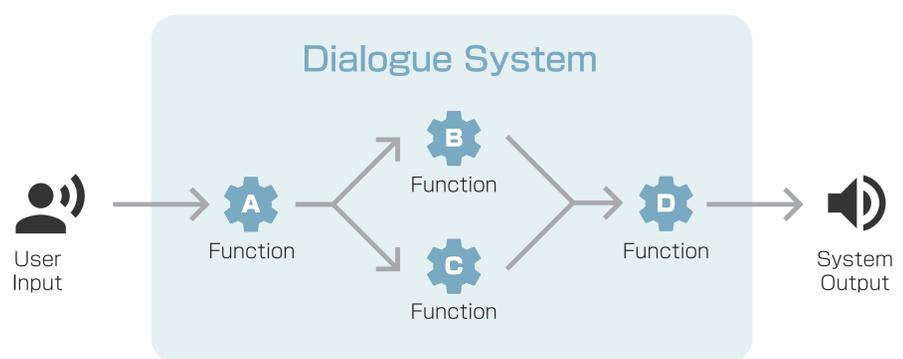


リアクティブ対話システム

柔軟に振る舞いを変え継続的に機能を改善する対話システム

研究の概要

リアクティブ対話システムは、構成する各機能の結合度を下げ、より柔軟に振る舞いを変更できる対話システムを実現するためのアーキテクチャーです。株式会社 Nextremer ではこれまでいくつかの対話システムを開発・運用しており、サービスの実運用においてユーザー満足度を向上させることが重要で、そのためにはシステムへの細かな変更が多く発生することが明らかになりました。本アーキテクチャーにて実現されるリアクティブ対話システムでは、柔軟な変更を可能にするために、構成する各機能を独立した環境で動作する1つの関数として扱い、それらの関数を合成することで対話システムを構成します。これにより、対話システムの動作中でも容易にシステムの変更が可能となり継続的な改善を可能にします。

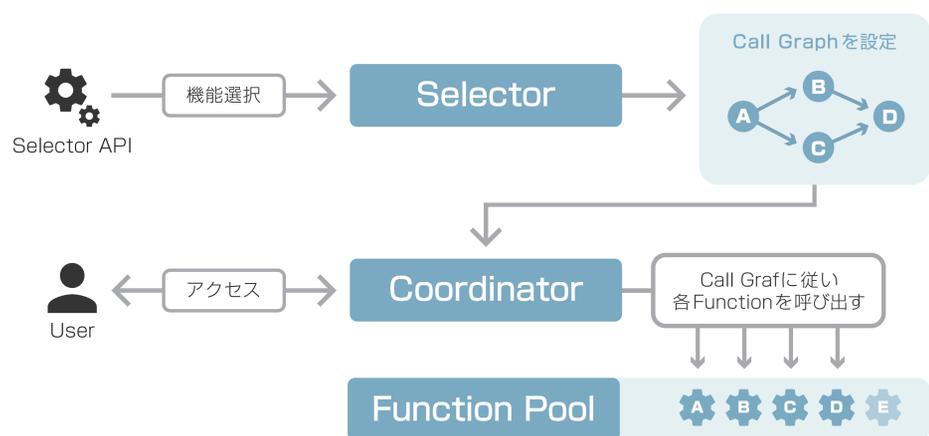


$$Y = f_d(f_b(f_a(x)) + f_c(f_a(x)))$$

合成関数のように複数の機能を組み合わせた対話システム構成

システムアーキテクチャー

リアクティブ対話システムは大規模 Web サービスで広く利用されているマイクロサービス、関数型プログラミングに影響を受けて設計されています。リアクティブ対話システムを構成する要素は主に Function, Selector, Coordinator があります。Function は1つの独立した機能の単位を表し、言語理解、対話状態管理、対話知識検索、行動選択、応答生成などが含まれます。Selector は実行する Function とそれらの実行順序を定義した Call Graph を設定します。Coordinator は Selector によって設定された Call Graph に沿って各 Function を呼び出し対話システム全体を実行します。



本技術を応用した対話システムの展望

対話システムが持つ種類が増えるほど対話システムが対応できる要求の幅が広がるため Function の種類を増やすことでそれを実現できます。Function の開発者はクローズドなグループの開発者だけでなく、一般の開発者が自由に開発・公開できるプラットフォームへの発展も考えることができます。また、Call Graph の設定を自動化することで、対話のドメインや状況が変わる度に無数にある Function の中からユーザーからの要求や状況に応じて適切なものを選択し、状況に応じて振る舞いを変えるリアクティブ対話システムの実現が可能となります。

