活用している CPG (Central Pattern Generator)、つまり歩行に

関係する周期信号を生み出す神経回路の模倣をまず試みた。これが予想に反してまったくだめ。次に別の深層強化学習とシミュレーションを組み合わせた方法も試みたが、これも実機適応がだめ。しかし最終的に、非常にコストの安い、実機だけで済む強化学習に落ち着いた。まさに村松君自身がすばらしい学習を遂げたプロジェクトだった。(田中邦裕 PM 担当)

やました たくみ
■ 山下 琢巳

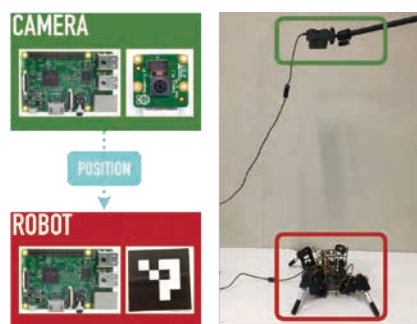
クリエイター向けのブロックチェーン Proskenion^{☆11}

ビットコインを支える技術として生まれたブロックチェーンは、最近の大きな技術トレンドになっている。仮想通貨(暗号資産)のためだけでなく、選挙, 所有権, 経歴, 資格など, いろいろな「保証されるべき資産」の, 改竄不能な, 非中央集権的で単一障害点耐性のある分散台帳型の管理に使える社会インフラになるのではないかという期待が高まっているからである。そのため, いわゆる「俺のブロックチェーン」提案が相次いでいる。特に金融関係では大規模な開発が群雄割拠している。

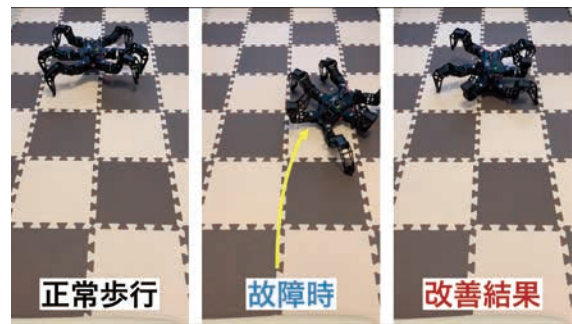
そんな中, 山下君は, エンタテインメント領域におけるコンテンツ配信を, クリエーター中心の立場で支えるブロックチェーン技術を開発した。名前は Proskenion で, 観客席から見て舞台を額縁のように区切る構造物を指すギリシャ語に由来する。

現在, App Store, Google Play, Amazon Kindle な

☆11 <https://github.com/proskenion/proskenion>



■ 図-15 前進の強化学習



■ 図-16 脚を故障させたときの修正

どの中央集権企業による、クリエイターのコンテンツ流通の寡占化が進んでいる。その手数料は驚くほど高い。流通業者よりもコンテンツの一次供給者が評価される社会に変えようというのが Proskenion の狙いである。つまり、クリエイターが頒布プラットフォームを提供するシステムの運営者をクリエイターが選ぶ、クリエイター主体の収益化プラットフォームを作ろうというわけである。

表-1 にその基本アイデアを示す。仕組みは後述するが、クリエイターが良いコンテンツを頒布する行為を「マイニング」とする、いわば Proof of Creator を合意形成の基本とするのである。これでも、システム運営者にはトークンの価値向上というメリットがある。

このほかにも、オンラインでのコミックマーケットとも言えるような同人誌即売会、ミスコンのような短期的投票システムに応用可能である。

このような応用を可能にするのが、ブロックチェーン Proskenion の 3 つの特徴である。

- (1) プリミティブな命令セットの組合せで高い表現力を持つ
- (2) 独自 DSL である Proskenion Domain Specific Language (ProSL) によってインセンティブ／合意形成のカスタマイズが容易である
- (3) インセンティブ／合意形成を動的に変更する際に、ブロックチェーンを互換性のない 2 つのブロックチェーンに永続的に分岐する、いわゆる「ハードフォーク」をしなくてよい
特に (3) はエンタメ領域に役立つ。ProSL はチューリ

ング完全ではなくしてあるので、予期せぬ無限ループが起こらないようになっている。なお、Proskenion は誰でも参加できるパブリックなブロックチェーンである。このあたりを表-2 にまとめた。

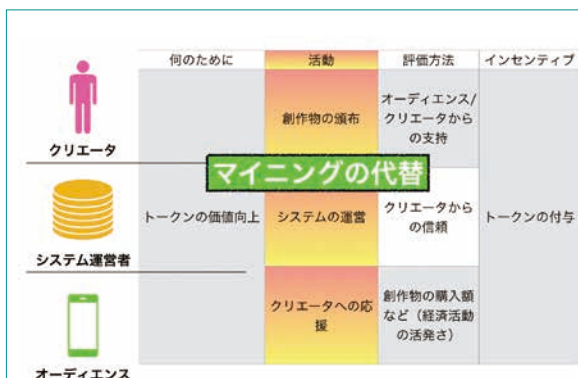
上述の Proof of Creator の 1 つの実現例を ProSL ではなく、以下、日本語で書いてみよう。クリエイター同士の関係をフォロー／フォロワーを示す有向グラフで表現する。有向グラフから PageRank を計算して出した「クリエイターの rank」をクリエイターの信頼度とする。信頼度の上位 4 人によって指定されたシステム運営者たちがブロックの生成を順繰りに行う。

こうして、より良いクリエイターがノードを選択するクリエイター主体の運営システムが誕生するというわけである。このほかにもクリエイターの信頼度の更新のタイミングなどはプログラムをいじればいつでも変更できる。Proskenion は GO 言語をメインに使用して書かれたかなり大規模なプログラムである。

山下君はすでに Staked (株) の創業メンバとして CTO になっている。(竹迫良範 PM 担当)

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) Giverty 技術顧問、東京大学名誉教授。本会フェロー。



■表-1 クリエーター中心のブロックチェーン

比較表	Ethereum	Xem	Ripple	Fabric	Proskenion
Public/Private	Public	Public	Private	Private	Public
合意形成の柔軟性	○ PoW→PoS	△ PoI	△ Ripple Consensus	△ Endorsement Policy + Ordering Service	◎ DSL で記述
扱える対象	◎ チューリング完全	△ 通貨only	△ 通貨only	◎ チューリング完全	○ プリミティブな命令群を用意
特徴/用途	・万能だが重い ・法としての用途に有用	・高速かつ安定 ・金融機関での利用に特化	・高速 ・国際送金の決済基盤に利用	・高速かつ ・Ethereum 並みの自由度 ・BaaS 領域に有用	・ハードフォークをメイン ・センティブ/合意形成 ・設計が変更可能 ・社会実験/エンタメ ・領域に有用

■表-2 他のブロックチェーンとの比較