

タスクとコンテンツが独立開発可能な 対話シナリオ記述環境の作成

中村寛 藤本英輝 河野恭之 木戸出正継

奈良先端科学技術大学院大学 情報科学研究科

〒630-0192 奈良県生駒市高山町 8916-5

E-mail: {hirosi-n,eiki-f,kono,kidode}@is.naist.jp

概要

本研究では、一般ユーザが容易に対話システムを作成できる対話シナリオ記述環境を作成した。容易に対話システムが作成出来るように、共通の実行系で対話ルールファイルを入れ替えて対話を行うシステムを想定した。記述が複雑になる対話ルールファイルを容易に作成できるように、対話シナリオを定義した。対話シナリオは、コンテンツとタスクを組み合わせることで容易に作成できるようにし、対話システムの知識、対話ルールの知識を持たないユーザでも、容易に対話シナリオが出来るような記述環境を作成した。

A scenario authoring environment for ubiquitous dialogue systems whose task and contents can be independently developed

Hiroshi Nakamura Eiki Fujimoto Yasuyuki Kono Masatsugu Kidode

Graduate School of Information Science, Nara Institute of Science and Technology

8916-5, Takayama-cho, Ikoma-shi, Nara 630-0192, Japan

E-mail: {hirosi-n,eiki-f,kono,kidode}@is.naist.jp

Abstract

This paper presents a scenario authoring environment for ubiquitous dialogue systems that can be easily developed and maintained by non-experts. The dialogue engine for such ubiquitously spread dialogue systems should have an ability to load “scenarios” depending on contexts/tasks. The scenarios should be easily developed and maintained. In our system, the developer describes dialogue scenarios and they are compiled into a set of production rules. The initial version of a scenario is generated by combining task and contents descriptions.

1 はじめに

本研究の目的は、開発コストのかかる対話システムの作成を、対話システム作成の知識が十分でないユーザが対話システムを容易に作成するための対話シナリオ記述環境を作成することである。

ユーザとの対話を行いタスクを遂行するタスク指向対話システムは一般的にドメイン依存、またはタスク依存での開発が行われている。対話システムを新たに作成する際に既存のシステムの一部を流用するなどの作業は困難であり、ほぼ一から作り直すことになるので莫大なコストがかかる。このような対

話システム作成にかかるコストを軽減するために、対話システムの移植性、拡張性を上げるための研究が行われている。

鈴木らは、移植性の高い音声対話システムの研究を行っている [1]。これは以前研究が行われていた移植性を考慮した対話システム [2] がサポートしていた、検索タスクをもとに、予約タスクの対話が可能なシステムに変更したものである。また、鈴木ら [3] は対話のタスクを細分化して基本モジュールを定義し、そのモジュールを組み合わせることで対話システムを実装した。

本研究は、ユーザと対話を行いタスクを遂行する

家庭用ロボットや、美術館や博物館で展示品ごとに解説する案内システムなど、様々な場所においてユーザと対話を行いタスクを遂行するシステムを作成するための基盤となる実行系の確立と、その基盤上での対話シナリオ記述の簡便化とコストの削減が目的である。実行系を共通で利用し、実行する対話の内容を自由に入れ替えられる対話システムが作成できれば、場所や物に応じた対話システムが容易に作成できる。このような記述環境を作成するには、異なるタスク対話を記述する共通のルールファイルの形式、使用環境に応じたルールファイルが容易に作成できる記述環境が必要になると考えられる。

本研究では,AIML[4] ベースの対話ルールを生成する、対話シナリオ記述環境を作成した。記述環境内で、対話シナリオ作成者は、あらかじめ作成されたタスクモジュールとコンテンツを組み合わせ対話シナリオを作成する。その後本システムは作成者が作成したシナリオを AIML 形式の対話ルールファイルに変換する。

以下本稿では、システムが生成しユーザとの対話に用いられる対話ルールファイル、システム作成者が記述する対話シナリオの形状と概要をまず述べる。次に、我々が作成した対話システムの概略を述べる。本システムを用い対話シナリオを作成する時の対話シナリオ作成者の役割を述べ、我々が作成したタスク対話処理モジュールの例をいくつか説明する。最後に、システムが自動的に行うシナリオ作成支援について述べ、まとめる。

2 対話シナリオ

本研究における対話システムの概要を図 1 に示す。我々は、実世界上の様々な場所や物に、対話ルールファイルが埋め込まれており、対話システム(実行系)が、場面に依りてそれらのルールファイルを読み取ってユーザとの対話を行う世界を想定している。例えば、美術館で来場者が本実行系を搭載した PDA をもって閲覧し、各展示物に『展示物案内対話ルールファイル』が埋め込まれているとする。実行系は来場者が展示物の前に来たとき、その展示物のルールファイルを読み込み、対話による案内、解説、

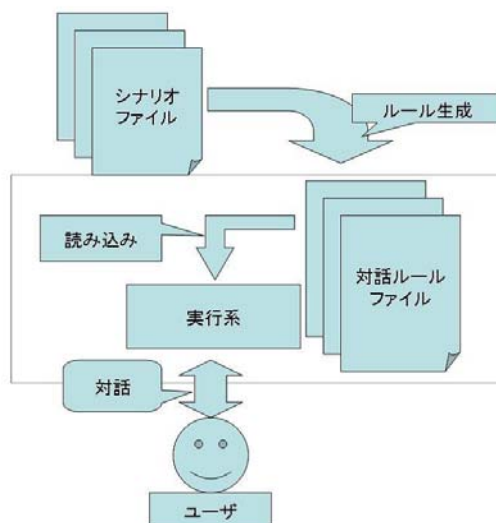


図 1: 対話システム概要

```

<category>
  <pattern>予約</pattern>
  <template>
    <condition name="state">
      <li value="true">
        予約を開始します
        <think>
          <set name="topic">reserve</set>
        </think>
      </li>
      <li value="false">
        現在予約できません
      </li>
    </template>
  </category>

```

図 2: AIML 記述例

ユーザ要求への対応などを行う。また、展示物ではなく美術館全体の案内を受けたいときには、館内全体案内用のルールファイルを読み込み実行する。このように本実行系では、ルールファイルのみを入れ替えることで場面に依りたユーザとの対話が可能になる。ここで重要なことは、対話システム作成者が実行系の詳細を把握していなくても、対話シナリオを記述できることである。

2.1 対話ルール

個々の対話ルールは、あるユーザ入力に対するシステムの応答が規定されたプロダクションルールとして記述される。単純なルールの組み合わせで対話システムを実現することにより、以下の利点が得られる。

- 実行系が容易に構成可能
- 最小単位でのルールの追加・削除が容易

しかし、以下の短所もある。

- システムの全体像の把握が困難
- ルールから対話の状態を推定するのが困難
- ルールの記述量が膨大

プロダクションルールは極所的な入出力関係の記述・把握には適しているが、全体像の把握には適していない。このため、システム作成者やその更新を行おうとする者は、記述された個々のルールを確認することはできるが、システムの全容及び、対話の流れを把握することが困難である。また、複雑な状態遷移をするタスク対話のルールを記述するのは、ルールが複雑になり、記述量も膨大になる。このため本研究では、図 1 に示すようにシステム作成者は状態遷移が表現できるシナリオを記述することとし、そのシナリオを基にシステムが対話ルールベースを作成することとした。この構成により、ユーザと対話を行う実行形の実装が容易となる。

対話ルールファイルの形式は AIML[4] である。AIML は XML ベースの対話記述言語である。pattern タグで囲まれた部分とユーザからの入力を比較し、一致したときに template タグ内に記述された出力や内部状態の変更を行う。pattern と template の組み合わせを category として定義する。AIML 記述の例を図 2 に示す。

2.2 対話シナリオ

対話システム作成者が対話ルールを直接記述するのは上記のように現実的でないため、対話ルールを容易に生成できる記述環境が必要になる。本研究では、システム作成者が対話シナリオを記述する環境を作成した。

対話シナリオとは対話の状態定義と、状態間の遷移規則が記述されたものである。システム作成者はコンテンツ記述とタスク記述を組み合わせで対話シナリオを作成する。コンテンツにはコンテンツフレームにそったデータが記述されており、タスクには未定義の状態と、状態間の遷移ルールが記述されている。コンテンツとタスクの詳細は 3 章で述べる。

システム作成者は、選択したタスクにおいて枠組みのみが定義されている各状態記述にコンテンツのデータを適用すると共に、必要な情報を追加することで対話シナリオを生成する。記述環境は、システム作成者によって記述された対話シナリオを対話ルールに変換する。タスク内で状態の枠組みと遷移ルールが定義されているので、システム作成者はタスク内の各状態に適応させるコンテンツのデータを選択するだけで、対話シナリオが生成できる。記述環境で対話シナリオを自動的に対話ルールに変換するので、システム作成者は対話ルールのことを詳細に知らなくても、対話ルールの記述が容易に作成できる。このように、対話シナリオ記述に必要なタスクとコンテンツの形式を定義し、対話シナリオからルールを生成する記述環境を作成することで、容易に対話ルールを作成することを実現した

3 対話シナリオ記述環境

図 3 に作成した対話シナリオ記述環境の概要を示す。記述環境はタスクモジュールとコンテンツを統合して対話シナリオを生成する。対話に利用されるコンテンツをコンテンツフレームに沿って作成するコンテンツ作成者、タスク対話の未定義の状態と遷移ルールが記述されたタスクモジュールを作成するタスク作成者、コンテンツとタスクを組み合わせでシナリオを作るシナリオ作成者、の 3 種類の作成者が対話シナリオ記述に関与する。本章では、それぞれの作成者が行う作成内容を説明する。

3.1 コンテンツ作成者

コンテンツ作成者とは、あらかじめ設定されたコンテンツフレームに沿ってコンテンツ内容を追加する人を指す。コンテンツフレームとは、ある特定のコンテンツの記述に必要な情報の枠組みがフレーム

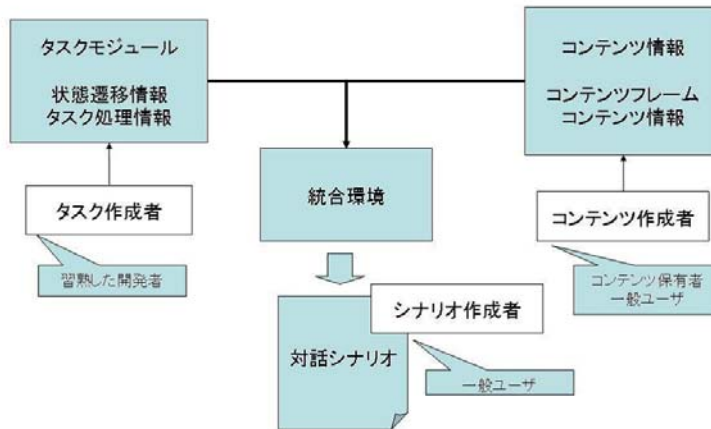


図 3: 作成環境

表 1: コンテンツフレーム 例：ホテル宿泊プランフレーム

一般名称	項目属性	入力例
宿泊プラン名	Name	シングルプラン
部屋	Room	シングル
人数	Member	1
使用料金	Charge	7000
食事内容	Food	無し

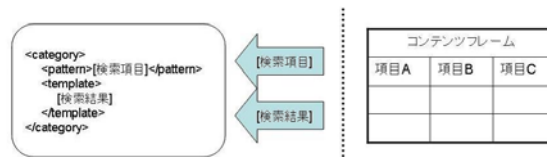


図 4: 検索用シナリオ

形式で定義されているものである。

個々のコンテンツフレームには、表 1 に示すように記述項目が設定されており一般名称と項目属性が定められている。一般名称は、タスクに組み込まれるときに変更可能な変数名である。項目属性は、記述する項目の変数型を宣言したものであり、また記述項目に一意につけられて変更不可能な物である。

表 1 にホテル個室フレームの例を示す。表のように『ホテル宿泊プランフレーム』を考えると、宿泊プランをあらわす「宿泊プラン」、最大宿泊人数を示した「人数」、「部屋」の「使用料金」、「食事内容」、などが上げられる。

3.2 タスク作成者

タスク作成者とは、記述環境内で使用されるタスクモジュール作成を行う者を指す。タスク作成者は対話システムの知識、および出力する対話ルールの知識を有している開発者とする。

3.2.1 タスクモジュール

タスクモジュールは、タスクの枠組みのみが定義されている状態が記述されている。タスクモジュールは、コンテンツと独立に記述されることを想定している。ただし、コンテンツ項目を想定して記述する必要があり、特定のコンテンツに依存したタスクを作成することも可能である。

検索タスク

簡単な作成例として、特定の項目で検索を行う検索タスクの例を示す。我々が作成対象としているのは予め作成された対話ルールを読み込み、対話を実行するシステムでなので、検索項目の情報が既知でなければいけない。そのために、検索タスクを実行するときには、あらかじめ受付可能な検索項目とその検索結果を対話ルール中に記述しておく。

検索タスクモジュールには、検索用の対話ルールのテンプレート書式が記述されており、指定されたコンテンツの検索用項目と検索結果用の項目から、

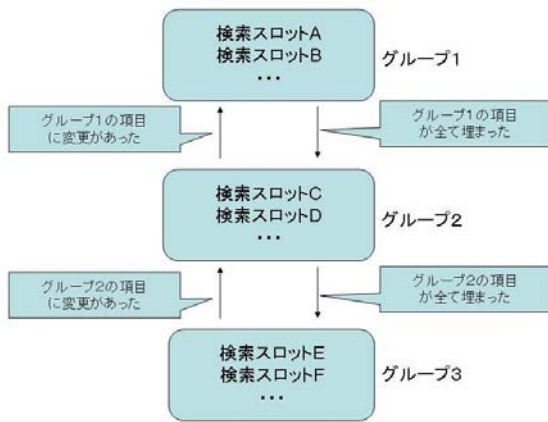


図 5: 依存スロット検索タスク状態遷移

コンテンツのデータを読み取りテンプレート書式に沿ってルールを生成する (図 4)。このときの使用項目の選択は、シナリオ作成者が行うことになる。3.2.2 章でシナリオ作成者の説明を行う。

単純な検索タスクのほかにも様々なタスクが考えられる。我々が作成したタスクの例を以下に示す。

依存スロット検索タスク

依存スロット検索タスクの簡単な状態遷移モデルを図 5 に示す。このモジュールは複数のスロットの依存関係を考慮しながら検索を行うタスク対話である。具体的な例を挙げると、新幹線を検索するときには依存スロット検索が想定でき、『新幹線の種類』 + 『出発駅』から、自動的に『到着駅』になる候補が絞られる。

このタスクを遂行する対話ルールを作成者が直接記述しようとする、記述が複雑になり、またルールの記述手法についてよく理解しておかなければならない。このような複雑な記述を必要とするものはタスクモジュールとして開発を行い再利用できるようにしておく。

予約タスク

このタスクは予約を遂行するために、コンテンツ固有の情報を取得し、唯一特定できるコンテンツの選択を行った後に、ユーザの情報を取得するタスクである。検索タスクと類似する部分があるが、ユーザ固有情報を取得するという、コンテンツ中にはな

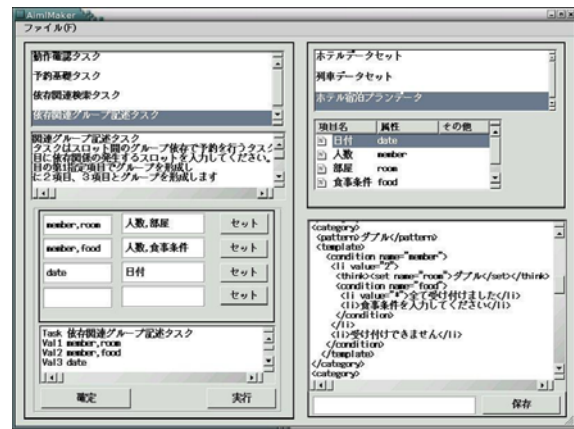


図 6: インタフェース画面

い状態のモデルが定義されている。

3.3 シナリオ作成者

シナリオ作成者とは、これまで説明してきたタスクとコンテンツを使い対話シナリオ記述環境で対話シナリオを作成する者を指す。シナリオ作成者は、対話ルールや対話システムの知識が十分でないユーザでもよい。

シナリオ作成者が行う作業は、コンテンツの選択、タスクモジュールの選択、コンテンツ項目の選択となる。タスクモジュールとコンテンツはすでに用意されているもののなかから選択する。

コンテンツ項目の指定とは、図 4 の例ならば、コンテンツフレーム中の項目から、検索に利用する項目、検索結果に利用する項目を指定することである。

『ホテル宿泊プランコンテンツ』を『検索タスク』に適用させるときには、『宿泊プラン名』が検索結果になり、『人数』が検索項目となりえるので、このような、コンテンツ項目の割り当てをシナリオ作成者が行う。

3.4 記述環境

シナリオ作成者用のインタフェース画面を図 6 に示す。このインターフェースでシナリオ作成者の支援を行う。

タスクとコンテンツを指定し、コンテンツの項目を選択しタスクを統合する。タスクモジュールは、コンテンツの項目を割り当てるだけで対話ルールを生成する。

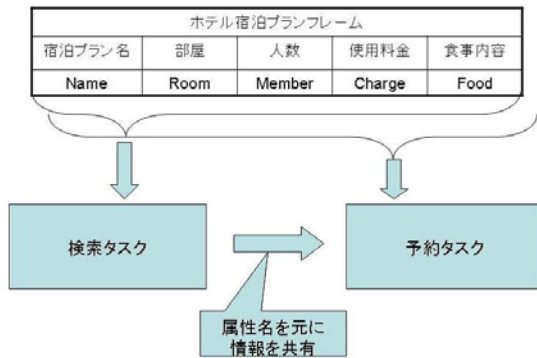


図 7: 情報共有

4 シナリオ作成支援

本章では、対話シナリオ記述環境で対話シナリオを作成するときに1タスクと1コンテンツの適用以外の支援の内容を説明する。

4.1 異種タスク同一コンテンツ連結

複数のタスクを組み合わせることでタスクを遂行するときのシナリオ作成支援について説明する。

個々のタスクモジュールは、それぞれ独立して記述されており、また、タスク作成者が異なっていることも考えられる。しかし、検索タスクを実行した後に検索に利用した入力データを引継ぎ、予約タスクに移行することも考えられる。このように、同一コンテンツで複数タスクを連結したようなシナリオを作成するときには、あるタスクで得た情報を他のタスクに引き継ぐ処理が必要になる。コンテンツを共有してタスクを作成するので、コンテンツの属性情報を元にタスク間を連結する。ホテル宿泊プランコンテンツと検索、予約タスクの例を挙げる。2種類のタスクを同一のコンテンツに適用させたとき、システムはユーザから得た情報をタスク間で共有してタスクを遂行するシナリオを生成する。情報共有のために、コンテンツの項目情報を利用する(図7)。

4.2 応答項目追加

提案する対話シナリオ記述環境では、入出力に使用する文をコンテンツ中から取得して、対話ルールに当てはめて対話を行う。しかし、それだけでは応

答文に不十分な点があるので、それを補うためにできる範囲で自動的に応答文を追加する。そのために、コンテンツ項目に割り当てられた項目属性に注目する。項目属性は、変数の型宣言のようなもので、コンテンツ作成者が Date 属性の項目に情報を追加しようとするときには日付の項目しか入力することができない。Date 属性の場合であれば、検索などを行うときに『来週の 曜日』や、『明後日』などのユーザ入力があることが考えられる。このようなパターンは、コンテンツ中に記述することができないので、そのようなパターンの変換リストを予め用意しておき、システムが対話ルールを生成するときに、このリストを参照してパターンに対応したルールを記述するようにした。

5 まとめ

本研究では、対話システムに詳しい知識を持たない作成者が対話システムを作成できる対話シナリオ記述環境を作成した。対話システムが行う対話のルールを対話シナリオとし単純な対話ルールの記述に落とし込み、コンテンツ作成者とシナリオ作成者を支援する記述環境の作成を行った。

今後は、タスクモジュールの種類の定義、減少した作成コストの定量的な評価を行う必要がある。

謝辞

本研究の一部は、科学研究費補助金(若手研究(B)14780287)によるものである。

参考文献

- [1] 鈴木, 梅田, 小暮, 中川: 移植性の高いデータベース検索・予約用音声対話システム, 情報処理学会, 研究会資料, 2004-SLP-53(5), (2004)
- [2] 小暮, 中川: 移植性の高いデータベース検索用音声対話システムの試作, 情報処理学会, 音声言語情報処理学会, (1999)
- [3] 鈴木, 池ヶ谷, 野口, 他: モジュラリティの高い対話制御ルール設計とその具体的な対話ドメインへの適用方法に関する研究, SIG-SLUD-A303-12, pp.73-78, (2004)
- [4] AIML 1.0.1: <http://www.alicebot.org/TR/2001/WD-aiml/>