

複数チャットボットの連携による業務支援フレームワークの提案

Proposal of a business assisting framework with multiple chatbots

角田 啓介 *¹ 岡崎 優尋 *¹ 小柳 隆人 *¹ 長森 藤江 *¹ 箕浦 大祐 *¹
 Keisuke TSUNODA Masahiro OKAZAKI Ryuto KOYANAGI Fujie NAGAMORI Daisuke MINOURA

*¹NTT コムウェア株式会社
 NTT Comware Corporation

This paper introduces a business assisting framework with multiple chatbots. In enterprises, business activities are processed with different employees in multiple departments. Most of existing approaches for business assistance are classified into (1)an application of individual activity support and (2)large business suite for assisting business activities processed with different employees in multiple departments. However, to improve business activity more effectively, it is important for a business assisting approach to be applied to business activities processed with not only an employee, but different employees in multiple department, and to be easier to use for each employee and flexible for changing business process in each department. In this paper, we propose a new business assisting framework with multiple chatbots. In the framework, each department designs and implements the chatbot which assists its business process as approach (1). Then, these chatbots are connected with each other using shared database for assisting business activities processed with different workers in multiple departments as approach (2). We implemented a prototype of proposed framework applied to service order operation and discuss the effectiveness and issue of our proposal.

1. 緒言

ビジネス環境の競争激化や先進国における労働人口減少に伴い、多くの企業で業務、特に定型的業務の効率化が求められている。従来、ビジネスにおける業務効率化のアプローチは、(1)各従業員が自身の業務に合わせてツールを導入し、自分の業務を効率化する(2)全社的に業務プロセスを刷新した上でそれに合わせたシステムを全社に導入する、のいずれかが主であった。しかしながら、(1)のアプローチでは従業員個人の範囲しか業務を効率化できない点、(2)のアプローチでは企業における経営陣や管理者がトップダウンで導入したシステムが各部署や従業員の実態にあっておらず業務効率化に寄与できない可能性がある点 [Buehrer 05] が、それぞれ課題であった。そのため、企業における業務を効果的に支援するためには、複数の従業員や部署を横断する業務を、各組織や従業員の実態へ柔軟に対応できる手法で支援する必要があると考えられる。

本稿では上記の課題を踏まえ、ある業務を担当するグループを部署と定義した上で、部署ごとの業務を対話的に支援するチャットボットを導入した上で、各チャットボットを疎結合に連携させることで、柔軟かつ効果的に業務を支援できるフレームワークを提案する。そして業務の具体例として、お客様があるサービスに申し込んだ時に発生するサービス申込対応業務を取り上げ、そのような業務プロセスを支援するプロトタイプを実装することで、提案フレームワークの効果と今後の展望について述べる。

2. 関連技術

2.1 従業員個人の業務支援

モバイルデバイスの普及に伴い、場所を問わずに従業員個人の業務を支援するアプリケーションやシステムが多数提案されてきた。その中でも特に近年、チャットボットを用いたシ

ステムが多数提案されており、例としてメッセージングアプリを用いたスケジュール管理支援システム [Comware 17]、ソフトウェア開発チームの支援システム [Storey 16] が挙げられる。チャットボットを用いることで、ユーザは普段使い慣れたモバイルデバイスのメッセージングアプリ上で、チャットボットの発言に基づいた対話的操作により、そのバックエンドに接続された複数のシステムへアクセスできるようになる。その結果、ユーザは場所を問わず簡単に、システムを用いる業務を実施できるメリットがある [Tsunoda 17]。一方、チャットボットのふるまいは業務プロセスに合わせてシナリオという形で設定しておく必要があるが、多数の部署や従業員が関連する業務を対象とすると大規模化かつ複雑化してしまうため、一般的には従業員個人やある特定のチーム内での業務支援といった限られた範囲で利用される。

2.2 業務フロー全体の支援

業務効率を大きく向上させるためには、従業員個人やある部署だけでなく、複数の従業員や部署を横断する業務フロー全体を対象とし、それを最適化させると共に、最適化されたフローに合った大規模なシステムを導入することが最も根本的な解決策とされる。このような活動は“IT Transformation”と呼ばれる [Deloitte 16]。一方で、“IT Transformation”は準備や導入に多大な時間と費用がかかる上、新たなシステムが必ずしも各部署や個人の業務実態と合致しないため、有効に活用されない事例があるとされる [Buehrer 05]。

他方、2000年代後半から、業務志向でより柔軟なシステム構築の考え方として Service Oriented Architecture(SoA) [Laskey 09] が提唱されている。SoAでは業務フロー全体を対象とするのではなく、ある単位(例えば部署毎)の業務フローのみを対象とし、それを支援する一つのソフトウェアモジュールを作成する。そして各モジュール間で共通なインタフェースを定義し、接続性を確保することで、複数のモジュールを組み合わせによる全社的な業務支援システムを構築するという考え方である。SoAのメリットとして、モジュール間インタフェースが共通化されているため、あるモジュールで支援する

業務プロセスに変更がある場合でも柔軟に対応できる点が挙げられる。

3. 提案フレームワーク

本稿では、効果的な業務支援を目指して複数部署を横断する業務を対象としつつも、従業員各個人や各部署にとって利用しやすく、かつそれらの業務実態や業務フロー変更に対応できる業務支援フレームワークを提案する。提案フレームワークの概要を図1に示す。本フレームワークでは、ある一業務を担う単位を部署と定義し、部署毎にその業務を支援するチャットボットが導入される。各チャットボットは、当該部署における業務プロセスに合わせたシナリオスクリプト、そのスクリプトを実行させるスクリプト実行エンジン、ボットとユーザの間に必要となるメッセージングアプリ等のUIと接続するUI接続機能を持つ。その上で、複数導入されたチャットボットは共有DBを用いて疎結合に接続されている。提案フレームワークにより、各従業員個人や各部署の業務実態やその変化に合わせた対話的かつ柔軟な業務支援と、複数部署を横断する効果的な業務支援を両立することができる。

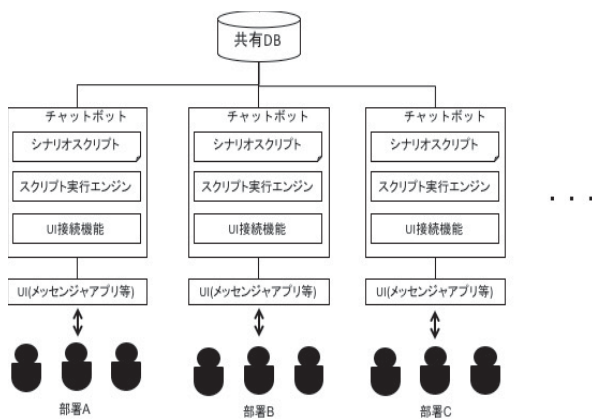


図1: 提案フレームワーク

4. プロトタイプの設計と実装

4.1 対象とする業務プロセス例

プロトタイプ実装の対象として、本稿ではお客様対応を行うフロントエンドの部署(以下、フロント部署)と事務処理等を行うバックエンド対応の部署(以下、バックエンド部署)が連携したサービス申込対応業務を取り上げる。対象とする業務では、フロント部署がお客様から申込書を受領して申込書を確認の上、バックエンド部署へ申込書を送付する。バックエンド部署は申込書を元に申込内容の確認とシステム操作を実施した上で、結果をフロント対応の部署へ通知することで、お客様に申込完了、または不可を伝える。さらに管理者はこのような業務の状況を随時確認し、業務の停滞やトラブル等を発見したら随時サポートを行う。対象となる業務フローを図2に示す。

4.2 設計と実装

図3にプロトタイプの構成図を示す。プロトタイプでは3つのチャットボット(フロントボット、バックエンドボット、管理ボット)が実装され、それぞれの部署の業務に合わせたシナリオを持つ。シナリオはスクリプト言語で記述されているため、簡易に変更が可能である。また、各チャットボットは共有DB

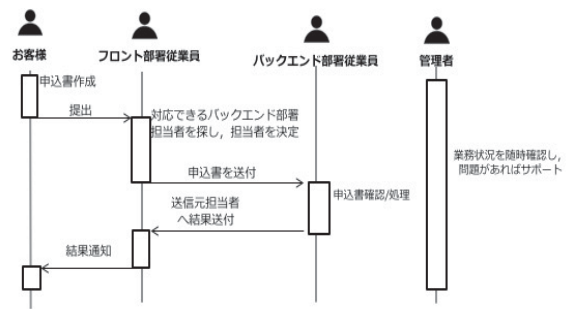


図2: 対象となる業務フロー

を介して相互に通信を行うため、共有DBにおけるデータ操作方法さえ定義しておけば、それ以外の各チャットボットのシナリオは自由に変更可能である。なお、プロトタイプではフロントボット及びバックエンドボット用のUIとしてNTTコムウェアのシャイン@TALK [Comware 18]を、管理ボット用のUIとしてWebブラウザを用いた。

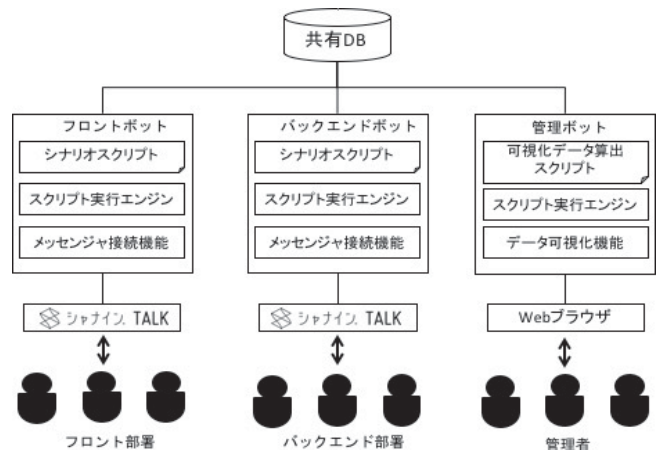


図3: プロトタイプの構成

図4にプロトタイプ導入後の業務フローを示す。プロトタイプにおいて、フロント部署の従業員はお客様から申込書を受領し、項目の漏れ抜けなどを確認後、申込書画像を撮影の上フロントボットに送信し、その後スタンプを送信する。このスタンプが、フロントボットのシナリオで定義された業務開始コマンドとなっている。フロントボットは申込書画像データを受信し、シナリオで定義された処理、例えば画像品質やサイズ、記述内容の確認などを実施後、問題がなければ共有DBの指定されたフィールドに申込書画像とその受領情報を書き込む。一方、バックエンドボットは常に共有DBの指定されたフィールドを監視しており、もし新たな申込書画像と受領情報があった場合はそれらを取得する。その後、複数存在するバックエンド部署の従業員のうち、現在業務を実施していない従業員を1名選定した上でバックエンドボットは当該従業員を新たなチャットルームに招待する。そして共有DBより取得した情報を送信し、申込書画像を元に業務を実施するように依頼する。当該従業員は送信された情報とバックエンドボットに記述されたシナリオに基づいて業務を実施する。具体的には申込書画像の確認と、その内容に基づいた他システムへの情報入力や書類作成などである。業務終了後、当該従業員が情報(申込の完了/未了やその理由)とコマンドとなるスタンプをバックエンドボット

トに入力すれば、その情報は共有 DB を通じてフロントボットに送信され、フロントボットは送信元のフロント部署従業員にその旨を通知する。一方、管理者は随時、管理ボットが共有 DB から随時取得したデータより、可視化データ算出スクリプトに沿って算出・可視化した業務状況データをブラウザを通じて確認する。管理者はもし所要時間が長い申込などを発見した際は、当該従業員へのサポートを実施する。

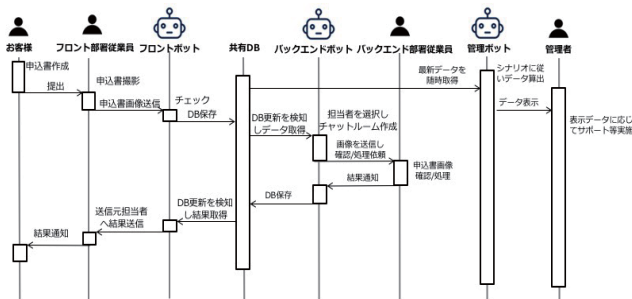


図 4: プロトタイプ導入後の業務フロー

プロトタイプとして実装したフロントボットとフロント部署従業員とのチャットルームの例を図 5 に、バックエンドボットとバックエンド部署従業員とのチャットルームの例を図 6 にそれぞれ示す。図 5 に示したように、フロント部署従業員は申込書を撮影後、フロントボットに画像と「依頼」というスタンプを送信することで、申込書の確認・処理依頼がバックエンド部署へ送られる。また図 6 に示したように、バックエンドボットからチャットルームに招待されたバックエンド部署従業員は画像を受け取り後、「確認中」というスタンプを送信することで、確認・処理中であることを送信でき、「完了」スタンプを送信すること確認・処理の完了をフロント部署へ通知できる。このように画像の撮影・送信とスタンプ送信を中心とすることで、従業員は最小限の入力操作で簡易に業務を遂行できる。

管理者ボットの画面例を図 7 にそれぞれ示す。図 7 における各行は各申込の処理状況を表わしており、フロント部署、バックエンド部署それぞれの従業員名、業務状態(開始, バックエンド部署担当未割当, 実行中, 完了), 開始からの経過時間がそれぞれ表示される。管理者はこの画面を元に、所要時間が長い申込があった際の従業員へのサポートや、1 日の全データに基づく平均所要時間の把握及び改善策の策定などが実施できるようになる。なお、本プロトタイプにおける管理ボットは必ずしもキャラクタとの対話形式にする必要性はないと考え、ブラウザを UI として単に結果が随時可視化される形態としているが、例えば通知機能などが必要な場合は、メッセージングアプリに接続し、他のボットと同様にキャラクタとの対話形式にすることも可能である。

5. 考察

図 5, 図 6 で示したように、提案フレームワークに基づいたプロトタイプにより、各従業員はメッセージングアプリ上における簡易な入出力のみで業務を実施することができようになる。また、図 7 で示したように、管理者向け画面ではフロントボットからの各申込の処理状況や開始からの経過時間などが一覧表示されており、逐次更新されるため、管理者は現状の業務状況の把握やボトルネックの把握、ある期間のデータを分析することによる業務改善の検討が簡易に実施できるようになる。さらに重要な点として、各部署の業務を支援するこれらのチャ



図 5: フロントボットの画面例

ットボットは共有 DB のみを介して連携しているため、各部署の状況、例えば申込書の確認ステップの追加、従業員の変更や増減、管理 KPI の変更が生じた場合でも、自部署のチャットボットのシナリオスクリプトのみを変更すればよいため、柔軟に対応しやすいというメリットを実現できた点が挙げられる。

一方で、共有 DB の設計が全体に影響を及ぼすため、共有 DB のスキーマなどが根本的に変わるような場合は必ずしも変化に柔軟に対応できない可能性がある。しかしながら、提案フレームワークにおいてはチャットボットの追加や削除も当初より想定されるため、最初から共有 DB のスキーマ等を厳密に定め、全部署に一齐導入するのではなく、まずは一部の部署のみにチャットボットを導入して運用し、特に共有 DB などに關する課題を抽出の上、課題を解決して随時導入部署を拡大するといった、スモールスタートによってある程度問題を軽減できると考えられる。

6. まとめと今後の展開

本稿では、企業における業務支援を効果的に実施するため、複数部署を横断した業務を効率化を目的とした業務支援フレームワークを提案した。具体的には、部署ごとの業務を対話的に支援するチャットボットを導入した上で、各チャットボットを疎結合に連携させることで、柔軟かつ効果的に業務を支援できるフレームワークを提案した。そして、申込書に基づくサービス申込対応業務を対象にプロトタイプを実装し、その有効性と課題について述べた。

今後の展開としてはまず、提案フレームワークを企業における実業務に適用した際の効果を定量的・定性的に評価することが挙げられる。また、4 つ以上の部署が連携するような複雑な業務への適用と、そこから生じる課題の抽出と解決が挙げられる。さらに、提案フレームワークに基づく業務支援システムが導入されると、共有 DB をはじめとするシステムのログデータより業務全体のフローが把握できるようになるため、例えば並列処理化できるプロセスの発見など、業務プロセスの改善案が自動的に抽出できる可能性がある。そのため、対象業務の拡大と導入と合わせ、上記のような業務プロセス改善案の自動抽

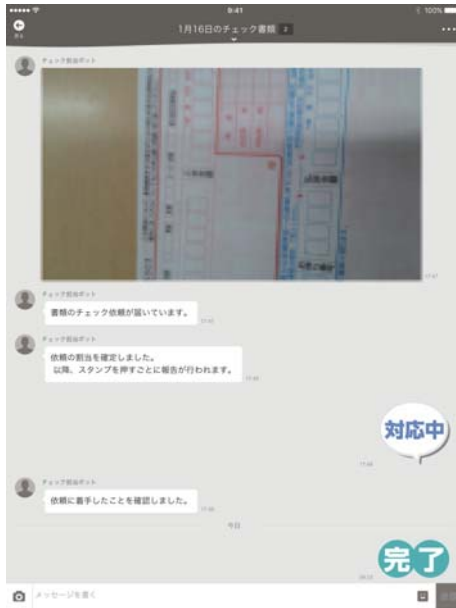


図 6: フロントボットの画面例

業務状況

id	業務	開始時刻	受付者	担当者	状態	経過時間
257	書類チェックタスク(ファイル添付)	2018/01/12 16:04	受付 太郎	確認 太郎	新規	0:00:40
260	書類チェックタスク(ファイル添付)	2018/01/12 16:04	受付 太郎	確認 花子	進行中	0:00:38
265	書類チェックタスク(ファイル添付)	2018/01/12 16:04	受付 太郎		未開始	0:01:17
254	書類チェックタスク(ファイル添付)	2018/01/12 15:50	受付 太郎	確認 太郎	終了	0:15:23
253	書類チェックタスク(ファイル添付)	2018/01/12 15:49	受付 太郎	確認 花子	終了	0:15:40
252	書類チェックタスク(ファイル添付)	2018/01/12 15:46	受付 太郎	確認 花子	終了	0:18:44

図 7: 管理ボットの画面例

出手法についても検討を進める予定である。

参考文献

- [Storey 16] Storey M, and Zagalsky A.: Disrupting Developer Productivity One Bot at a Time, *Proc. the 24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, pp.928–931 (2016).
- [Comware 17] NTT コムウェア: スケジューラ AI, <https://www.nttcom.co.jp/comware.plus/solution/201607.2.html>(2018.3.1. 閲覧).
- [Buehrer 05] Buehrer, R., Senecal S. and Pullins E., Sales force technology usage - Reasons, barriers, and support: An exploratory investigation. *Industrial Marketing Management*, Vol.34, pp.389-389(2005).
- [Salesforce 17] Salesforce.com, <https://www.salesforce.com/> (2018.3.1. 閲覧).
- [Tsunoda 17] 角田 啓介, 後藤 充裕, 北野 孝俊, 中村 浩司, 箕浦 大祐, チャットボットを用いたリモートワーク支援手法

の提案, 情報処理学会研究報告 グループウェアとネットワークサービス, 2017-GN-100(38)(2017).

- [Deloitte 16] Deloitte: Next generation automation - Transform your business processes with robotic and intelligent automation, <https://www2.deloitte.com/content/dam/Deloitte/ch/Documents/technology/ch-en-technology-next-generation-automation.pdf> (2018.3.1. 閲覧).
- [Laskey 09] Laskey K. B. and Laskey K., Service oriented architecture, *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol.1, Issue 1, pp.101-105(2009).
- [Comware 18] NTT コムウェア, シャナイン TALK, <https://www.nttcom.co.jp/shanaing/> (2018.3.2. 閲覧)