

チャットボットを用いた社内情報サービス管理方式 Method of Intranet Information Service Management using Chatbot

木下 順史[†] 薦田 憲久[‡] 藤原 融[‡]
Junji Kinoshita Norihisa Komoda Toru Fujiwara

1. はじめに

企業や組織においてパブリッククラウドやプライベートクラウドの活用が進展しており、IaaS (Infrastructure as a Service) や PaaS (Platform as a Service), SaaS (Software as a Service) といった様々な形態の情報サービスを社内サービスとして活用し業務を行っている。しかし、これら社内の情報サービスの管理を人手で行うのは煩雑であるという問題があった。

既存の解決手法としてセルフサービスポータルによる社内情報サービス管理が行われているが、GUI (Graphical User Interface) の開発はコストがかかり、予算や人員が限られた社内情報サービスにおいては GUI を改修し続けることが困難という課題があった。

このため、パブリッククラウドにおいてはチャットボットを用いた様々なサービスとの連携[1][2]や問い合わせ対応への適用[3][4][5][6]が行われており、社内情報サービスにおいてもチャットボットの活用が試みられてきた[7]。しかし、社内情報サービス管理への適切な適用方法や有効性は明らかではない。

そこで本論文では、チャットボットを用いた社内情報サービス管理方式について検討を行い、実環境に適用して効果や影響を測定することで有効性を検証する。

本論文の以降の構成は次の通りである。2 章で社内情報サービスの現状と課題を説明し、3 章で提案方式を説明する。4 章で実環境への適用を踏まえて効果や影響を確認する。最後に 5 章で本研究のまとめを述べる。

2. 社内情報サービスの現状と課題

2.1 社内情報サービスの現状

企業や組織においては、顧客管理やプロジェクト管理、研究開発等々、多種多様な業務が存在し、それらを支えるために、社外のパブリッククラウドや社内に構築されたプライベートクラウド等、様々なサービスを社内情報サービスとして利用者である従業員に提供している。セキュリティや内部統制に対応すべく、社内の IT (Information Technology) 部門が中心になってそれら社内情報サービスを管理している。社内情報サービスの管理は手作業で行われる場合が多く、さらにサービスによって窓口や利用開始までの手続きが異なる等、IT 部門と利用者の双方にとって煩雑であった。

これらの問題を解決するために、セルフサービスポータルを設けることによるサービス窓口の一本化や、利用開始までの手続きの自動化が試みられてきた。しかし、ポータルの GUI 開発には工数がかかる。また、サービスの変更や仕様変更に伴って GUI を改修し続ける必要がある。IT 部門はコスト部門であるため、人員や予算が限られており、

GUI を改修し続けるのは困難であった。

一方、パブリッククラウドサービスにおいては、チャットボット (あるいはチャットコマンド) による様々なサービスとの連携が進展している。チャット上でチャットボットに指示することで、連携サービスからの通知を受ける、あるいは連携サービスの機能を活用することが出来る。チャットボットを用いることで、新しく専用の GUI を開発することなく、テキストや音声等を用いた直感的なインタフェースを通して、利用者が日常的に利用しているコミュニケーションツールを活用し続けることが出来る。

社内情報サービスにおいても、利用者からの問い合わせ対応等のヘルプデスクにチャットボットを利用する試みが行われている。社内の定型業務において、利用者からの問い合わせに対する応答をチャットボットに行わせることで、対応工数を削減するという狙いがある。個々のサービスにおける問い合わせ対応の簡略化だけでなく、様々な社内情報サービスの管理にもチャットボットを活用し得る。

2.2 チャットボット利用時の課題

社内情報サービスとして使用するパブリッククラウドやプライベートクラウドのサービスは多種多様である。サービス毎にチャットボットを設けるとチャットボットが乱立し、利用者が混乱して問い合わせが増加するため、工数削減につながらない。また、サービス毎に個別にコミュニケーションツールとの連携方式を検討しなければならない。チャットボットの開発を支援するフレームワークやサービス、ソフトウェアは存在するが、利用者のテキストや音声の認識をサービス毎に個別に実装するのは煩雑であり、サービス提供者の負担になる。

これに対し、図 1 に示すように一つのチャットボットで複数の社内情報サービスの管理を行おうとすると、一つのチャットボットに多種多様なサービスとの連携機能を実装する必要があり、チャットボットが肥大化する。サービスによっては、利用開始時にアカウント作成だけでなく、社内ネットワークや社内システムとの接続等の複雑な準備を行う必要があり、連携機能が複雑なものがある。このため、サービスの追加や更新のたびに肥大化及び複雑化するチャットボットを改修しなければならず、手間が増えるばかりかミスが入り込む可能性も大きくなる。

さらにサービスによっては、テナントやプロジェクト等のグループの概念を有する。個々の利用者にはサービスを利用させるだけでなく、グループを設けてメンバシップを管理し、グループへの参加申請や承認を行う等、複数の利用者との連携を通じたグループ管理を行う必要もある。

[†](株)日立製作所 Hitachi Ltd.

[‡]大阪大学 Osaka University

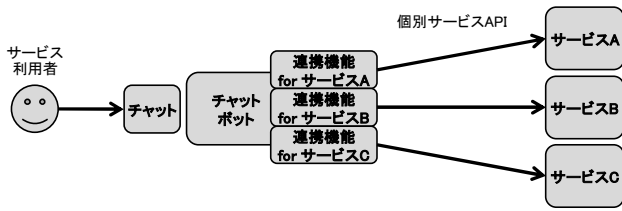


図 1 単一チャットボットによるサービス連携

3. サービスゲートウェイを用いた社内情報サービス管理方式

3.1 概要

本論文では、チャットボットの肥大化及び複雑化を解決するために、チャットボットとサービスの間にサービスゲートウェイを設けることで、機能の分離とインターフェースの共通化を行う。さらに、利用者との連携によるグループ管理に対応するために、チャットボットを介したグループ管理のワークフローを行う。

3.2 サービスゲートウェイ

図 2 に示すように個々のサービスとの連携機能をサービスゲートウェイとしてチャットボットから分離し、チャットボットと個々のサービスゲートウェイ間のインターフェースを共通化することで、チャットボットを単純化すると共に変更を不要とする。また、チャットボットあるいはチャットボットの関連サービスに利用者のテキストや音声の認識機能を実装することで、サービスゲートウェイでの当該機能の実装を不要とする。

チャットボットとサービスゲートウェイ間のインターフェースとして、サービス提供者がサービスの登録や削除を行うためのサービス登録インターフェースと、サービス利用者がサービスの利用開始や利用停止、その他のサービス機能の利用といったサービス利用操作を行うためのサービス利用インターフェースを設ける。

サービス登録インターフェースにおいては、サービス利用者にサービス名称とサービスゲートウェイのエンドポイント、当該エンドポイントにアクセスするためのクレデンシャルを登録させる。これにより、利用者がチャットボットからサービス名称の一覧を取得出来ると共に、利用者がチャットボットにサービス名称を付与してサービス利用操作を伝えることでチャットボットが当該サービスのサービスゲートウェイのエンドポイントにサービス利用操作を依頼することが出来る。

筆者らが運用する社内の研究開発環境において提供している個々のサービスの利用形態を調査した結果、チャットボットを用いて提供すべきサービス利用操作を共通操作と個別操作に大別出来ることが分かった。例を表 1 にまとめる。例えば Gitlab¹や Redmine²等、主に SaaS 形態のサービスにおいては、サービスの利用開始に伴うアカウント作成や利用停止に伴うアカウント削除といった共通操作のみを

チャットボット経由で提供すれば良く、利用者がグループ管理を含む残りの操作を SaaS のインターフェースで直接行うことが出来る。一方、Kubernetes³や OpenStack⁴等、主に PaaS/IaaS 形態のサービスにおいては、アカウントの作成や削除だけでなく、グループ管理に伴う様々な操作をチャットボット経由で提供する必要がある。これは、これらのサービスにおけるグループ管理が、社内のイントラネット接続ポリシーに合わせたテナントネットワークのイントラネットへの接続、社内のセキュリティポリシーに合わせたデフォルトでのアクセス制御設定等、環境に依存した事前準備を伴うためであり、これらの事前準備をチャットボット経由で行った上で、利用者に PaaS や IaaS のインターフェースを提供する必要がある。このような事前準備を、利用者により一部の管理権限を委譲することで実現することも可能であるが、利用者がグループ管理以上の作業を行う必要があり煩雑になってしまう。また、例えばベアメタルサービス等、主に API (Application Programming Interface) しか持たないような独自サービスにおいては、チャットボット経由で全ての操作を提供することで、個別の GUI 開発を回避出来る。以上のことから、サービス利用インターフェースにおいては、サービスの利用開始、利用停止、利用状況確認、その他の個別操作の 4 種類を設ける。

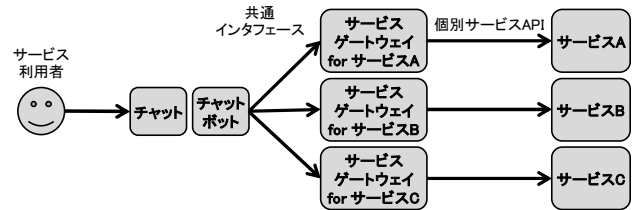


図 2 サービスゲートウェイ

表 1 サービス利用操作例

サービス	共通操作	個別操作
Gitlab, Redmine	<ul style="list-style-type: none"> ・利用開始 ・利用停止 ・利用状況確認 	無し
Kubernetes, OpenStack	<ul style="list-style-type: none"> ・利用開始 ・利用停止 ・利用状況確認 	<ul style="list-style-type: none"> ・プロジェクト参加 ・プロジェクト離脱 ・独自サービス公開 ・独自サービス公開解除 ・ネームスペース参加 ・ネームスペース離脱
ベアメタル	<ul style="list-style-type: none"> ・利用開始 ・利用停止 ・利用状況確認 	<ul style="list-style-type: none"> ・ゾーン作成 ・ゾーン削除 ・物理リソース追加 ・物理リソース削除 ・VLAN 追加 ・VLAN 削除

¹ Gitlab は、GitLab BV の米国およびその他の国における登録商標もしくは商標である。

² Redmine は、Jan Schulz-Hofen 及び Jean-Philippe Lang の商標または登録商標である。

³ Kubernetes は、The Linux Foundation の米国及びその他の国における登録商標又は商標である。

⁴ OpenStack は、米国とその他の国における Open Stack Foundation の登録商標または商標である。

サービス登録インタフェースと利用インタフェースを図3に示す。サービス毎に利用操作の内容が多種多様であっても、チャットボットとサービスゲートウェイ間のインタフェースを共通化し、個々のサービス固有の個別操作をサービスゲートウェイに委譲することが出来る。

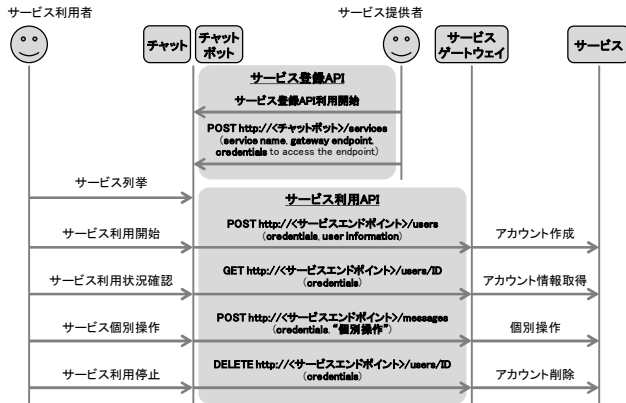


図3 サービス登録及び利用インタフェース

3.3 グループ管理ワークフロー

チャットボット経由でグループ管理を行うために、サービスゲートウェイにグループ管理機能を設け、サービスゲートウェイが管理者権限を用いて対象サービスに対するグループ作成や削除、メンバシップ管理を代行すると共に、チャットボットを介してメンバシップ管理のワークフローを利用者に提供する。

チャットボットは一般的に、チャットボットにメッセージを送った利用者へ応答するが、グループ管理においては、利用者からチャットボットへのグループ参加要求に対し、チャットボットが当該グループの管理者に当該要求に対する承認を確認し、当該承認をもって改めてチャットボットが要求元の利用者に結果を返す必要がある。グループの管理者が即時に承認するとは限らないため、チャットボットが要求元利用者への応答として結果を即時に返すことができないとは限らない。

そこで図4に示すように、利用者からの要求をチケットとして管理し、チケット内に要求元利用者のIDや要求を行ったチャットのチャンネルID、要求先グループの管理者のIDといった情報を格納する機能と、チケットの内容に応じて承認結果を要求元利用者へ発信する機能、及びそれらの機能を用いるインタフェースを、チャットボットがサービスゲートウェイに提供する。これにより、サービスゲートウェイがチャットボットを介して利用者間のグループ管理ワークフローを実現出来る。

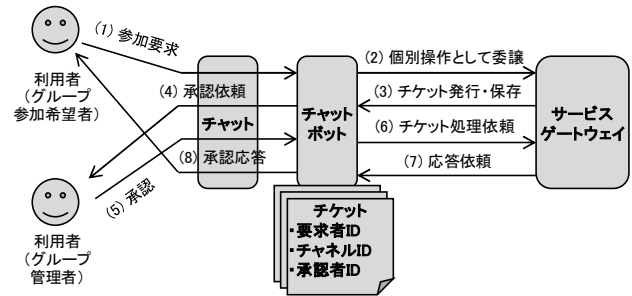


図4 グループ管理ワークフロー

4. 提案方式の評価

4.1 評価方法

提案方式の有効性を評価するために、筆者らが管理者として運用する社内の研究開発環境において、利用者にチャットボットを介した各種サービスの提供を行い、チャットボットの肥大化及び複雑化の回避度合いと、サービスゲートウェイやグループ管理による管理工数削減効果の評価する。チャットボットの肥大化及び複雑化の回避度合いとして、サービス追加によるチャットボットの改変有無を評価基準とする。また管理工数削減効果として、手作業対応工数を試算し、チャットボット導入後と比較する。評価環境において、2020年3月時点でのチャット及びチャットボットの提供状況は表2の通りである。チャットユーザ数は累計であり同時参加ユーザ数ではない。

表2 評価環境

期間	2016年4月～2020年3月
累計チャットユーザ数	約2800
ソフトウェア	チャット: Rocket.Chat ¹ チャットボット開発フレームワーク: Hubot ²

また、管理工数削減効果の評価の妥当性を確認すべく、チャットボットによるサービス利用状況への影響の有無を確認した。チャット及びチャットボットの提供開始前後も含めて最も長い期間提供していたサービスの新規登録者数（年合計及び月平均）の推移を図5に示す。当該サービスはチャット及びチャットボットの提供開始前である2015年9月から提供しており、チャットボットを提供するまでの間は利用者からの問い合わせに人手で対応していた。年合計の新規登録者数は年ごとに多少の増減はあるが、月平均の新規登録者数には大きな変動は無い。チャットボットがサービスの利用増加に影響しておらず、チャットボットによりサービス利用が阻害されているわけでもない事が分かる。チャットボットの有無に関わらずサービス操作が定期的に発生していたと考えることができる。

¹ Rocket.Chatは、Rocket.Chatの登録商標もしくは商標である。

² Hubotは、Github, Inc.の商標である。

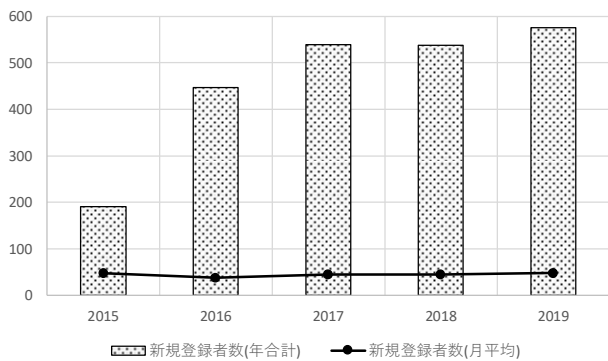


図5 サービスの新規登録者数事例

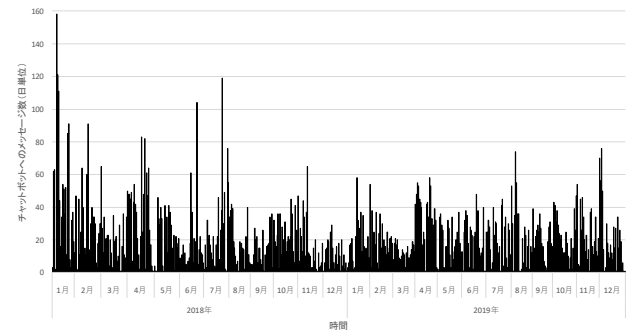


図6 チャットボットへのメッセージ数

4.2 評価結果

肥大化及び複雑化の回避度合い： 表2の期間中にチャットボットを介して提供された累計サービス数は約100であった。累計サービス数には、テスト用のサービスや、未公開で終わったサービス、サービスゲートウェイを用いていないサービスも含まれている。また、あくまで累計であり、同時に提供していたわけではない。様々なサービス提供が試行される中で、チャットボットの改修は、バグフィックスや開発フレームワークのバージョンアップ等を除いて、2017年10月以降一度も発生しなかった。サービスの追加の際には当該サービスのサービスゲートウェイのみ開発すればよく、提案方式が有効に機能した。

管理工数削減効果： チャットボットを介してサービス提供を行った期間のうち、2018年1月から2019年12月の2年間に利用者がチャットボットに送信したメッセージ数の推移を日毎に集計した結果を図6に示す。平均すると1日当たり24件程度であるが、変動が大きく、日によっては1日に100件、最も多い日だと1日に160件程度のメッセージがチャットボットに送信されたことが分かる。これらチャットボットに送信されたメッセージは何らかのサービスの登録やサービスの利用といったサービス操作を伴うものである。個々のサービスによって複雑さは異なるが、仮に一つのサービス操作に人手で対応するのに5分/件とすると、管理者は1日平均 $5 \times 24 = 120$ 分、日によっては $5 \times 100 = 500$ 分を利用者からのサービス操作要求に費やす必要がある。時にはサービス操作要求が全く無い日もあるため、これらの対応の人手で行うのは効率が悪い。チャットボットにより社内サービス管理を効率化できた。

また、チャットボットを介した個々のサービス操作の平均所要時間を測定した結果を表3に示す。SaaS型サービスにおけるサービス利用開始のような単純な共通操作については200~300msで完了するのにに対して、IaaS型サービスにおけるプロジェクト参加のように、利用者間でのグループ管理ワークフローを伴う個別操作については、ある利用者があるグループへの参加を申請してから、当該グループの所有者が承認するまでに平均して約2時間を要している。これをチャットボットではなく管理者が人手で仲介して行くと、グループの所有者への確認やフォローなど、1件あたり2時間かかる処理を管理しなければならず煩雑である。チャットボットを介して利用者間で申請・承認を実施してもらうことでサービス管理者の対応を削減できる。

表3 サービス操作の平均所要時間

サービス名	平均完了時間 (ms)
Gitlab	サービス利用開始： 243
Redmine	サービス利用開始： 213
OpenStack	プロジェクト参加・承認： 7535438

4.3 結果の考察

チャットボットへのメッセージ数は期間中に減少することではなく、常に発生し続けていることから、チャットボットを介した社内サービス提供は利用者を受け入れられている。試験中のもも含めてサービス登録も多数行われており、サービスを提供する場として活用されている。

5. おわりに

期間中にチャットボットがエラー等でダウンし、利用者がサービスの利用操作が出来なくなる事が何度かあった。チャットボットが社内サービスの単一障害点になるため、チャットボット自体の高信頼化を今後検討する必要がある。チャットボットを介して提供されるサービスの妥当性評価や審査についても今後の検討課題である。

参考文献

- [1] “Create a bot for your workspace”, <https://slack.com/help/articles/115005265703-Create-a-bot-for-your-workspace>, last visited 2020-04-14.
- [2] “How chatbots could change customer service over the next 5 years”, <https://www.businessinsider.com/how-chatbots-could-change-customer-service-over-the-next-5-years-2017-9>, last visited 2020-04-14.
- [3] A. Xu, Z. Liu, Y. Guo, V. Sinha and R. Akkiraju, “A New Chatbot for Customer Service on Social Media”, Proc. of the 2017 Conference on Human Factors in Computer Systems, pp.3506-3510, 2017.
- [4] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan and M. Zhou, “SuperAgent: A Customer Service Chatbot for E-commerce Websites”, Proc. of ACL 2017, pp.97-102, 2017.
- [5] M. Nuruzzaman and O. K. Hussain, “A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks”, IEEE 15th International Conference on e-Business Engineering, 2018.
- [6] “Chatbots in Call Centers”, <https://aws.amazon.com/chatbots-in-call-centers/>, last visited 2020-04-14.
- [7] “社内問い合わせの負担を軽減する、賢い問い合わせ窓口の作り方”, <https://www.ntt.com/bizon/ai-helpdesk-chatbot.html>, last visited 2020-04-14.