

チャットボットにおける 管理容易化のためのシナリオ動的実行方式 Dynamic Execution Method of Scenario for Easy Management in Chatbot

岩崎 信也^{†‡} 薦田 憲久[‡] 藤原 融[‡]
IWASAKI SHINYA[†] KOMODA NORIHISA[‡] FUJIWARA TORU[‡]

概要： チャットボットの作成には、シナリオと呼ばれるチャットボット用の会話テンプレートを作成しなければならず、準備や更新に大きな労力が必要となる。本研究では、チャットボットを適用する際のシナリオ作成や更新の労力を低減するため、ユーザーとの対話内容に基づき、その後の対話に必要な情報を既存システムから取得し、指定された変換方法に従い取得データから動的にシナリオの一部を生成し実行する方式を提案する。実運用予定のチャットボット開発で提案方式を適用したシナリオの作成を検証した結果、提案方式は手動でシナリオを作るのに比べて、シナリオの作成労力を半分以上にできること、ならびに、既存システムの情報を更新するだけでシナリオが変更されるためチャットボットの知識を持たない担当者でも、容易にシナリオを更新できることを、確認した。

キーワード： チャットボット, 対話システム, シナリオ

1. はじめに

チャットボットは、対話型 UI (User Interface) を用いて、人とコンピューターが人間による会話と同じように対話できるシステムであり、人への親和性が高く、最近利用が進んできている。チャットボットにより人が実施していた一定のタスクの代替を目的とする場合、シナリオと呼ばれるチャットボット用の会話テンプレートを作成しなければならず、準備や更新に大きな労力が必要となる[1]。

そこで、本研究では、シナリオの作成・更新の効率化を目的として、シナリオの一部を、他のシステムが保持する情報から動的に生成する方式を提案する。

2. チャットボットにおけるシナリオ

2.1 シナリオ

チャットボットでは、事前にシナリオと呼ばれるチャットボット用の会話テンプレートを定義し、チャットボットに組み込む必要（以後、組込シナリオと呼ぶ）がある。組込シナリオでは、チャットボットで遂行するタスクに合わせて、「利用者に対してどのような質問を提示するか」「利用者の返答によって、次の質問をどのようにするか」といった、ユーザーとチャットボット間での会話の状態や対話の流れを定義する。なお、ここでの「タスク」とは、注文処理や申請処理などの、既存では人や別の手段で実施していた一定の業務を指す。

組込シナリオには、ステップとスロットがある。

(1)ステップ

チャットボットが、ユーザーに発言するメッセージ（質問）や選択肢、ユーザーからの返答に対する処理をステップとして定義する。それらのステップを条件付き遷移によってつなげることで対話の流れを構築する。

(2)スロット

ユーザーからの発言内容や処理結果などを保管する領域をスロットとして定義する。スロットに格納することで、ステップ内でユーザーの返答を参照することができる。

申請処理の組込シナリオ例を図 1 に示す。この例では、ユーザーに名前と生年月日を聞き、返答をスロットに格納、年齢を計算後、未成年の場合は申請不可としている。組込シナリオは、図 1 のように、ステップにより 1 対話ずつの発言内容や処理、遷移条件を定義する。この組込シナリオを組み込んだチャットボットで、未成年が申し込みした場合の対話例が図 2 である。このようにチャットボットの組込シナリオを定義することで、チャットボットはユーザーとの対話の状況により、その後の対話内容を変化させることができる。

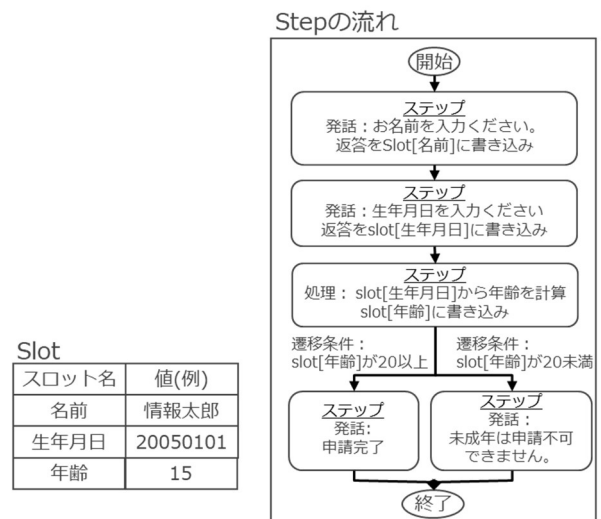


図 1 組込シナリオの例
(20 歳以降のみを対象とする申請処理の組込シナリオ)

[†] 株式会社日立システムズ

[‡] 大阪大学大学院情報科学研究科



図 2 申請処理の組込シナリオの対話例
(申込者が未成年時の申請不可対応)

標準的に、組込シナリオは、チャットボットのシステムにより JSON や XML などによるデータ構造で定義される。組込シナリオの作成は、組込シナリオ用のデータ構造を手作業で記述する場合や、GUI ベースに組込シナリオを作成し組込シナリオ用のデータ構造を出力するシナリオ作成ツールが提供される場合がある。シナリオ作成ツールでは、シナリオの流れをグラフィカルに表現し、ステップ間の遷移設定やステップの入力を容易にすることで、シナリオの作成を支援する。本研究での組込シナリオは、JSON ベースのデータ構造を持ち、シナリオ作成ツールにより作成される。

2.2 シナリオの作成における問題点

チャットボットの組込シナリオの準備は、対話の内容を一つずつ定義しなければならず労力を要する。特にユーザーの選択結果により、その後のシナリオの流れが大きく変わるあるいは分岐が多数のケースでは、組込シナリオが大きく複雑になる傾向があり、組込シナリオを準備や更新するには労力を要する。

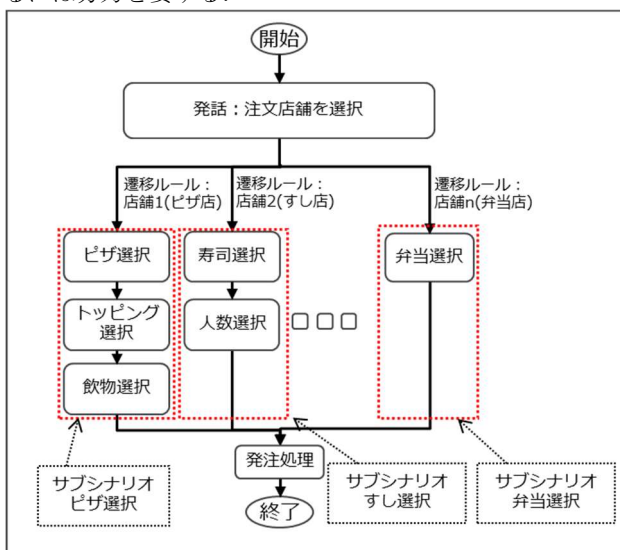


図 3 組込シナリオが大きく複雑になるケース
(出前注文システムにおける注文タスク)

図 3 は、多様な店舗が出店する出前サービスの料理注文シナリオである。出前店舗を選択後、店舗により、その後のシナリオの流れが異なる。例えば、ピザ店の場合、ピザを決定後、トッピングや同時に飲物などを注文するかを確認するが、弁当店の場合は、弁当を選択するだけで注文が完了する。このように店舗別の注文タスクがある場合、シナリオは、図 3 内点線枠で示すとおり、選択された店舗別の注文タスクを実行するステップの流れ(これを以後サブシナリオと呼ぶ)を必要とし、店舗が増えれば増えるほど多量になるため、作成に大きな労力を要する。

また、商用でのチャットボットは、導入企業側に業務システム(以後、既存システムと呼ぶ)があり、WEB システムや電話対応などの既存の手段があるところに並行で導入されることが多い。その結果、チャットボットの組込シナリオの発話文や選択肢などと、既存システムの WEB システムや、電話対応用のマスターで同じ情報を管理しなければならなくなり、労力の増加につながる。

3. 先行研究

チャットボットに関する研究として、ユーザーの発話内容を意図したとおりに認識するための意味解析の研究が盛んである。シナリオに定義されていない発話でも意味が通じればチャットボット側が対応する事例[2]や、ユーザーの発話を解釈し、必要なステップを省略する(例えば、苗字と名前を順に聞くところを、ユーザーが一回で氏名を発話した場合に適切に苗字と名前に分解し名前を聞くステップを省略)事例[3]などがある。意味解析は、ユーザーにフリーテキストで入力させ、その意図によりシナリオを制御する。理想的には、ユーザーの質問に対して人と同じように適切に解釈し、返答できるようになる。しかし、現状の意味解析技術では、人の意図を完全に解釈することは難しく、タスクの実行を目的とするチャットボットでは、フリーテキストによる入力によってシナリオを制御するのは難しい。

特定の業務に特化したチャットボットを提案している事例も多い。旅行代理店のチャットボットとして、旅行代理店が保有するナレッジを基にアーキテクチャーを最適化し、できる限り旅行代理店の担当者と同じ対話ができる事例[4]、ユーザーの希望に応じた医薬情報を提示する事例[5]、QA 対応へ特化しユーザーによる質問の意図にあった返答を提示する事例[6]などがある。これらの研究は、研究対象としている特定の業務にチャットボットを導入するには最適であるが、そのほかの業務には適用できず汎用的ではない。

4. 提案方式

4.1 シナリオの動的生成

本研究では、既存システムにチャットボットのシナリオを生成できる情報(質問や選択肢など)があり、その情報を API ベースに取得できることを前提に、図 3 のように、組込シナリオが大きく複雑になるケースで、サブシナリオを既存システムの情報から動的に生成、実行する方式を提案する(以後、提案方式を動的実行方式と呼ぶ)。複雑かつ多量となるサブシナリオを動的に生成することで、組込シナリオの作成労力を削減し、これまで適用が難しかったケースでのチャットボットの導入をめざす。

動的実行方式の処理と図 3 に動的実行方式を適用した例を図 4 に示す。動的実行方式はデータ取得とサブシナリオ化に分かれる。

(1) データ取得

データ取得は、既存システムに対して API ベースのリクエストを発行しサブシナリオの生成のために必要なデータを取得する。データ取得では、以下の 2 つのパラメーターを入力する。

- ・取得先 API
取得先 API は、データを取得するための既存システムの API の URL である。
- ・スロット情報
スロット情報は、取得先 API のパラメーターに埋め込み、API 先に問い合わせるための情報である。

図 4 の例では、既存システムの API (“xxx.jp?id=ピザ店”) からサブシナリオの生成に必要なデータを取得しており、ピザ店の店舗情報である 3 つのオブジェクトを持つ配列を取得している。

取得データは、そのままではシナリオのデータ構造と異なるためチャットボットで利用できない。サブシナリオ化では、取得データの配列順にステップを生成し、ステップごとに配列の内部オブジェクトの Key 値を質問文(発話の内容)として設定し、Value 値が配列であれば、その値を選択肢とする。スロットは、質問文をスロット名としてステップごとに作られ、ユーザーの返答を格納する。このため、図 4 の例では、メニューと人数を選択する 3 つのステップを持ち、それぞれのステップの返答結果用のスロットを持つサブシナリオが生成される。

(2) サブシナリオ化

動的実行方式の出力として生成したサブシナリオがチャットボットシステムに渡される。

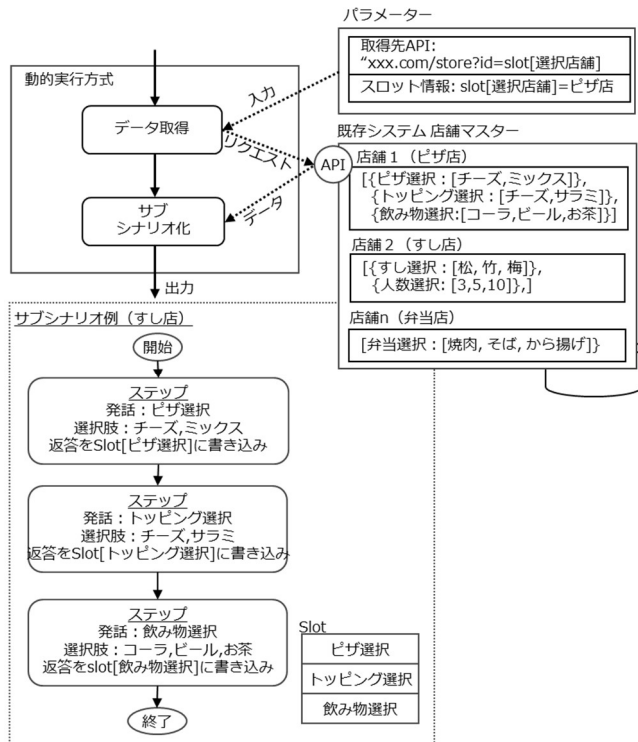


図 4 動的実行方式の処理と処理例

動的実行方式によるサブシナリオの生成の結果、ユーザーから見たチャットボットの対話の流れは動的実行方式を適用しない場合と変わらない。このため、動的実行方式を適用することで、組込シナリオの内、複雑かつ多量となるサブシナリオを事前に作成する必要がなくなる。

上記の方式を実現するシステム構成を図 5 に示す。動的実行方式は、組込シナリオに、動的実行用ステップを定義し、そのステップが開始された際に実行される。動的実行用のステップは、通常のシナリオと同じくシナリオ作成ツール上で作成し、動的実行方式で必要となる取得先 API などのパラメーターは、動的実行用のステップのオプション情報としてシナリオの作成者が指定できる。動的実行方式で生成されたサブシナリオは組込シナリオと同じく、チャットボットシステムが持つシナリオ制御によりユーザーとの対話に使われる。

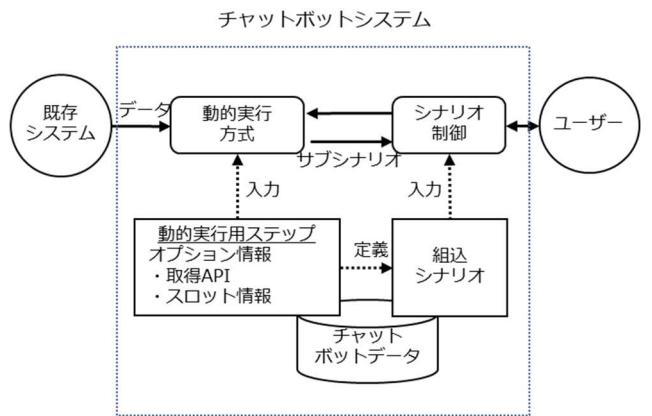


図 5 動的実行方式の構成図

4.2 シナリオの管理

チャットボットは、チャットボットの発話に応じてユーザーが返答するタイミングを制御できない。ユーザーが、対話の途中で離脱し、一定時間後に会話に復帰する可能性も考える必要がある。

会話の復元にはシナリオが必要になる。組込シナリオの場合、ユーザーが通過するサブシナリオ関係なく、単一のシナリオであり、チャットボットに組み込まれてユーザー共通で管理される。一方、動的実行方式により生成されたサブシナリオの場合、ユーザーとの対話結果によりユーザーが必要なサブシナリオのみ生成されるためユーザー別に異なる。このため、動的なサブシナリオの生成時に生成したサブシナリオをユーザーごとのデータ保存領域に格納する。ユーザーがチャットボットとの対話に復帰した際に、そのユーザーのサブシナリオを復元して実行する。復元時に再度既存システムの情報から再生成しない理由は、既存システムに負荷をかけないためである。チャットボットは、ステートレスなアーキテクチャーであることが多く、シナリオの復元回数が多くなる傾向にある。復元のたびに既存システムからデータを取得するのは、既存システム・チャットボット双方の負荷が高い。

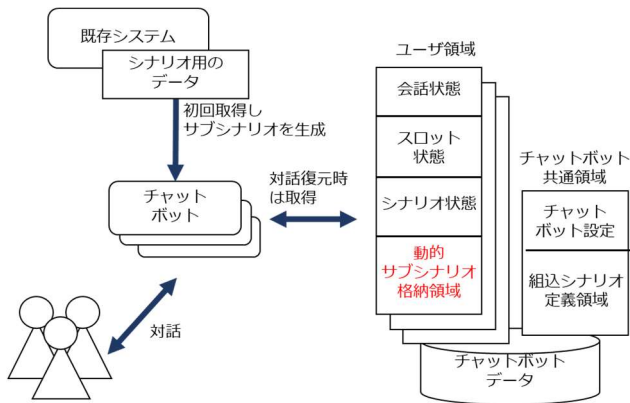


図 6 動的サブシナリオの管理の仕組み

動的に生成したサブシナリオの管理の仕組みが図 6 である。チャットボットの情報保存領域には、チャットボット全体の設定や、組込シナリオなどを格納するチャットボット共通領域と、ユーザー別にシナリオの進行度などの会話状態や、スロットの状態などを格納しておくユーザー領域を持つ。動的なサブシナリオは生成時、ユーザー領域に保管しておく。ユーザーが、会話に復帰時、シナリオの進行度から離脱前に動かしていたシナリオを復元し再実行する。復帰対象のシナリオが動的シナリオだった場合、ユーザー情報領域から対象のシナリオを復元し、再実行する。また、既存システムの情報の更新時は、動的サブシナリオを破棄することで最新の動的サブシナリオを再度生成する。

5. 実験

動的実行方式を適用したシナリオの作成をチャットボットの導入ケースで試行し、動的実行方式を適用しない場合と、シナリオの作成労力を比較した。

実験の対象としたシナリオは、採用応募のユースケースであり、実運用に向けて試行中のものである。このユースケースは、チャットボットのユーザーは申請対象となる企業を選択し、選択した企業に設定されている質問に返答する。返答が完了すると返答結果がデータベースに登録されて終了する。質問のためのサブシナリオの生成に動的実行方式を用いた。組込シナリオの作成はシナリオ作成ツールを利用した。

シナリオの作成に要した労力を比較した結果を表 1 に示す。作成に要した労力は、シナリオの定義を作成するのに必要だった時間を対象としている。具体的には、以下の方法で算出した。

・動的実行方式を適用：

実運用に向けたチャットボット開発における SE の作業記録から、シナリオの作成に要した時間を求めた。尚、この時間には、シナリオの流れや文面などの設計時間や、動的実行方式以外のシナリオに組み込まれるロジック（例えば、企業の一覧表示）の開発時間なども含まれている。

・動的実行方式を適用しない：

SE が 1 企業分の質問を含めたシナリオを作成し、その作成時間から、全企業のサブシナリオ作成時間を求め、サブシナリオ以外の組込シナリオの作成時間と合算した。

表 1 シナリオの作成労力の比較結果

動的実行方式を利用	動的実行方式を利用しない
93 時間	218 時間

シナリオの作成に必要な時間を「動的実行方式を適用しない」場合と比べると「動的実行方式を適用する」場合では、57%削減できた。

また、このケースでは、シナリオ作成の知識のない業務担当者が情報を更新する必要があったが、動的実行方式により、既存システムの情報を更新するだけでシナリオが変更されるためチャットボットの知識を持たない担当者でも、容易にシナリオを更新できることを確認し、業務担当者からも本方式は有用であるとのコメントを得られた。

6. まとめ

本研究では、チャットボットを適用する際のシナリオ作成や更新の労力を削減するため、ユーザーとの対話内容に基づき、その後の対話に必要な情報を既存システムから取得し、指定された変換方法に従い取得情報をシナリオに変換することで、動的にシナリオの一部を生成し実行する方式を提案した。結果として、試行したチャットボットの導入ケースにおいて、提案方式は手動でシナリオを作るのに比べて、シナリオの作成労力を半分以上にすることができた。さらに、既存システムの情報を更新するだけでシナリオが変更されるためチャットボットの知識を持たない担当者でも、容易にシナリオを更新できることを確認した。

参考文献

- [1] 狩野芳伸, “コンピューターに話を通じるか: 対話システムの現在,” 情報管理, vol. 59, no. 10, pp. 658–665, 2017.
- [2] X. Wu, K. Ito, K. Iida, K. Tsuboi, and M. Klyen, “りんな: 女子高生人工知能,” 言語処理学会第 22 回年次大会 発表論文集, pp. 306–309, 2016.
- [3] M. McTear, “Conversation modelling for chatbots: current approaches and future directions,” *Stud. Zur Sprachkommun. Elektron. Sprachsignalverarbeitung*, 2018, Accessed: Jun. 05, 2020.
- [4] A. Argal, S. Gupta, A. Modi, P. Pandey, S. Shim, and C. Choo, “Intelligent travel chatbot for predictive recommendation in echo platform,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2018, pp. 176–183.
- [5] C. Avila et al., “MediBot: An Ontology based Chatbot for Portuguese Speakers Drug’s Users,” presented at the 21st International Conference on Enterprise Information Systems, Jun. 2020, pp. 25–36, Accessed: Jun. 05, 2020.
- [6] G. Guy Andrade, G. Silva, F. Duarte Júnior, G. Santos, F. Lopes de Mendonça, and R. Sousa Júnior, “EvaTalk: A Chatbot System for the Brazilian Government Virtual School,” in *Proceedings of the 22nd International Conference on Enterprise Information Systems*, Prague, Czech Republic, 2020, pp. 556–562.