SDV実現のためのロボット開発用クラウド基盤の構築

Building Cloud Platform for Robot Development to Realize SDVs

●中　尾　拓　真*　　藤　井　北　斗*
Takuma NAKAO　　Hokuto FUJII

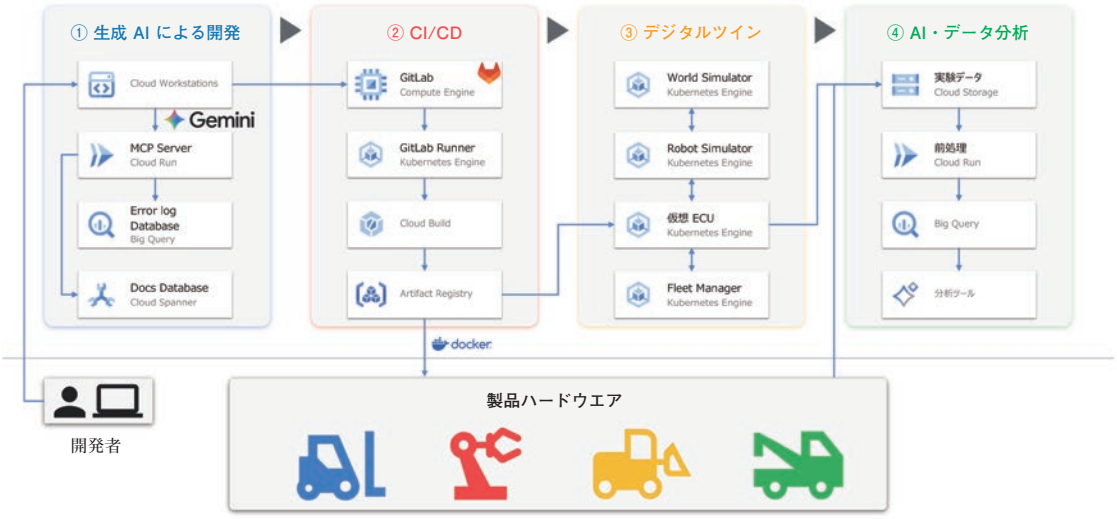

図 1　全体アーキテクチャ
System architecture overview

物流・搬送現場の労働力不足を背景に，ソフトウエアアップデートによって製品機能が進化し続けるSDV（Software Defined Vehicle）の実現が求められている。当社はSDV検証用自動搬送フォークリフト（AGF：Automated Guided Forklift）の開発のための，生成AIとGoogle Cloud*1を用いた次世代のロボット開発基盤を構築した。本基盤は生成AIによる開発支援，CI/CD，デジタルツイン，データ分析の４層を備えている。具体的には，チーム全体でソフトウエア開発環境を統一できるCloud Workstations，エラー解決ナレッジデータベース，CI/CD管理基盤，コードとドキュメントのグラフRAGなどを開発した。また，シミュレータによるAGF動作の机上検証機能も備えており，実機のみでの開発と比較して開発効率が向上した。本クラウド上の開発基盤の構築事例および，SDV検証機であるAGFの開発事例を解説する。

Against the backdrop of labor shortages in logistics and material handling, there is a growing demand for Software-Defined Vehicles (SDVs) whose functionality evolves through continuous software updates. To address this need, we developed a next-generation robot development platform that leverages generative AI and Google Cloud to build an Automated Guided Forklift (AGF) as a vehicle for SDV verification. The platform comprises four layers: generative AI-assisted development support, CI/CD, digital twins, and data analytics. Specifically, it integrates Cloud Workstations to unify the development environment across the team, a knowledge database for error resolution, a CI/CD management infrastructure, and a Graph-RAG system for code and documentation retrieval. Furthermore, the platform enables simulator-based desk verification of AGF behavior, thereby improving development efficiency compared to conventional methods relying solely on physical hardware. This paper describes the architecture of this cloud-based development platform and presents a case study on developing the AGF for SDV verification.

1　はじめに

SDVの価値は，ソフトウエアを通じて製品価値を継続的に高め，新たな収益機会を創出できることである。たとえば，スマートフォンは購入後もアプリ追加やOS更新ができる。これを自動車で実現したものがSDVで，OTA(Over The Air)ソフトウエアの更新によって，自動運転機能の追加や燃費の向上などを実現する事例が見られるようになった(1)。製品の販売後も，ユーザが機能の購入やサービスの契約を継続できることで，新たな付加価値が生まれる。

SDVの場合，ソフトウエアの開発サイクルを短い間隔で実行することが，ユーザの満足度に直結する。常に魅力的な機能やサービスが手元に届き，製品が進化し続けるという体験をユーザに提供することで納入後の顧客満足度向上につながる。

しかしながら，重工業ではソフトウエアの開発サイクルを高速化することは難しい場合が多い。ソフトウエアの開発サイクルは実装，検証，配信，フィードバックの収集からなっており，ハードウエアを伴う開発では，実機検証がそのボトルネックとなる。場所や安全への制約によって実機検証には時間がかかるので，単にソフトウエア実装だけを高速化しても開発サイクルの全体の高速化にはつながりづらい。

＊技術本部

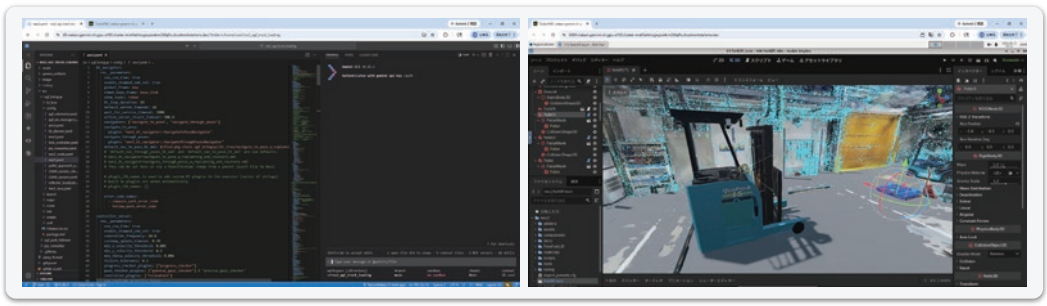

図2　クラウド上の統合開発環境
Integrated development environment in cloud

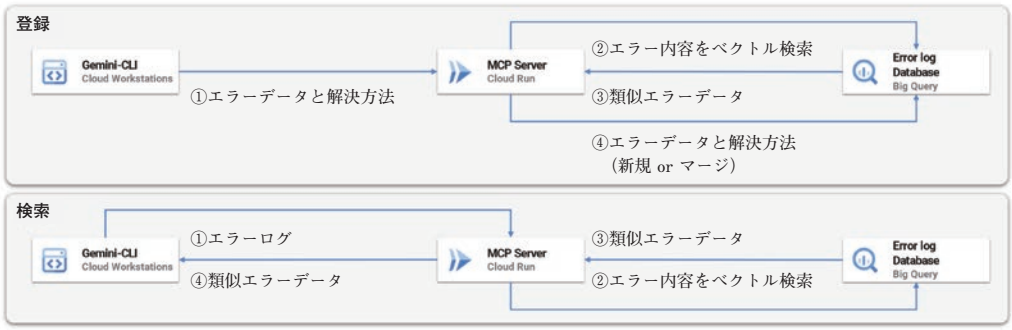

図3　エラー解決ナレッジベースの処理フロー
Workflow of error resolution knowledge base

そこで，SDV実現のためのロボット開発用クラウド基盤を構築し，実際のロボット開発に使用することで有用性を検証した。この開発基盤には，ソフトウエア実装の高速化を目的としたAIコーディングエージェントやCI/CDに加え，実機検証の省力化のためのシミュレータ環境が含まれる。本報ではAGFを検証の対象とした。

2　次世代ロボット開発基盤の全体アーキテクチャ

開発プロセス全体をGoogle Cloud上で安全かつ高速に循環させ，開発全体のリードタイムを短縮することをロボット開発基盤の基本方針とした。

本基盤は，開発加速に向けた要求・要件を満たすべく，次の①～④の機能層から構成される。**図1**に，全体アーキテクチャを示す。

① 生成AIによる開発支援層：コード生成やエラー修正の支援を行い，開発者のアイデアを迅速にソースコードへ変換する役割を担う。

② CI/CD管理層：ビルド，テスト，およびドキュメント作成などの定型業務を自動化・標準化し，開発者が本質的な作業に注力できる環境を提供する。

③ デジタルツイン層：3Dシミュレータを活用したテスト自動化環境を構築し，物理的な場所や安全性の制約を受けることなく，高速な検証イテレーションを可能にする。

④ データ分析層：シミュレーションおよび実機検証から得られた実験データを集約・分析し，次の改良フェーズにフィードバックするためのデータパイプラインを形成する。

これら各機能層をGoogle Cloud上で統合することにより，ハードウエアへのソフトウエアインテグレーションや実機検証などを効率化するロボット開発基盤を実現した。

3　実装した要素技術および機能

2章で定義した各機能層における具体的な要素技術および機能について解説する。

3.1　生成 AI とクラウドによる開発支援層の機能詳細

本層は，開発者におけるコーディングおよびエラー解決プロセスを支援し，要求機能を迅速にソースコードへ変換することを目的とする。開発環境の提供から知見の構造化に至る3つのシステムを包括的に実装している。次に開発した3つのシステムについて説明する。

3.1.1　Cloud Workstations による開発環境の一元化と標準化

ソフトウエア開発環境を組織内で一元管理するべく，Google Cloud のクラウド上の統合開発環境である「Cloud Workstations」を導入した。Cloud Workstationsは，事前に定義した開発環境を実行できるサービスである。本基盤では，開発ツールや，コーディングエージェント (CA) をあらかじめインストールした開発環境を作成して展開した。これにより，チーム内の開発者は開発環境導入時のインストールや複雑なセッティングをすることなく，CAを含む複数の開発ツールを利用できる。本環境はクラウド上に構築されているため，インターネットにアクセスできる場所であればどこでも開発することが可能となる。また，開発環境のCPU性能やメモリ容量やGPUの有無などを自由にスケールさせることもできる。クラウド上の開発環境のセキュリティ面においては，Google

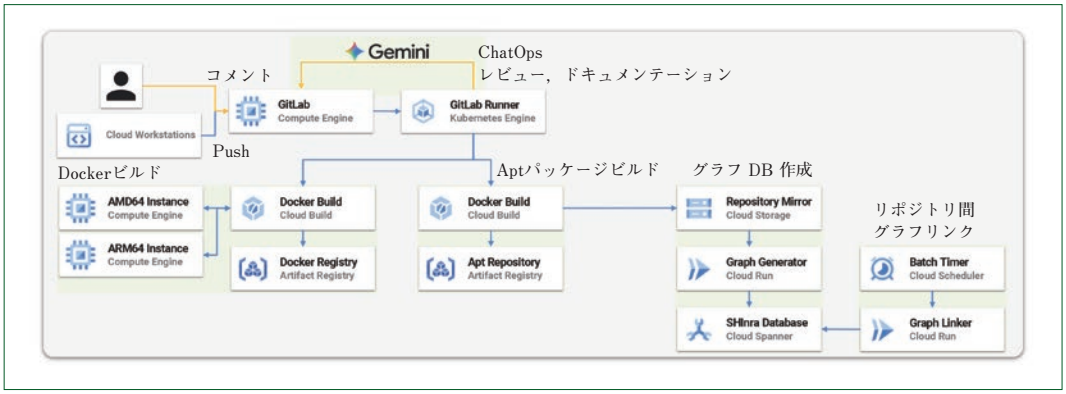

図4　CI/CD管理基盤
CI/CD Management platform



図5　ハードとソフトのデカップリング
Decoupling of hardware and software

CloudのIAP (Identity-Aware Proxy) およびIAM (Identity and Access Management) を適用し，不正アクセスの防止と適切な権限管理を両立した。**図2**に，Cloud Workstationsの利用例を示す。

Cloud Workstationsの導入によって，従来は数週間かかっていたオンプレミス環境における開発用ワークステーションの調達から情報システム部門でのセッティング，および個別の環境構築が不要になり，開発開始までの工数を削減できた。

3.1.2　エラー解決ナレッジデータベースによる知見の共有

本システムは，開発および実行時に生じるエラーログをキーとして，解決策の知見を蓄積・検索する機構である。**図3**に，処理フローを示す。開発中にエラーが発生した際，CAが自動的に本システム上でエラー内容のベクトル検索を行うことで，過去の類似事例およびその解決策を取得できる。また，解決されたエラー事例はCAによって自動でナレッジデータベースに追加される。

これらの機能によって，従来は個々の開発者や単一のCAのセッション内に閉じていたエラー解決の知見を，複数の開発者やその開発者が使用するCAに共有することが可能となった。また，CAが事前に学習していないエラーであってもナレッジデータベース上で解決方法を検索することができる。

3.1.3　コードとドキュメントのグラフデータベース

本システムは，社内の膨大なソースコードおよび関連ドキュメントの依存関係を，グラフデータベース(Cloud Spanner)上で構造化したグラフRAG (Retrieval-Augmented Generation)である。CAは，外部に公開されていない内製ライブラリの仕様を学習していない。そこで，本システムでは内製ライブラリのコードとドキュメントをグラフ構造化して，CAが内製ライブラリの仕様を検索可能とする。

情報検索においては，意味的な類似性を捉えるセマンティック検索と，関数やモジュール間の静的な依存関係を追跡するグラフ検索を複合的に組み合わせ，関連情報を的確に抽出するアプローチを採用した。

3.2　CI/CDによる開発プロセスの標準化

本層は，CI/CDにおける自動化ワークフローをテンプレートとして一元管理する基盤である。各開発者が個別にパイプラインを構築し記述する手間を省き，標準化されたテンプレートを再利用することで，設定工数の削減とビルド・検証品質の均一化を実現している。具体的には，コンテナイメージのビルドおよびデプロイ機能，ならびにコードレビューやドキュメント生成を担うAIによるサポートチャット機能を備える。

また，ロボット開発においては，さまざまなプロセッサアーキテクチャに対応したCIによるビルドプロセスを有している。

さらに，ドキュメンテーションに関しては生成AIを用いてソースコードから仕様書を自動生成する。これら一連の処理は，GitLab[*2]の更新履歴上のコメントからコマンド実行が可能であり，コードとドキュメントの乖離を防ぐ仕組みとなっている。**図4**に，CI/CD管理基盤の動作概要を示す。

3.3　デジタルツイン層およびデータ分析層の機能詳細

物理的な制約を持つ実機検証を省力化するべく，デジタルツイン層およびデータ分析層に次の機能を実装した。

3.3.1　SIL (Software-In-the-Loop) シミュレータの構築

本SILシミュレータは，車両の形状や機構の動作，センサ

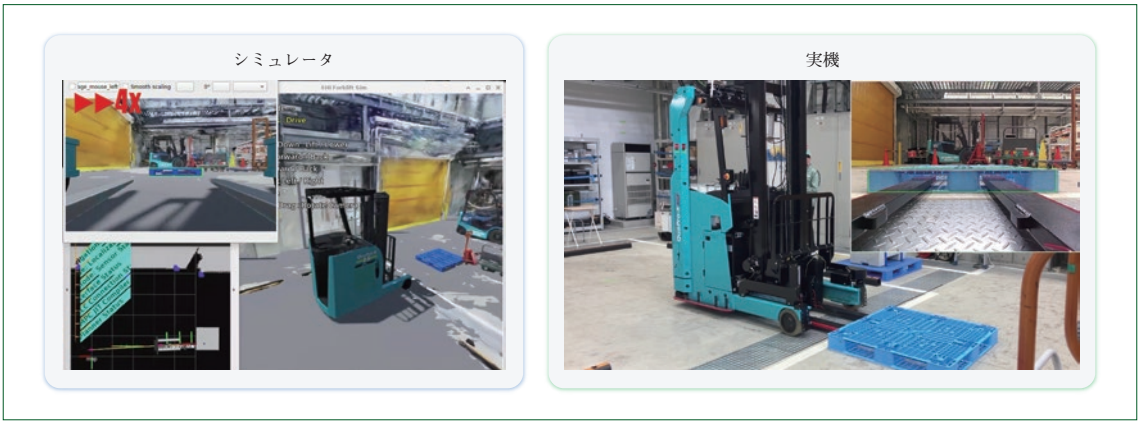

図6　シミュレータとAGF実機での実験
Experimental snapshots: simulator vs. real AGF

や車両運動モデル，動作環境の形状を3D仮想空間上に再現した。実ハードウエアと同一の通信プロトコル(UDP，HTTPなど)で仮想ハードウエアと通信するSIL構成を採用した。通常はシミュレータで検証したコードにハードウエアドライバなどを追加して実機検証する。しかし，本構成のシミュレータであれば，通信プロトコルが実機と同一であることから，ドライバを含むソースコードの机上検証が可能である。すなわち，机上検証で動作を確認済みの信頼性の高いコードを変更することなく実機にデプロイできる(**図5**)。

3.3.2　Google Cloud を活用した統合データ分析パイプライン

データ分析層は，シミュレータによる検証および実機走行試験のログデータを収集し，アルゴリズムの定量評価とバグの早期発見を可能にする。シミュレータや実機上で計測された車両の自己位置，速度ベクトル，アクチュエータの制御指令値などのログデータをクラウド上に保存し，可視化ツールを用いてブラウザ上でデータの確認と検証を可能とした。

4　AGF実機での実験

本開発基盤の効果検証をAGFで行った。具体的には，自己位置推定，ナビゲーション，荷役操作などの基本自律化機能と，単体機能として開発されていたパレット検出アルゴリズムとの統合を検証対象とした。検証シナリオは，指定されたパレット近傍まで移動したAGFがパレット検出アルゴリズムの認識結果に基づき，パレットに対して正対するようにアプローチ経路を生成し，フォークを挿入してパレットの運搬をするというものである。

事前にシミュレータを用いて，経路生成やフォーク動作のシーケンスに論理バグや不具合がないかを検証してから実機実験へ移行した。その結果，初回の実機実験において，パレットの自動検出からアプローチ，フォーク挿入，運搬にいたる一連の荷役機能を動作させることに成功し，実機実験中のバグ修正も抑制できた。**図6**に，シミュレータおよび実機を用いた実験環境を示す。

5　ロボット開発基盤の成果と今後の展開

5.1　3つの成果

1つ目は，開発プロセスの高速化である。CAの活用によりコード生成，デバッグおよびドキュメント自動生成にかかる時間を短縮した。これにより，開発者のアイデアを迅速にコード化でき，実装スピードが向上した。

2つ目は，実機開発における安全性と信頼性の両立である。物理的制約の大きい実機検証において，シミュレータ上での事前検証によって時間のかかる実機検証を減らした。

3つ目は，チーム開発の実現である。従来，開発環境が開発者ごとに異なっていたが，Cloud Workstationsによる開発環境の統一，GitLabによるソースコードの管理などによってチーム開発が可能となった。

5.2　今後の展開

今後は，本基盤による開発プロセスをさらに発展させ，製品ライフサイクル全体に拡張していく。具体的には，SDV実現に向けて，安全で耐障害性の高いOTA配信技術の開発を進めるほか，データの収集とその分析機能を発展させ顧客体験の向上につなげる。また，より高度な自律化機能開発を目的に，シミュレータをフィジカルAIのVLA(Vision-Language-Action)や，VLM (Vision-Language-Model)などに活用する。

6　むすび

(1) 生成AIとGoogle Cloudを用い，SDV実現に向けた次世代のロボット開発基盤を構築した。
(2) コーディングエージェントとCI/CDによって開発工程全体の効率化を実現した。
(3) シミュレータの活用によって新機能を安全に検証しつつ，実機検証の工数短縮を実現した。
(4) 大規模なロボットシステム開発において，チーム開発が可能な開発基盤を整備した。

＊1　Google Cloud（およびかかる資料で使用されるその他の Google Cloud 商標）は，Google LLCの商標
＊2　GITLAB is a trademark of GitLab Inc. in the United States and other countries and regions.

(参考文献)
(1)　経済産業省ウェブサイト,「モビリティDX戦略」を策定しました, 経済産業省, https://www.meti.go.jp/press/2024/05/20240524005/20240524005.html, 参照 2026年7月9日.