Technology Reports
AIエージェント　自然言語処理　音声認識／合成
新AIエージェント特集

ユーザの自然言語指示を実行可能なサービスに変換する多目的対話エンジン

イノベーション統括部　住谷 哲夫　山崎 光司
サービスイノベーション部　鎌土 記良　田中 剛
サービスデザイン部　角野 公亮

生活の中で人が普通に使っている話し言葉は曖昧な表現も多い．多目的対話エンジンとは，そうした多様かつ，不明確な自然言語を分析，解釈しサービスに展開する機能を有するエンジンである．その構成は自然対話プラットフォーム，音声認識機能部，音声合成機能部，サービスプラットフォームフロントエンド，ユーザ向けダッシュボード，開発者向けダッシュボードからなる．本稿では多目的対話エンジンの全体概要やエンジンの構成要素について解説する．

1. まえがき

多目的対話エンジンは（**図1**），音声認識／合成，自然言語処理だけでなく，API（Application Program Interface）*1による外部サービスとの連携ができるのが特長である．それによりさまざまなデバイス上でユーザと対話したり，コンテンツを提供したり，また，デバイスを操作するエージェントも簡易に搭載できるので，ユーザにより多くの新しい体験を提供することができる．

ドコモは2012年にサービスを開始したコンシューマ向けサービス「しゃべってコンシェル」，2015年に開始した法人向けサービス「自然対話エンジン」，2016年の「おしゃべりロボットfor Biz」などの提供を介し，800万人を超えるユーザによる30億回以上の発話ログを蓄積，自然言語処理技術における対話性能の向上を図ってきた．多目的対話エンジンは，そのまさにこの6年間のノウハウが詰まったエンジンである．

＊1　API：ソフトウェアの機能を他のプログラムから利用できるように切り出したインタフェース．

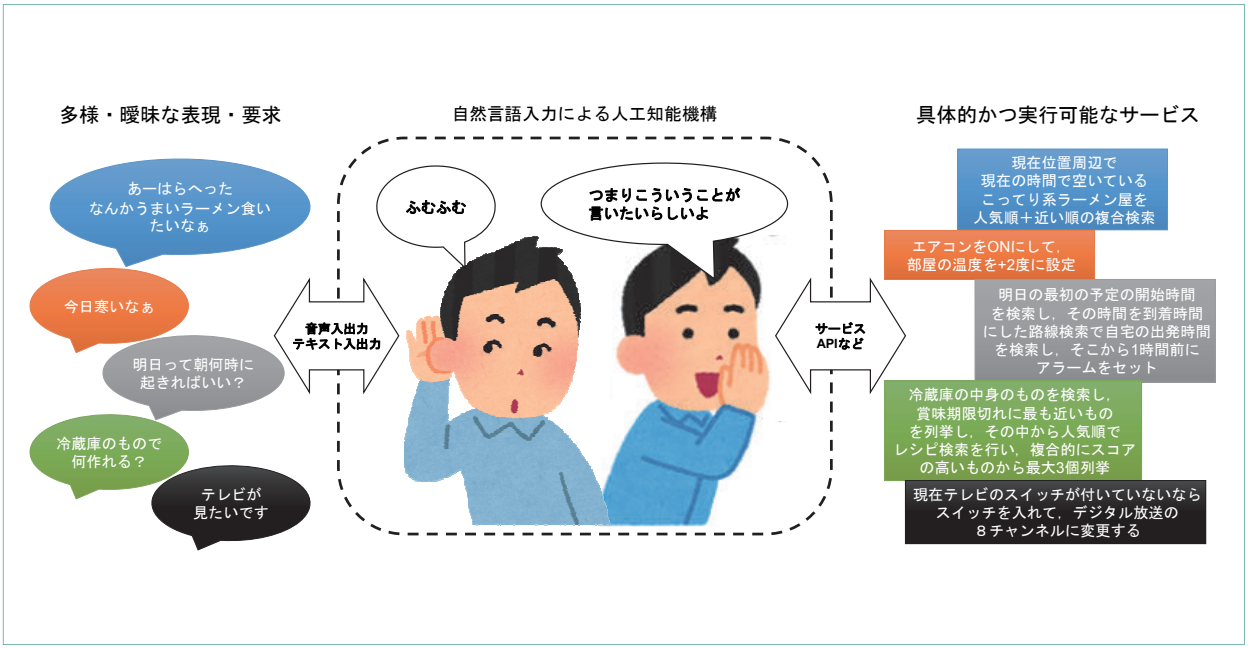

図1　多目的対話エンジンの概要

2. 多目的対話エンジンの技術

2.1　多目的対話エンジンのシステム構成

多目的対話エンジンは，サービスプラットフォームフロントエンド（SPF：Service Platform Frontend），音声認識機能部（ASR：Automatic Speech Recognition），自然対話プラットフォーム（NLU：Natural Language Understanding），音声合成[*2]機能部（TTS：Text To Speech），およびユーザダッシュボード（以下，UDS：User Dashboard），開発者ダッシュボード（以下，DDS：Developer Dashboard）から構成される．対話におけるシステム全体の流れを**図2**に示す．

・ユーザの発話から音声認識

通常の音声認識では各デバイスがデバイスSDK（Software Development Kit）[*3]を包含している．そして，ユーザの発話に応じてエンジンに対してあらかじめ取得した認証トークンと，音声，テキストのいずれかを送信する．ユーザからのすべてのリクエストは最初にSPFにて受け取り，認証トークンが検証される．認証トークンの検証後，音声データはASRに送信され，認識結果がテキスト形式でレスポンスされる．

・自然言語処理から外部サービス連携

NLUに送信された認識結果は，自然言語処理される．自然言語処理の結果判定されたタスクに応じて外部サービスとの連携を行い，ユーザが必要とする情報を取得する．例えば，「今日の天気は」という認識結果が来た場合，天気予報サービスとの連携と判断され，天気予報情報を取得するべくAPIをリクエストし，そのレスポンスを基にユーザへの返答内容をテキストで作成する．

・音声合成からユーザへの返答

作成された返答内容はTTSに送信され，あらかじめ指定された音声モデルで音声データ化される．生成された音声データはSPFを通じ，デバイスに送信され，ユーザからの問合せに対

＊2　**音声合成**：テキストから人工的に音声データを作り出し，テキストを読上げできるようにする技術．
＊3　SDK：アプリケーションを作成するときに必要となる，ドキュメント，ツール，ライブラリ，サンプルプログラムなどからなる開発キット．

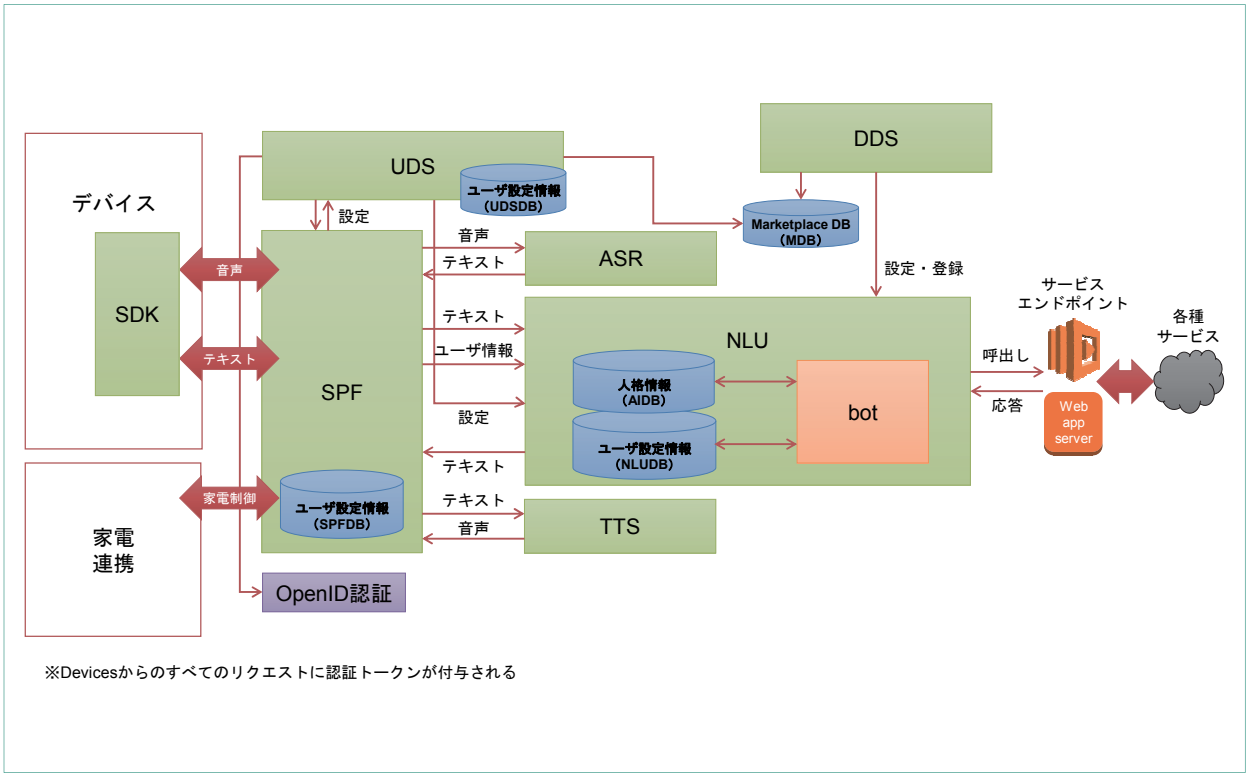

図2　多目的対話エンジンのシステム構成

して音声での回答を返す．ユーザの発話開始から，会話の終了までデバイスSDKとSPFはWebSocketでの接続を維持し，テキストよりもデータが大きくロードに時間がかかる音声による返答にもかかわらず，高速なレスポンスを可能にしている．

2.2　SPF

前述の通り，SPFはクライアントからの連続した対話要求に対し，後段に接続されたバックエンドエンジンの回答の組合せから対話回答を生成し，クライアントに返答する機能をもつ．そのソフトウェアアーキテクチャを**図3**に示す．

単一プロセスにおけるSPFの内部はBlockという処理単位が，Edgeで結ばれ，データ受渡しを行う構造となっている．これにより，音声対話で用いるさまざまな処理を自在に組み替えることができ，変化の早い音声対話技術への柔軟な対応が可能になった．

また，処理がBlockとEdgeに分離されることにより，Block同士が共有するデータはEdgeに集約することができ，各処理の並行性を高めることができる．これにより，テキスト処理だけでなく，音声のような多量のデータを並行処理する必要があるリアルタイムストリーミング*4処理も同一サーバ上で実現することができる．

次に，Blockがもつソフトウェアのレイヤ構造を図3(b)に示す．SPFにおいては，速度やOS最適化に依存する（ED：Environment-dependent）処理は，C＋＋（一部は環境に依存した言語）による高速化を行っている．

また，SPFでは，さらにこれらをラップする環境

*4　ストリーミング：NW上で音声や映像データを送受信するときの通信方法の一種．データを受信しながら，同時に再生を行う．

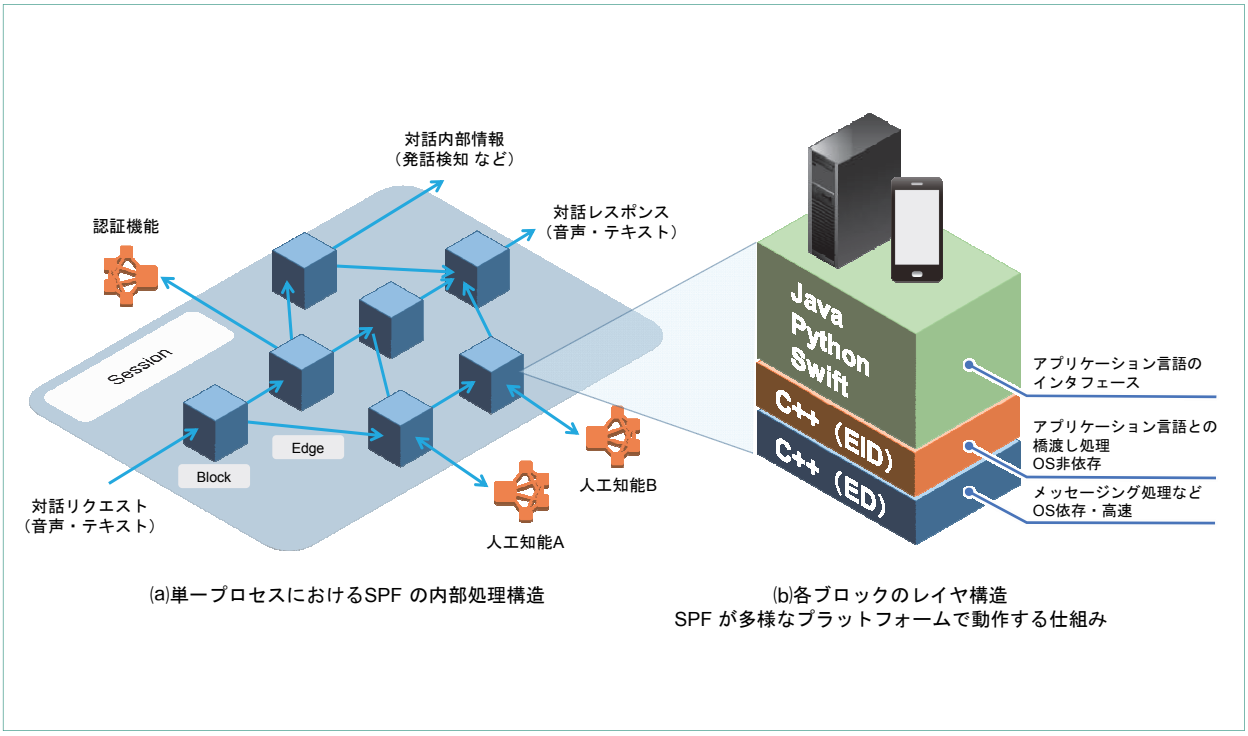

図3　SPFのソフトウェアアーキテクチャ

非依存，プログラミング言語非依存（EID：Environment-independent）のレイヤを介することで，多様なプログラミング言語でBlock処理を記述することができる．これにより，異なる言語で書かれたプログラム間をSPFが媒介することで，サービスの開発期間を短縮することが可能となった．

また，デバイスSDKも本アーキテクチャを用いることにより，サーバ環境のみならず，スマートフォン，組込Linux*5環境などにおいても高速で安定した音声対話処理を実現している．

2.3　NLU

NLUでは，xAIMLと呼ばれる記述言語を用いて上記エージェントを実現する対話システムを構築している．xAIMLとはAIML1.1を基にドコモの自然言語処理技術*6で機能拡張を図った，対話エージェント構築のための記述言語である．

xAIMLでは，1つの対話エージェントを構築する際に，その対話エージェントがもつ機能をそれぞれ独立したbotとして設計することができるため，効率的なシステム開発が可能である．多目的対話エンジンにおけるメインエージェント*7，エキスパートエージェント*8においても，複数のbotを連携させることでさまざまな機能を実現している．

(1)メインエージェント構築のためのデザインパターン

NLUは，人間と会話ができる対話サービスや製品を開発するにあたって，開発者が自由にカスタマイズしてサービスや製品に組み込むことが可能なプラットフォームである．この特長は，対話システムを開発するために役立つ部品を自由に組み合わせることで，めざす対話エージェントを容易に開発できるところである．多目的対話エンジンでは，このNLUを用いて，メインエージェントとエキスパートエージェントの2種類の対話エージェントを構成し，

＊5　組込Linux：携帯情報端末や家電製品など，CPUとソフトウェアが搭載され，用途が特定されている機器のうち，Linux OSで動作するもの．
＊6　自然言語処理技術：人間が日常的に使っている言語（自然言語）をコンピュータに処理させる技術．
＊7　メインエージェント：ユーザとの対話のフロントに立つエージェント．サービス提供者が好きなキャラクタを作成することができ，サービスとデバイスを繋ぐことができる．
＊8　エキスパートエージェント：サービスに特化したエージェントで，メインエージェントから呼び出される．サービス提供者がサービスを既存のメインエージェント（ロボット，botなど）に自由に提供できる．

それらを連携可能とする．

メインエージェントとは，対話システムにおける主人格であり，ユーザとの対話を制御するエージェントである．ユーザがエージェントに対して発話をした場合（以下，ユーザ発話），まず，ユーザの識別子から接続すべきメインエージェントが特定される．その後，メインエージェントで発話内容からユーザの意図が解釈*9され，メインエージェントにおいて実行可能なタスクであった場合はそのままタスクを実行する．一方，例えば，「dグルメお願い」などの特定のエキスパートエージェントを呼び出すような発話がなされた場合は，ユーザ発話をエキスパートエージェントに引き渡し，その後の処理およびユーザとの対話もエキスパートエージェントに引き継ぐ．このように，メインエージェントとエキスパートエージェントは明確に役割が分かれたエージェントとして，それぞれ自由にカスタマイズができる．

また，タスク指向の対話シナリオ，エキスパートエージェントの呼出しシナリオ，コマンド系の対話シナリオ，雑談対話シナリオなど，各シナリオ群単位で優先度を設けることができ，サービスごとに優先されるシナリオを変更することができる．そして，メインエージェントはパートナー企業であれば自由に設計，開発ができ，上記各シナリオ群を自由に組み合わせたり，オリジナルのメインエージェントを開発したりすることも可能である．

エキスパートエージェントとは，メインエージェントから呼び出される，ある特定の分野に特化した専門家エージェントである．DDSによって誰でも簡単に開発することが可能で，開発したエキスパートエージェントは審査を経てAIエージェントAPI専用のMarketplaceに公開することができる．そこで公開されたエキスパートエージェントは，ユーザが使ってみたいと思った時に自由に追加・削除でき，ユーザごとのエキスパートエージェントの管理は，UDSで行うことになる．

⑵bot連携機構によるエキスパートエージェントの実現

多目的対話エンジンにおける各エキスパートエージェントは，DDSによってそれぞれが1つの独立したbotとして生成される．ユーザはUDSを通じて任意のエキスパートエージェントを有効化することで，以下のエキスパートエージェントに関連する機能が利用できるようになる．

・エキスパートエージェントごとに決められた呼出しワードをメインエージェントに入力することで，対象のエキスパートエージェントに取り次いでもらう．
・呼び出したエキスパートエージェントとの対話が終了するまでの間，ユーザの入力は対象のエキスパートエージェントへ転送される．
・エキスパートエージェントからの公序良俗に反する出力をフィルタし，どのような出力をしようとしたかDDSの管理モジュールへ送信する．
・システム管理者やエキスパートエージェント開発者が，エキスパートエージェントの機能を停止することができる．

多目的対話エンジンでは上記の機能を実現するため，ユーザごとのエキスパートエージェントを管理する機能，対話中のbotにユーザの入力を取り次ぐ機能，NGワードのフィルタ機能，エキスパートエージェントのステータスを管理する機能をそれぞれ独立したbotとして設計している（**図4**）．

また，多目的対話エンジンを利用して作成される各種メインエージェントは，これらのbotを利用することで，エキスパートエージェントを提供するための機能を簡単に実装することができる．

2.4 UDS

UDSではユーザ・デバイスの認証処理および，エキスパートエージェントに関わる設定を行うこ

＊9 **意図解釈**：ユーザの発話文章（自然言語）からユーザが意図していることを機械学習などによって特定する技術．ユーザの意図を「タスク」と呼び，例えば，「明日の天気は」「明日は晴れるかな」「明日って雨？」などの文章はすべて天気タスクに判定される．

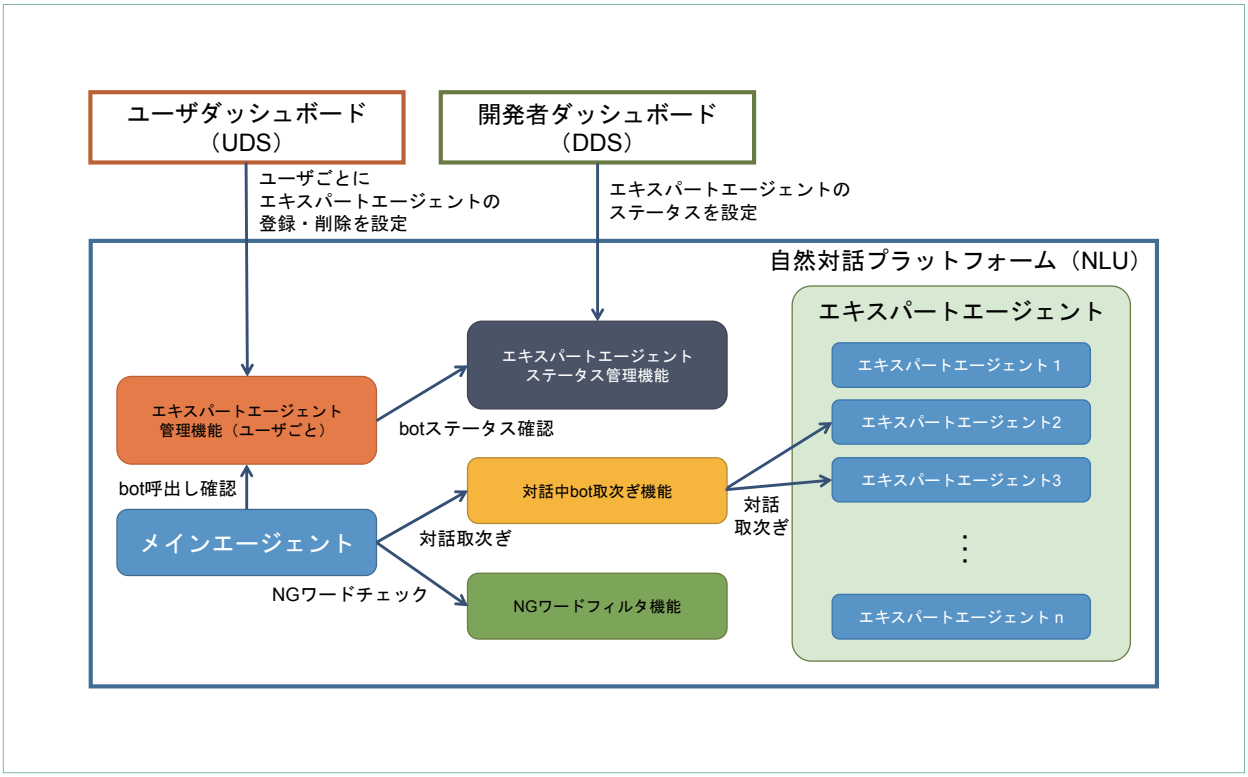

図4　エキスパートエージェントを実現するbot構成

とができる［1］．また，GUI*10だけでなく，REST API*11によってもデバイスの認証を行うことができる．

トライアル環境におけるUDSのイメージは**図5**に示す通りである．UDSは以下の機能を提供している．

・認証機能：ダッシュボードへのログインはOIDC（OpenID Connect）*12をサポートしているdアカウント認証，Google*13認証に対応できる．
・デバイス一覧表示／新規登録：各デバイスに紐づくデバイスIDを登録し，ユーザと紐づけることができる．また，現在登録されているデバイスの一覧を表示し，必要に応じて削除をすることができる．
・ヒストリ表示機能：メインエージェントおよびエキスパートエージェントとの対話の履歴を表示することができる．
・ホームデバイス連携機能：IoTアクセス制御エンジンと連携し，UDSから連携する家電の登録やタグ付などを管理し，エキスパートエージェントから操作するように設定できる．
・サービス追加機能：DDSで登録されたエキスパートエージェントの一覧が表示され，登録することができる．エキスパートエージェントを利用する場合には，事前にこの登録が必要となる．また，エキスパートエージェント側でアカウント連携が必要な場合には，登録時にOAuth 2.0の認可が必要となる．

2.5　DDS

DDSではエキスパートエージェントの作成機能を提供する［2］．

＊10 GUI：操作や表示の対象が絵で表現され，直感的な操作や視認性に優れるインタフェース．
＊11 REST API：RESTの制約に従ったAPI．RESTはRoy Fielding氏が2000年に提唱した設計原則を基に発展した，ソフトウェアアーキテクチャのスタイル．
＊12 OIDC：インターネット上にあるさまざまなWebサイトや，モバイルアプリなどを利用する際に1つのIDで認証を実現できるようにするID連携の仕組み．
＊13 Google：Google, LLCの商標または登録商標．

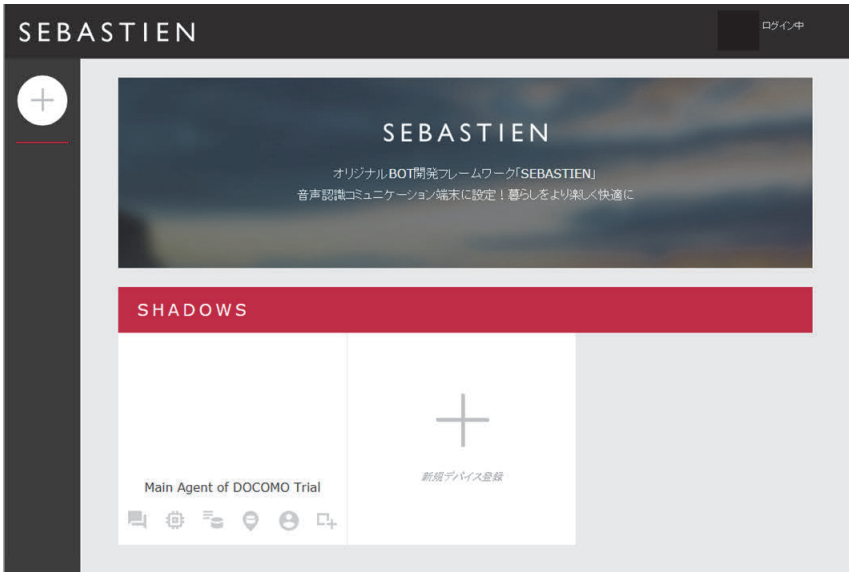

図5　ユーザーダッシュボードイメージ

⑴DDSが提供するエキスパートエージェント作成機能の設計思想

エキスパートエージェントの最終目的は「具体的なサービスをユーザに提供すること」なので，そのための対話設計が行えるようになっている．

また，対話設計は，自然言語処理の知識や対話設計の知識などは特別に必要なく，画面上の設定値を埋めるだけで完結するようにツールが設計されている．そのため，対話サービスを作ったことがない開発者でも簡単，かつ速やかにエージェントが作成できる仕様となっている．

⑵エキスパートエージェントの対話設計

DDSで作成するエキスパートエージェントは，サービス提供に適したスロットフィリング＊14を具備したタスク志向型の対話設計を採用している．具体的な対話シーケンスを下記に示す．

①エキスパートエージェントが発話から意図（Intent＊15）とパラメータ（Slot）を抽出する．

②曖昧なユーザ発話に対しては，エキスパートエージェントから「聞き返す」ことでタスク実行を促す．

③最終的に特定した意図情報（Intent，Slot）を，設定されている外部プログラム（API）へPOSTリクエスト＊16で送信する．

④外部プログラムは受け取った意図情報をもとにサービスを実行する．

⑤外部プログラムはサービスの実行結果と合わせてユーザへの返答（対話文）を返却する．

⑥エキスパートエージェントはプログラムからの返答をユーザに返す．

3. あとがき

本稿では，ユーザから発せられた多様・曖昧な表現を含む自然言語を解釈し，具体的かつ実行可能なサービスに変換する多目的対話エンジンについて解

＊14 スロットフィリング：タスクを実行する際に必要なパラメータをテキストから抽出する機能．例として，天気検索タスクを実行するために「今日の東京の天気は？」というテキストから，「今日」と「東京」という「時間」と「場所」というパラメータを抽出すること．

＊15 Intent：エキスパートエージェント内で定義できるタスクを示す．

説した．メインエージェント，エキスパートエージェントの概念を有し，さまざまなデバイスの上でユーザと対話をしたり，コンテンツを提供したり，デバイスを操作してユーザに新しい体験を提供することが可能となった．今後は，基礎的，かつ，複数のプラットフォーム等で共通に必要となる対話技術の研究開発や，各分野の高度意図解釈技術の研究開発など，高度な対話を可能にするエージェント生成技術に取り組み，APIを高度化させていきたい．

文　献

［1］ SEBASTIENホームページ.
https://users.sebastien.ai/
［2］ Expert Agent Developer Dashboardホームページ.
https://developers.sebastien.ai/

＊16 POSTリクエスト：HTTP通信でクライアント（Webブラウザなど）からWebサーバへ送るリクエストの種類の1つで，URLで指定したプログラムなどに対してクライアントからデータを送信するためのもの．GETやHEADなどのリクエストはヘッダのみだが，POSTリクエストの場合にはボディ部があり，ここに送信したいデータを記述する．大きなデータやファイルをサーバに送るのに使われる．