

エージェント協調のための次世代ワークフローエンジンの開発 —AI と人間の協働を実現する次世代業務自動化基盤—

1. 背景

GPTをはじめとする大規模言語モデル（LLM）の急速な発展に伴い、AI が自律的にタスクを完遂する「AI エージェント」への期待が高まっている。しかし現実には、実運用される AI エージェントの 7 割はプロンプトエンジニアリングのみで構成され、自律性は厳しく制限されている。デモ環境では動作する AI エージェントが本番環境では安定稼働しないという「理想と現実のギャップ」が深刻な課題であり、自律性を「制御可能なプロセス」に載せる Agentic ワークフローへの転換が求められている。

他方で、本番運用に耐える堅牢性と、AI エージェントに必要な動的制御の双方を満たすツールはいまだ存在しない。エンタープライズ向けワークフローエンジン（JP1、Airflow 等）は堅牢な運用基盤を備えるが、タスク構成が DAG 定義時に固定され AI エージェントの動的な意思決定や Human-in-the-Loop（HITL）に対応できない。一方、AI エージェント開発ツール（LangChain/LangGraph や Mastra 等）は柔軟なフロー制御が可能だが、運用監視・障害分離・分散スケールリングが未成熟で本番運用に耐えない。「開発のしやすさ」と「本番運用の堅牢性」を両立するツールが存在しないことが、AI エージェントの実業務導入を阻む根本的な障壁であり、本プロジェクトの出発点である。

2. 目的

本プロジェクトでは、AI エージェントの動的意思決定機能と従来型 ETL 処理の堅牢性、さらに Human-in-the-Loop（HITL）を高度に統合した次世代ワークフローエンジン「Graflow」を開発することを目的とした。直感的な Python DSL による優れた開発者体験（DevEx）と、動的タスク生成、スケーラブルな分散実行基盤、チェックポイント/リジューム、HITL 機能、細粒度エラー制御といった本番運用品質の機能群を一つのプロダクトに統合する。

設計アプローチとしては「責任分離」を基本方針とした。AI エージェント内部の推論・行動ロジックは特化したフレームワーク（Google ADK、PydanticAI 等）に委ね、Graflow はワークフロー全体のオーケストレーション——タスク間の依存関係管理、スケジューリング、障害復旧、状態管理——に専念する。これにより、急速に進化するエージェント技術に追従しつつ、ワークフロー基盤としての堅牢性と安定性を維持した構造を実現する。

3. 製品・サービスの内容

Graflow は、Python ベースの DSL を核とした次世代ワークフローエンジンである。「責任分離」の設計哲学に基づき、SuperAgent の実装は特化フレームワーク (Google ADK 等) に委譲し、Graflow はタスク間の依存関係管理、スケジューリング、障害復旧、状態管理といったオーケストレーションに専念する (Fat Node 設計)。DAG と状態機械を融合したハイブリッド設計により、演算子 (>>, |) による静的構造定義と `next_task()/next_iteration()` による動的遷移を同時にサポートする。

主要機能は以下の 9 つである。

- (1) **Dynamic Task Generation** : 実行時にノードとエッジの双方を動的に生成する機能。データ量や状況に応じた柔軟なタスク構成を実現する。
- (2) **Scalable Fleet Execution** : 4 層構成 (Orchestrator、Worker Fleets、Communication Channel、Checkpoint Store) の分散実行基盤。1 行のコード変更でローカルから Redis 分散実行に切替可能であり、Kubernetes HPA・AWS ECS Auto Scaling に対応する。
- (3) **Channel-Based Communication** : TypedChannel による型安全な分散状態共有。分散環境でも IDE の自動補完と型チェックが利用可能である。
- (4) **User-Controlled Checkpoints** : 任意時点での状態保存とリジューム。ストレージはローカル、S3、Redis から選択可能である。
- (5) **Human in the Loop** : 承認、テキスト入力、選択肢等のフィードバックをワークフロー内に組み込む機能。Slack などの Webhook 連携に対応する。チェックポイント機能と連携し、承認待ち中はリソースを自動解放する。
- (6) **Full Observability** : LangFuse・OpenTelemetry 統合によるエンドツーエンドのトレーシング。
- (7) **プラグ可能なタスクハンドラー** : プロセス内直接実行、Docker コンテナ隔離実行、Cloud Run/Lambda 等のリモート実行を選択可能。
- (8) **細粒度エラーポリシー** : Strict Mode、Best-effort、At-least-N、Critical Tasks の 4 種のビルトインポリシーで並列グループの障害制御を実現。
- (9) **LLM 統合** : LiteLLM 経由の統一 LLM 呼び出しと Google ADK 等の SuperAgent フレームワーク統合を提供。

動作環境は Python 3.10 以上、コンテナランタイム (Docker 等) である。Apache 2.0 ライセンスの OSS として GitHub 上で公開済みであり、プロダクトサイト (<https://graflow.ai/>) にてドキュメント、チュートリアル、API リファレンスを提供している。

主なアプリケーション例は以下のとおりである。

- (a) **商談準備レポートの自動生成** (図 1 参照) : ニュース・決算/IR 情報を並列収集しインサイト抽出と Critique を経てブリーフィング資料を出力。商談準備時間を 80%削減する。
- (b) **大規模文書の適応的分割処理** (図 2 参照) : RFP 回答作成 (セクション別に法務/技術/価格の Agent を動的生成し並列処理)、契約書レビュー (条項種別に応じたリスク検出 Agent を動的生成し、再帰処理で高リスク条項のみ深掘

り)、M&A デューデリジェンス (数百文書を並列処理し統合 DD レポートへ集約) 等に対応する。

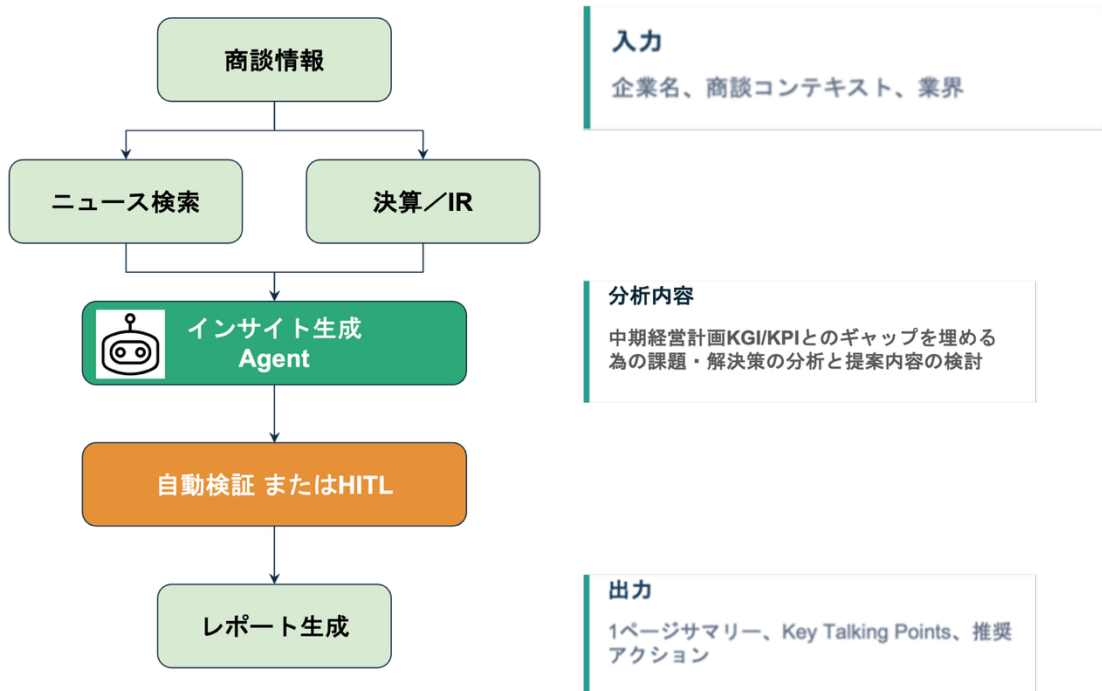


図 1 商談準備レポートの自動生成

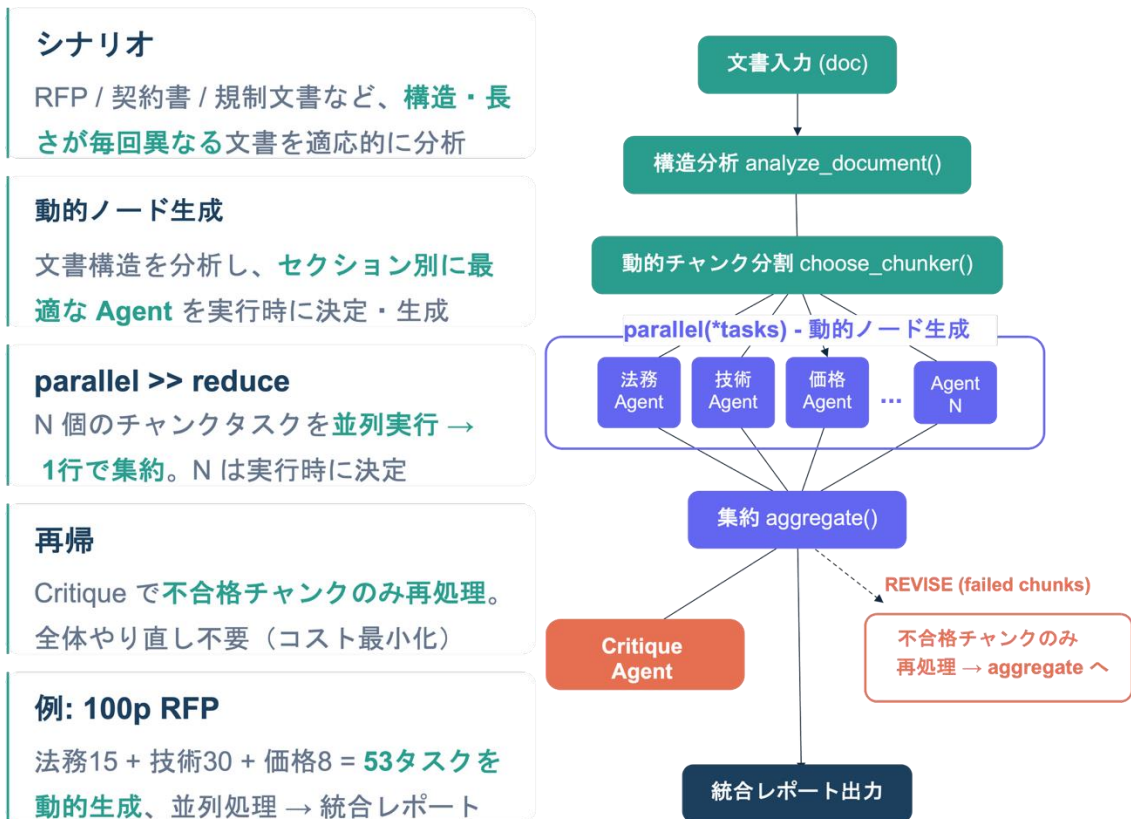


図 2 大規模文書の適応的分割処理の例

4. 新規性・優位性

Graflow の新規性は、従来型の堅牢なワークフロー管理機能と AI エージェントの動的処理、さらに HITL を単一プロダクトに統合した点にある。差別化の核心は以下の 3 点である。

第一に、動的タスク生成である。Airflow は DAG 定義時にタスク数が固定され、LangGraph はノードがコンパイル時に固定されエッジのみ動的である。Graflow ではノードとエッジの双方を実行時に動的に生成でき、例えば 100 ページの RFP をセクション別に 53 個の Agent タスクへ動的分割し並列処理した後に統合する処理が 1 行で記述できる。また、不合格チャンクのみを選択的に再処理する再帰パターンにより、LLM コストも大幅に削減される。

第二に、細粒度エラーポリシーである。既存ツールの並列実行は all-or-nothing に限定されがちだが、Graflow は Strict Mode、Best-effort、At-least-N、Critical Tasks の 4 種のポリシーでビジネス要件に即した障害制御を実現する。

第三に、チェックポイント×HITL の連携設計である。承認待ち発生時にワークフロー状態を自動保存し、承認後にチェックポイントから再開することで、数時間から数日の承認フローでもリソースを占有しない。この組み合わせは既存のワークフローエンジンやエージェントフレームワークには見られない。

加えて、分散環境での型安全 (TypedChannel)、ローカル/分散切替、直感的な Python DSL により、DevEx とエンタープライズ堅牢性を両立する独自の価値を提供する。

5. 事業普及（または活用）の見通し

2026 年 1 月に Apache 2.0 ライセンスで GitHub 上に OSS 公開し、プロダクトサイト (<https://graflow.ai/>) も開設済みである。開発者が評価・導入を開始できる状態にあり、ドキュメント、チュートリアル、API リファレンスを整備している。

事業展開としては、OSS コミュニティを起点としたボトムアップの信頼形成を第 1 段階とし、技術サポートを通じたユーザー拡大 (第 2 段階)、OSS コアに対する長期サポート・セキュリティパッチ等の付加価値サービスによる収益化 (第 3 段階) の 3 段階で進める。

6. 期待される波及効果

Graflow により、以下の波及効果が期待される。

- (1) 業務自動化市場において、AI エージェントと堅牢な運用基盤を統合した新たなカテゴリを確立し、PoC から本番運用への移行障壁を低減することでエンタープライズにおける AI 活用を加速させる。

- (2) Apache 2.0 ライセンスでの OSS 公開により、開発者コミュニティの形成と技術的知見の共有を促進し、エコシステムの成長を通じてプロダクトの発展を加速する。
- (3) 直感的な Python DSL により非エンジニアでもワークフロー定義が可能になり、DX 推進における人材不足の課題解決に貢献する。
- (4) HITL の仕組みにより、完全自動化が困難な判断プロセスにおいても AI 活用の裾野を広げる。

7. イノベータ名（所属）

油井 誠（トレジャーデータ株式会社）

渡辺 博之（株式会社 AI Management Lab）

（参考）

- ・ Grafflow プロダクトサイト: <https://grafflow.ai/>
- ・ GitHub: <https://github.com/GraflowAI/grafflow>
- ・ 技術ブログ（Zenn）: <https://zenn.dev/myui/articles/64f31faaf1488c>
- ・ Grafflow 公式 YouTube チャンネル: <https://www.youtube.com/@grafflow-ch>