2025 年度第５回ロボフレ委員会講演内容
～生成 AI と歩んだ DX の一年～

講師:福知山公立大学　西田豊明 、記録：RRI　西垣戸貴臣

概要：講演者は AI 研究者としての経験を持ち、福知山公立大学副学長として大学 DX 改革を牽引した。従来のようにシステム部門に依頼するのではなく、生成 AI を活用して自らソフト開発を行い、AI との対話を通じてコード生成と改良を繰り返すことで、ノンストップで動く実用的なシステムをわずか２人で構築した。

（AI のこれまでの歴史）AI は 2 度の「冬の時代」を経て発展し、2000 年代後半の技術革新とコンピュータやインターネットの進展によって現在の生成 AI 時代を迎えた。講演者は著書『AI が会話できないのはなぜか』（2022）で、人間らしい会話に不可欠な「コモングラウンド（共有意識）」の欠如を指摘。ChatGPT の登場により AI がこの共有意識を一定程度理解できるようになり、人間に近いコミュニケーションが可能になった。

（生成 AI を使った学内の DX 改革）2020 年のコロナ禍では、学部長として大学運営のデジタル化を主導。Slack による資料共有や Zoom 会議の定着、出勤・稟議管理の電子化を実現した。副学長就任後は、生成 AI を活用して大学業務を効率化するシステム「FUJIN（Fukuchiyama Universal Joint Information Nexus）」を開発。会議室や公用車の管理から報告書作成まで支援する仕組みを 2 名体制で構築し、2024 年に全学運用を開始した。自然言語による指示で生成 AI がコードを改良する開発手法を確立。短期間で安定稼働を実現し、学生にも開発参加の機会を提供している。

（取り組みを通じて学んだ事）少人数でも生成 AI を活用すれば効率的なソフト開発が可能であり、明確な目的意識と信頼構築が成功の鍵になる。「起業家養成スクール」では、学生が生成 AI を使ってシステム開発を体験する教育プログラムを実施し、FUJIN の活動を通じて得たノウハウを若い世代に教育した。AI 時代には「コードを書く力」よりも「目的を設定し意思決定する力」が重要であり、人間が責任をもって判断できる素養を育むことが重要になる。深い素養と知識を持つエンジニアが増えていくことを望みたい。

１．はじめに

AI 研究畑を歩み、その後、大学の学部長を経て副学長になられた講演者が、大学のＤＸ改革に取り組んだ。今回の講演者は、ソフトウエアのリテラシーはある。しかし、最新の言語には詳しくは無い。ソフトを作ってくれるたくさんの部下がいるわけでもない。一方で、大学の事務の仕組みや内部の組織構造には詳しい。やりたい改革も頭の中にある。この場合通常であれば、システム屋と会話しながら仕様書を作って、ソフトを試作して試用する。試験運用の中で、仕様書の段階では思っていなかった課題や思い違いなどに気づき、ソフトを改良する。こうしたループを繰り返してシステムとして仕上げて行くのが通例である。

今回は、講演者自らが生成 AI を使ってソフトを試作・改良して、α版を完成させた取組を紹介する。講演者が生成 AI に注文を出し、生成 AI がコードを吐き出し、それをすぐにテストする。気に入らない部分があれば生成 AI に注文を出し、生成 AI が改良版のソフトを吐き出す。この小ループを１人で回す。ある程度納得するものができたら、それをソフト技術者に渡す。ソフト技術者も１人。この技術者がプロの目でコードを整備・点検して、学内のテストに臨む。本稿では、こうしたトライアルを通じて、大きなトラブルなく動いたソフトをたった２人のチームで作り上げた例を紹介する。第２章では、取組のバックグラウンドとして、AI のこれまでの歴史を振り返る。第３章では、生成 AI を使った学内のＤＸ改革の取組を紹介し、第４章で、こうした取り組みを通じた学んだことを紹介する。第５章はまとめである。

２．AI のこれまでの歴史

２．１　バックグラウンド：AI の歴史

バックグラウンドとして、最初にコンピュータサイエンスと AI の歴史を振り返る（図１）。

コンピュータサイエンスがはじまったのは 1936 年のチューリングマシンだと言われており、AI の研究が本格的にはじまったのは 1956 年と言われている。

そのルーツは、20 世紀の初めに生まれた計算哲学に遡る。人間の思考の間違いをなくすために考案された論理学の考えが数学化されてチューリングマシンになった。その後フォン・ノイマンがコンピュータを作り、商用コンピュータが生まれた。これをツールとして使ってAIのようなことを考えようという

人が現れ、AI の研究が始まった。

その後､AI の歴史は光と闇を繰り返すことになる。前述した通り、1956 年に夜明けを迎えるが、直後に実力が伴っていないことがわかって冬の時代を迎える。1970 年代から 80 年代にかけて、AI がビジネスの役に立つことを示した人々が現れ、再度希望の光が見えて来た。その後、まだビジネスに使用するには実力不足ということがわかり、長い冬の時代を迎えることになった。

2006 年から 2011 年にかけて深層学習のインパクトで、再び AI ブームが訪れ、それが今まで続いている。この進化の裏には、ムーアの法則に代表されるコンピュータの進化や、インターネットの進化によってデータの共有が進んだことなどがあり、これらが相まって AI の進化を支えてきた。

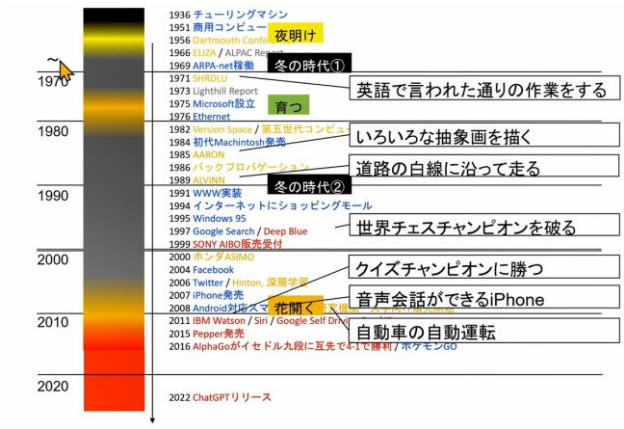

図１　AI の歴史

２．２　生成 AI 出現に至るまで

講演者は 2022 年に「AI が会話できないのはなぜか」という本を出した（図２）。

図２　AI が会話できないのはなぜか

人間同士の言語によるコミュニケーションは対話と会話に分類される。交渉によって対立の解消を図り、収束させるのが対話である。他方、話題の発展でワクワク感を発生させ、安らぎを与えるのが会話である。

講演者は、これまでの AI では会話できるという感覚が起きないと感じていた。それは、当時の AI には人との間にコモングラウンド（共有意識）をつくり、発展させる応用力が無いからであると考えた。人間同士のコミュニケーションでは、共有意識を作り上げて広げて行くことで会話が弾む。AI と会話出来ないのは、AI と一緒にコモングラウンドを共に発展させているとは感じられないからである（図３）。対話にもコモングラウンドは使うが、それは手段であり、問題解決のための最小限のコモングラウンドがあれば良い。要点を切り出して短く終わるのが良い対話である。他方、会話ではコモングラウンドは目的である。共有される意識を豊かにすることが会話の目的であると言ってもいい。良い会話は長く続き、共有意識が広がっていく。この頃の AI にはこれが無かった。

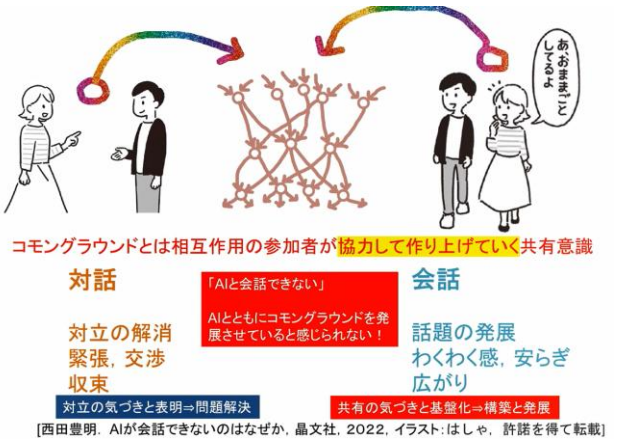

図３　コモングラウンド

2022 年 2 月にこの本を出した直後の 2022 年 11 月に ChatGPT が出て来た。これまでのような単機能では無く、何でもできる AI である。その前にも兆候はあった。GPT-2 や 3 を使ってみた。しかし実用的とは思えないものであった。しかし ChatGPT（GPT3.5）は全く違ったものとして出現した。汎用 AI と言っても良いものがすぐ手近に使えるものとして世の中に現れた。今の生成 AI はしばらくの間ならコモングラウンドを理解し広げていくことができるようになったが、まだそれをセッションを越えて続けていくことはできない。

２．３　生成 AI

生成 AI はとても単純な方法で動いている。大量の

文章を集めて事前学習すると自分の中に地図のようなものができる（図４）。このような状態で、例えば「東大寺の西には」という文章を入力すると、「東大寺は興福寺の東」と言う情報や「西は東の反対」等の情報から、次に関係がありそうな文章として、「興福寺があります」という出力を出してくる。

大型の機械学習（トランスフォーマー）は非常に高い言語能力をもつが、なぜそういうことができるようになったかを説明することはできない。

ベーシックな使用法は、基本的には言葉のやりとりである（図５）。生成 AI に何かを言うと、何かを返してくれる。それが非常にユーザの役に立つ。

生成 AI の国語力は強力である。少なくとも数１０分程度の会話に対応した国語力は非常に高い。大量のデータを学習済みであり、いろんなことを教えてくれる。ありがたいことに日本語で話しかけると日本語で返事をしてくれる。他の言語で話しかけて、返事してもらうこともできる。返事の中にはプログラムコードが含まれる（図６）。これを活用して IT を使った新しい価値を生み出すことを考えた。

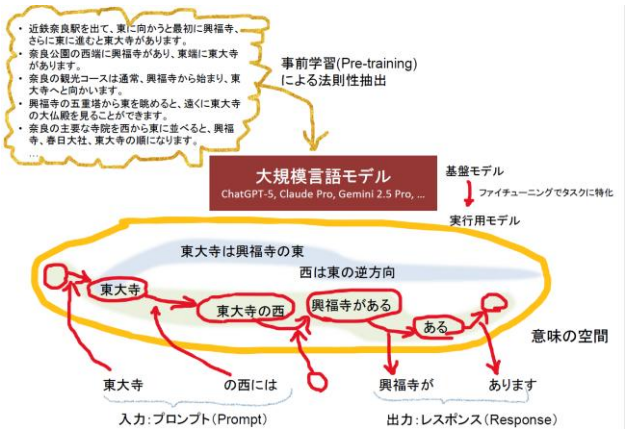

図４　大規模言語モデル

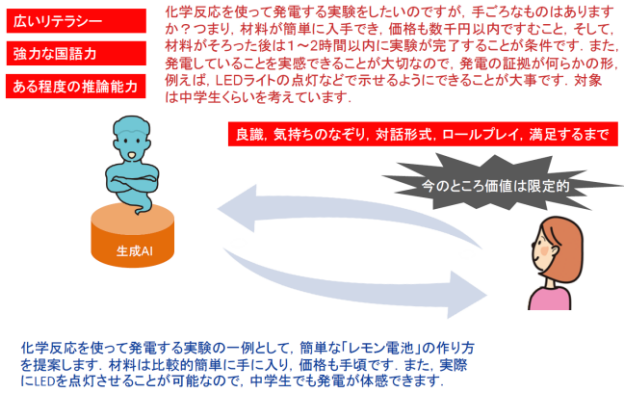

図５　生成 AI の通常の使い方

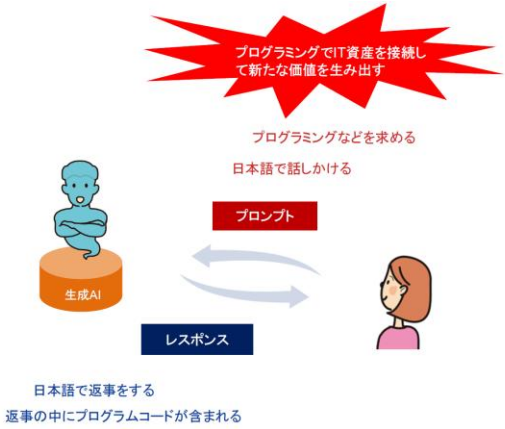

図６　プログラムを吐き出す AI

３．　生成 AI を使った学内の DX 改革

３．１　福知山公立大学でのあゆみ

講演者は 2020 年に福知山公立大の学部長に就任した。この年に新型コロナウイルスによるパンデミックが発生。入学式は無し。学生の登校も無し。学校運営をどうするのかを考えないといけない状況に陥った。こうした中で情報学部に関しては、資料は Slack と呼ばれるチームコミュニケーションツールを使って共有し、会議は Zoom によるオンライン形態で実施する体制を作った。その後、出勤管理や稟議書管理のデジタル化にも取り組んだ。パンデミックが収まった今でも教授会は Zoom を使ったオンライン形式で実施している。世の中には対面型でないとしっかりした議論は出来ないという人も居るが、少なくとも情報学部では、オンラインの形態が続いている。Zoom には盗聴のリスクが存在するため、入学試験の合否判定や試験問題の議論等は対面型で実施している。

2022 年度から副学長に就任した。情報学部長も兼務したが、2022 年度いっぱいで退任した。結果として比較的自由な時間ができた。前記した通り、この頃生成 AI が出現した。これを利用して、時空資源管理（会議室や公用車の管理）、年間計画・報告書の共同編集などの機能を持つ学内の総合的な DX システムの開発を始めた（2024 年）。

３．２　FUJIN システム

このシステムを FUJIN（Fukuchiyama Universal Joint Information Nexus）と名付けた。なお、この英語名称は、講演者が地元の両丹日日新聞に執筆していたコラムシリーズ「風塵のアトラクター」の「風塵」の頭文字をとって、生成 AI が作ったものである。

FUJIN には３つのレベルのシステムある（図７）。

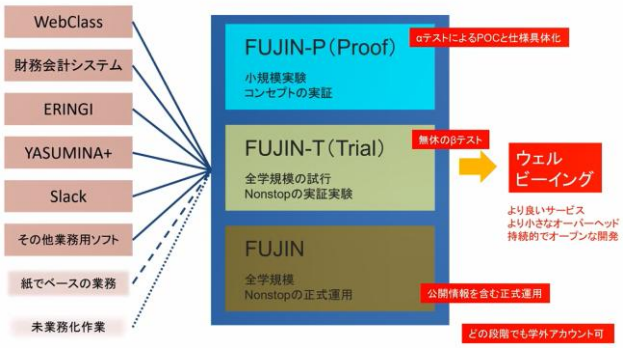

図７　FUJIN システム

FUJIN-P は Proof 版。α テストによる POC と仕様具現化のためのバージョンであり、小規模実験やコンセプトの実証のためのものである。FUJIN-T はトライアル版。無休のβテストを行うバージョンであり、全学規模の試行やノンストップの実証実験のためのものである。テストだからと言ってトラブルで停止してしまうと使っていただけない。設計段階から「計画停止を最小化し、原則として無停止運用を目指す」方針とした。結果として FUJIN-T は、これまでトラブル停止や計画停止を行わずに安定稼働している。FUJIN は最終の正式運用版であり、FUJIN-T をベースに今年度内に運用を開始することを目指している。

FUJIN では、プライバシーに関連するものやセキュリティが重要な業務は扱わない。そういうものは単独の業務システムとして分離し、必要に応じて結果のみを FUJIN に取り込む。例えば入学試験の作成や採点等のセキュリティの高い業務は従来のやり方で実施し、志願者数や合格者数などは FUJIN に取り込んで広報等の情報に利用する。FUJIN-P は講演者が 1 人で開発している。FUJIN-T は 1 人のソフトエンジニアが担当している。FUJIN-P、FUJIN-T 併せて 2 名体制で開発してきた。最終形態である FUJIN に関しても FUJIN-T を担当したソフトエンジニア 1 人で担当可能だと考えている。

３．３　FUJIN の足取り

図８に FUJIN のこれまでの足取りを示す。

図８　FUJIN のこれまでの足取り

2024 年 9 月に FUJIN-P の開発を開始した。すぐに大学のクラウドサーバを導入し、FUJIN-T の開発を始めた。12 月に、FUJIN-T で作った最初のアプリ（会議室、公用車等の管理システム）が動きはじめた。年明けに全学に対する説明会を実施し、2 月に古い会議室予約管理システムを停止し、完全に FUJIN-T の時空資源管理システムに切り替えた。今のところこのシステムは無事故で動いている。

小さい大学ではあるが、授業のための予約数は年間 7,500 件に及ぶ。これに加えて、試験期間やオリエンテーションや不規則的な講義にも対応する必要がある。更に、入学試験等の重要な予約に関しては、早めに予約が可能になるだけでなく、もし、事前に予約が入っていた場合は、特別権限で元の予約を解除できる等の新たなルールも制定した。こうした形で、このシステムが FUJIN-T の上で正式運用されている。近いうちに最終版である FUJIN を立ち上げ、この上での運用に移行することを計画している。

３．４　生成 AI の具体的な活用方法

生成 AI に助けてもらった開発過程を説明する。講演者は Python を本格的に学んだ経験はないが、他言語での開発経験はあるため、プログラムの構造や実装の考え方は理解している。会議室予約のシステムを作ったときは、「こういう画面遷移をして、こういうモジュールを持ったプログラムを作って欲しい」という内容の 2,000-3,000 文字程度の指示を生成 AI に与えた。詳細な仕様は与えていない。生成 AI がプログラムを提案してくると、動かしてみて、ここはこうしてほしい、あそこはこうしてほしいと言った注文を繰り返し与えて改良を重ねた。基本的にはソースコードには手を触れず、生成 AI に自然言語で指示して調整を進めた。ただし、セッションの容量制限に近づいたときは、ソースコード全体をダウンロードし、次のセッションに読み込ませて「これをベースに修正して」と引き継ぐことで、改良を継続した。安定して動くプログラムは 1000 行までくらい。生成 AI は指示内容を理解し、コードの適切な箇所を修正する。最初の頃はバグで止まることもあったが順調な場合は５分程度で完了する。逆に、うまくいかない時は数時間かかる場合もある。混乱がひどくなると、覚悟を決めて、それまでに作ったプログラムを全部捨てて作り直すことあった。こういう部分は人間がプログラミングする場合と同じである。

３．５　実際の稼働画面

図９に FUJIN-P の実際の稼働画面を示す。

図９　FUJIN-P の稼働画面

図中の箱一つが一つのアプリに相当する。最上段のオレンジの箱が特定の権限を持った管理者用のアプリである。管理者の任命や承認等に使う。その下のブルーの箱が一般用のアプリである。先ほど説明した会議室の予約申請のアプリなどいろいろな実験システムを開発した。更に下のグレーの箱は開発中のアプリを示す。FUJIN-P のアプリの完成度が高まると FUJIN-T に移行させる。逆に使えないものは却下する。将来は、学生の作ったアプリも、こちらで試験して、うまくいけば採用するようにしたい。採用されれば、学生にとってもプロジェクトの成果としてカウントすることでチャレンジ精神を高めたい。生成 AI の活用によって、半年強でここまでのシステムができた。

４．FUJIN への取組で学んだ事

４．１　学んだ事

２人でやったところが強みである。人手が足りない部分は生成 AI をフル活用した。意思決定のための大きな会議は不要であり、上司にお伺いを立てる等のプロセスも不要であった。

易しい問題からスタートした。セキュリティ、個人情報、プライバシーが問題にならない領域で、かつ、明確なメリットを示すことができる分野に注力した。

出口からスタートし、だんだん業務に近づくアプローチにした。出口は大学の執行部からの需要が大きい。学長等の需要が高く、圧力もかけやすい出口からスタートし、業務に近付いていく戦略とした。

明確な目標も重要。みんなをハッピーにするのが目的であるということを前面に出しながら進めた。

古いシステムと新しいシステムの切り替えのオーバーラップは短くした。新旧システムの併用は安心感はあるものの、作業量を増大させ、負担が大きくなるので、事務方からの抵抗が大きくなる。他方、事故や検討不足で一回でも皆に迷惑をかけてしまうと、信頼を失うので、絶対に避ける意識で進めた。

４．２　起業家養成スクールでの試み

この夏の起業家養成スクールで 20 名の学生を担当した集中講義を担当する機会があった。そこで、FUJIN を小型化した「といの森」という学習用レベルのシステム開発を題材とすることにした（図 10）。

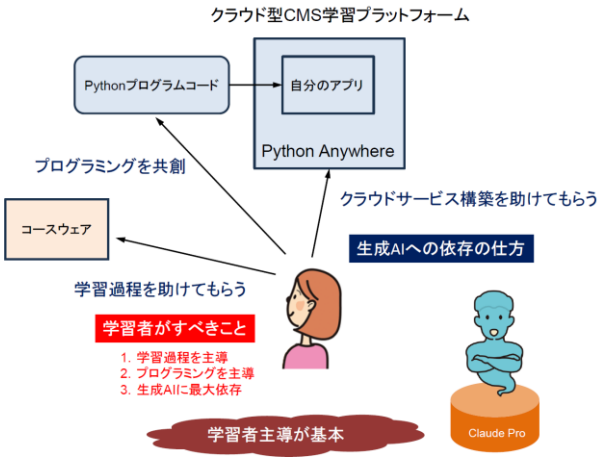

図１０　といの森

「といの森」は、ヤフー知恵袋のように、ユーザが疑問点や情報を持ち寄り、共有する協働型ツールである（図 10）。

コンテンツの作成・管理、ユーザ管理、カテゴリー分け、公開・非公開の制御等、４つの遷移画面からなる（図 11）。受講生にはこれらを生成 AI を使って作るコースを考えたが、事前チェックで、ハードルが高すぎて脱落者が多数出るのではないかというコメントがあったので、図 12 に示す 3 コース制を導入した。

入門コースでは、基本ツールである Python Anywhere を使う、生成 AI を使ったコーディングをする、等の基本動作を習得し、最終目標として電卓を作ることにした。中級コースでは、講演者が作った「といの森」のアプリのソースコードを全て開示し、これを組み立てて作る。余裕があれば、生成 AI で改良する。上級コースは、プロンプティングによって、生成 AI にプログラムを作ってもらうというフルスペックである。講演者も Slack と Zoom で質問を受け付けることにした。講演者は Slack で経験を共有できるようにした。

結果として講習はうまく行った。20 名の受講生に求めた事前スキルは IT リテラシーだけであり、プログラミング経験は求めなかった。20 名のうち 3 名が上級コースにチャレンジし、自らプロンプティングを行ってシステムを完成させた。中級コースの 9 名がソースコードを使ってシステムを完成させた。残りのほぼ全員が電卓作りまでたどり着いた。多くの

受講者が生成AIに質問していろいろな問題を解決し、事前に心配していた、講演者に質問が殺到して回答できなくなるという事態は発生しなかった。

図１１　といの森の構成

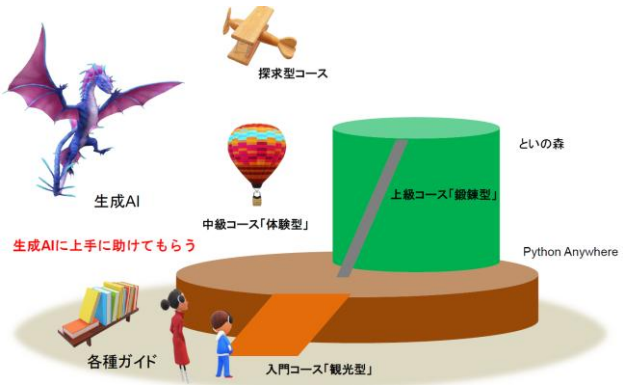

図１２　といの森の各種コース

４．３　さいごに

今後はプログラミング教育のやり方も変化すると思われる。具体的なプログラミングは生成AIに任せるとして、これまで教育されてこなかった「目標をたてる」という部分が重要になる。自動車が登場して馬の使い手の職業は無くなった。同じことが起きるかも知れない。

一方、人間にとって重要な役割は労力配分などの決定である。AIはヒントを与えてくれたりアドバイスをくれたりするが、最終決定は人間の仕事。それが重要。責任は人間。責任をとるためには素養が大切。あらゆる場面での問題が多くなる、そうなると素養をもったたくさんの人間が必要になる。深い素養と知識を持つエンジニアが増えていくことを望みたい。

５．まとめ

【DXへの取り組み】

AI研究者としての経験を持ち、大学の副学長としてDX改革を進めた講演者は、生成AIを活用して自らソフト開発を行った。システム部門との従来型開発ではなく、AIに文章で指示を伝え、生成されたコードを即時テストし、修正を繰り返す小さな開発ループを１人で回した。完成後はソフト技術者がセキュリティ面を調整し、学内テストを実施。結果、わずか２人でトラブルなく実用的なソフトを構築した。

【AIのこれまでの歴史】

コンピュータサイエンスは1936年のチューリングマシンに始まり、AI研究は1956年に本格化した。以後、AIは期待と失望を繰り返し、2度の「冬の時代」を経験した。2000年代後半、コンピュータ性能の飛躍的向上とインターネットの普及、データ共有の進展によりAIは再興し、現在の発展につながった。

講演者は2022年2月に『AIが会話できないのはなぜか』を著し、人間の会話に不可欠な「コモングラウンド（共有意識）」の欠如が、AIとの自然な会話を妨げていると指摘した。従来のAIにはこの共通基盤がなかったが、生成AIは少なくともセッション中には共有意識を理解し、より人間らしい応答が可能になった。著書刊行のわずか数か月後、ChatGPT（GPT-3.5）が登場し、単一機能ではなく多用途で実用的なAIとして社会に衝撃を与えた。生成AIは膨大な文章データを学習して言語の関係性をマッピングし、与えられた文脈に最も適した応答を生成する。仕組みは単純に見えるが、その内部の推論過程は未解明である。高度な国語力を持ち、自然な日本語で会話やプログラム生成も行える生成AIは、知識活用や新しい価値創出のための強力なツールとなりつつある。

【生成AIを使った学内のDX開発】

2020年に福知山公立大学情報学部長に就任した講演者は、コロナ禍で入学式や登校ができない中、大学運営のデジタル化を進めた。Slackで資料を共有し、Zoomで会議を実施する体制を整備。出勤管理や稟議書の電子化にも取り組み、現在も教授会はオンラインで継続している。2022年に副学長となり、生成AIの登場を受けて大学内業務を効率化する「FUJIN（Fukuchiyama Universal Joint Information Nexus）」の開発を開始。会議室や公用車の管理などを行う時空資源管理システムを中心に、年度計画や報告書作成などを支援する仕組みを構築した。FUJINはP（Proof）、T（Trial）、最終版の3段階で開発。Pは講演者自身が、Tは1名のソフト技術者が担当し、わずか2名体制で進められた。セキュリティを要する業務は別システムで処理し、結果のみをFUJINに統合する方針を採用している。2024年9月に開発を開始し、12月に初期版を稼働。翌年2月には全学で本格運用を開始し、ノンストップで稼働を続けている。会議室の予約数は年間7,500件を超え、特別権限による予約解除などの新ルールも導入。生成AIを

活用した開発では、講演者が Python を知らずとも、自然言語による指示でコード生成と修正を繰り返し、安定したシステムを構築した。半年余りで完成した FUJIN は、学生もアプリ開発に参加できる柔軟なプラットフォームへと発展している。

【FUJIN への取組で学んだ事】

講演者は、福知山公立大学での生成 AI 活用を通じて多くの学びを得た。最大の強みは、わずか 2 名のチームで柔軟に開発を進め、人手不足を生成 AI で補った点にある。意思決定の迅速化や大規模会議の排除により、スピード感ある改革を実現した。また、リスクの少ない領域から着手し、成果を示しながら信頼を獲得する戦略を採用。大学執行部など影響力の大きい「出口」から導入を進め、段階的に業務に近づける形で展開した。目的は「皆をハッピーにすること」であり、その理念を明確に示すことで周囲の理解を得た。

さらに、講演者は生成 AI 教育にも挑戦した。夏の「起業家養成スクール」では、FUJIN 開発の手法を応用し、20 名の学生に小規模システム開発を体験させた。題材は協働型 Q&A ツール「といの森」で、学生は生成 AI を活用しながら段階的に学習。上級者は自らプロンプト設計を行いシステムを完成、中級者は既存コードを活用し、初級者も電卓アプリを完成させるなど高い成果を上げた。ここで重要なのは、学生が自力でコードを書くのではなく、「生成 AI に何をどう指示すれば目的に近づけるか」を学ぶ点である。

今後のプログラミング教育は「コードを書く力」よりも「目標を立てる力」の育成が重要になる。AI が実装を担う時代には、人間がリソース配分や最終決定を行う責任を負うべきである。深い素養と判断力を持つ人材の育成こそが次代の教育課題になる。