

大規模言語モデルによる レビュー/フィードバック業務の代替について

NTTデータグループ 川上 蒼太
キャノンイメージングシステムズ 高頭 和輝

日立製作所 西嶋 駿
富士通 前田 翔

開発における問題点

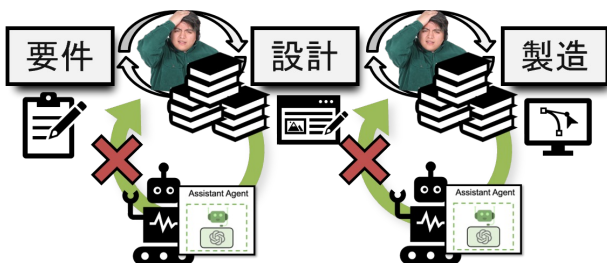
ソフトウェア開発の現場ではフィードバックループが推奨されるが、現場では業務が逼迫し実施できていない。原因は、自動化で業務を代替しようにも、**自然言語やプログラミング言語など複数の言語処理が必要**で、従来技術では実現が困難なためである。

手法・ツールの適用による解決

Large Language Model (LLM)は言語処理能力に優れており、自然言語やプログラミング言語を理解し、人間同様の文章やソースコードを生成可能である。
我々は、自然言語を多く使用するソフトウェア開発においてもLLMが有効と考え、**レビュー/フィードバック業務をLLMで代替し、現場の逼迫の解決を目指す。**

アプローチ

タスクが複雑、フォーマットが異なる
⇒単独のLLMによる代替は困難。



情報名	フォーマット
要件定義書	・自然言語(NL)
設計書	・自然言語 ・モデリング言語(ML)
ソースコード	・プログラム言語(PL)

NL: Natural Language, ML: Modeling Language,
PL: Programming Language

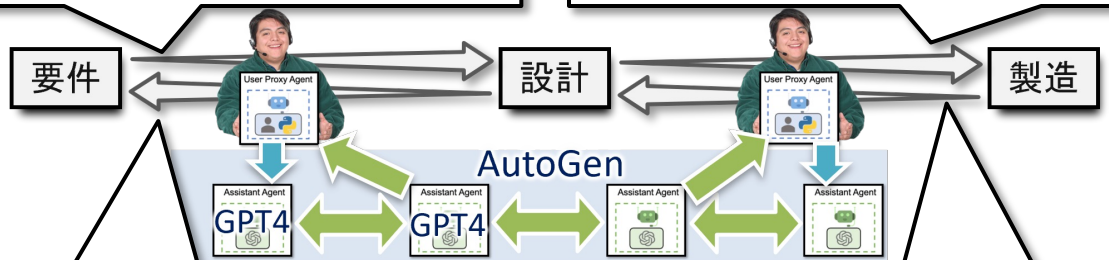
- ・複数のLLMを協調動作させるMulti Agent Frameworkを採用。
- ・各工程毎のケーススタディを実施し、システム開発全体を俯瞰。

要件→設計のレビュー

対象: 電池の充電率演算システム
入力: 要件定義書(NL)
出力: レビュー(NL)

設計→製造のレビュー

対象: Snake game(MetaGPTで生成)
入力: シーケンス(ML)、ソース(PL)
出力: レビュー(NL)



設計→要件のフィードバック

対象: Test Requirements to Models
入力: 要件定義書(NL)、設計(ML)
出力: フィードバック(NL)

製造→設計のフィードバック

対象: GraphQL Library(Ariadne)
入力: シーケンス(ML)、実装(PL)
出力: フィードバック(NL)

評価方法: 意図的に入力情報に不備を与え、LLMによる指摘率を評価。

結果・考察

LLMによるレビュー/フィードバック業務の代替が可能である見通し。

業務	開発工程	指摘率	判定	詳細
Rev.	要件→設計	21/23	○	網羅性◎。誤記(ドメイン知識)は指摘困難(0/2)。
	設計→製造	24/27	○	複数回のRev.で全バグ指摘可能。Agentのロール設定に課題。
F.B.	設計→要件	3/12	△	複数Agentが同一出力となり、低指摘率。
	製造→設計	15/19	○	網羅性◎。異常系パスは低指摘率。

解釈: 出力や工程の抽象度が高い(F.B. > Rev.、要件 > 設計 > 製造)ほど、
プロンプトによるLLMと人間の認識合わせが困難。⇒指摘率低下の原因と推測。

今後の課題と対策

業務への展開が望まれるが、残課題が存在。

- ①汎用性・再現性
⇒適用事例を増やし、適用範囲をノウハウ・ナレッジとして蓄積
- ②Rev./F.B.観点の与え方
⇒マニュアル整備による属人性解消
- ③環境設定と利用へのハードル
⇒チャット形式やCI連携によるUI改善