

## ЛОКАЛЬНЫЙ И ГЛОБАЛЬНЫЙ УРОВНИ УПРАВЛЕНИЯ ДИАЛОГОМ THE LOCAL AND THE GLOBAL LEVELS OF DIALOGUE MANAGEMENT

Ли И.В. (lee@iiias.spb.su)

Санкт-Петербургский институт информатики и автоматизации РАН

В статье кратко описана модель диалога на основе фреймового подхода для задачи представления справочной информации по авиарейсам. Также рассмотрена модель управления диалогом представленная локальным и глобальным менеджерами диалога.

### 1. Введение

Большинство реальных систем речевого диалога разрабатывается методами ограничения взаимодействия, при котором инициатива ведения диалога полностью передается системе [1]. Диалог становится управляемым, ограничивается свобода действий обоих участников диалога (пользователя и системы), что не характерно человеческому диалогу. В реальном диалоге человек, инициирующий диалог, имеет возможность выбирать удобную для него форму речи, последовательность подачи информации релевантной и нерелевантной текущей ситуации. Зачастую, человек стремится передать в одном высказывании сложную необходимую для запроса информацию. С этой точки зрения диалог для автоматических систем становится гораздо более сложным, поскольку задача состоит в том, чтобы построить систему, адекватную человеческому диалогу.

В идеале, в основе модели диалога должны лежать известные теории речевых и коммуникативных актов. На основе этих теорий строится семантический анализатор, способный вычленять объекты и действия, а также накладывать смысловые связи между выделенными объектами. В реальности же модели, разработанные для существующих диалоговых систем, являются более простыми и ограниченными [2]. Наиболее простой моделью диалога является конечный автомат, который содержит не только структуру диалога, а также все возможные диалоговые высказывания. Конечный автомат требует разработки строго фиксированной структуры задачи, а также фиксированной модели дискурса. Более гибкая модель на основе фреймов позволяет извлекать необходимую информацию из свободно сформулированного высказывания пользователя для выполнения определенного действия, поэтому большинство систем речевого диалога, претендующих на естественность человеко-машинного взаимодействия, строится на основе фреймового подхода [3,4,5].

Принятие решений о действиях системы выполняет менеджер диалога, в функции которого также входит взаимодействие с внешними источниками информации, формирование фрейма ответа системы и передача его на модуль синтеза речи. Эффективность работы менеджера диалога, в первую очередь, зависят от разработанной стратегии диалога. В простых стратегиях выполняется переход по направленному графу состояний системы. При этом связи между состояниями заранее определены, и пользователь строго следует указаниям, формализуя запрос в соответствии с требованиями. В том случае, если инициатива поочередно переходит то к пользователю, то к системе, можно говорить о смешанной инициативе [1, 6]. Именно такая стратегия позволяет учесть, когда и кто активен в определенный момент диалога, когда можно изменить ход диалога, а когда нет. При переходе инициативы к пользователю, система должна учитывать все аспекты запроса, а именно:

1. На каком уровне действий находится пользователь. Говорит ли о взаимодействии или выполняет задачу?
2. Какой речевой акт выполнил пользователь. К примеру, запрос или подтверждение?

Тогда, глобально, по отношению к решаемой задаче, функция системы состоит в вычленении необходимой информации из высказывания пользователя и выполнении действия, соответствующего полученному запросу. С другой стороны, пользователь может произнести высказывание, бесполезное, с точки зрения задачи, но направленное на управление системой или связанное с проблемами взаимодействия. В этом случае система решает локальную задачу анализа входного высказывания: проверяет его информативность и полезность. Следовательно, есть смысл условно разделить функции системы в отношении распознанной естественно-языковой последовательности с использованием локального и глобального менеджеров диалога [7].

В рамках данной статьи будут рассмотрены модель диалога на основе фреймов с использованием смешанной инициативы и два типа менеджера диалога: локальный и глобальный. Обе эти идеи не новы и используются в разрабатываемых системах [5, 7]. Мы же, основываясь на результатах этих разработок, пытаемся соединить преимущества обоих подходов, как нам кажется, успешно дополняющих друг друга.

## 2. Модель диалога на основе фреймов

Модель диалога должна представлять собой некую структуру, отражающую процесс диалогового взаимодействия между автоматической системой и пользователем. На каждом шаге взаимодействия выполняются определенные действия, определяющие ход диалога. Фрейм позволяет предоставить пользователю некоторую свободу дискурса и вести диалог в удобной ему речевой форме. Основным недостатком фреймового подхода считается то, что фрейм не учитывает историю диалога, поскольку в отличие от сетевой модели не содержит хронологическую информацию о развитии диалога. Однако, для решения конкретной задачи информации, предоставляемой фреймом вполне достаточно. Фрейм хранит полученные данные, позволяет определить следующий «открытый» слот и активировать связанное с ним действие системы.

В ходе построения модели прикладной задачи в первую очередь определяются основные цели и подцели. В настоящий момент нами разрабатывается модель речевого взаимодействия для прототипа системы «Полетели», которая выполняет задачу предоставления информации по авиа рейсам. Система позволяет осуществить речевой ввод запроса об интересующих рейсах и выполнить поиск по базе данных сайта [www.pulkovo.ru](http://www.pulkovo.ru). В ходе взаимодействия система решает ряд подзадач, порядок которых фиксирован. Каждая подзадача представлена фреймовой моделью, заполнение которой реализуется инициацией соответствующих поддиалогов. Таким образом, порядок подзадач задается жестко, т.е. управляется системой, а для представления моделей поддиалогов используются фреймы, дающие пользователю определенную свободу выбора инициативы и речевой формы. При таком подходе реализуется смешенная инициатива, когда активная роль переходит от системы к пользователю и обратно, что позволяет обеспечить надежность (когда инициативна система) и естественность речевого диалога (когда инициативен пользователь).

Порядок подзадач представлен в таблице 1. Фреймами не представлены подзадачи формирования запроса по рейсу, поскольку сама по себе задача взаимодействия с внешними источниками не связана с получением данных от пользователя.

№	Название подзадачи
1	Получение фрейма рейса. Фрейм 1
2	Формирование запроса к БД по рейсу
3	Обработка результата запроса по рейсу. Фрейм 2
4	Получение фрейма обратного рейса. Фрейм 3
5	Формирование запроса к БД по обратному рейсу
6	Обработка результата запроса по обратному рейсу. Фрейм 4

Таблица 1. Порядок подзадач системы «Полетели»

Каждый из указанных фреймов включает в себя слоты, значения которых необходимо получить для решения подзадач в процессе ведения поддиалога. Для примера представлен фрейм рейса в таблице 2.

Имя слота	Значение слота	Способ подтверждения значения слота	Присоединенная процедура
Пункт отправления			
Пункт прибытия			
Число			
Месяц			
Класс (эконом, бизнес)			

Таблица 2. Фрейм 1. «Рейс»

Пользователю предоставляется определенная свобода выбора последовательности предоставления данных слотов в рамках одного фрейма. К примеру, пользователь может начать запрос с определения даты рейса, а потом перейти к вводу данных о пунктах отправления и прибытия. В ходе взаимодействия система пытается вычлениить из входного высказывания значения слотов, заданных во фрейме. Процесс принятия решений при работе с фреймами выполняет менеджер диалога, структура которого описана в следующем разделе.

За четкое выполнение действий в модели диалога несет ответственность менеджер диалога (МД). Условно можно выделить два основных типа менеджера диалога по его функциям:

- Локальный МД, оценивающий достоверность извлеченной информации и отвечающий за разрешение проблем непонимания, возникающих из-за ошибок распознавания или непонимания пользователя;

- Глобальный МД, отвечающий за интерпретацию запроса и интеграцию в текущий контекст, а также выбор действия системы. Другие функции глобального МД состоят во взаимодействии с внешними источниками информации и формировании семантического фрейма системы, используемого в модуле синтеза речи при генерации выходного высказывания.

Оба менеджера следуют стратегиям, предназначенным для разрешения конкретных ситуаций, возникающих в процесс взаимодействия.

### *3. Стратегии локального менеджера диалога*

Локальный менеджер диалога выполняет внутреннюю обработку запроса пользователя, следуя определенным стратегиям. При этом обязательно рассматриваются пять основных ситуаций.

- (1) Ситуация «Успешное взаимодействие», когда пользователь задает вопрос приемлемыми способами. В этой ситуации выполняется простой возврат к глобальному менеджеру, который использует глобальную стратегию управления глобальным диалогом для активации следующего узла.

В ситуациях (2)-(5), управление остается на локальном уровне. В этих проблематичных случаях необходимо восстановление диалога:

- (2) Ситуация «Запрос повторения», когда пользователь просит повторить последнюю подсказку системы. При возникновении данной ситуации текущий узел реактивируется и его основная подсказка выполняется, если это первый запрос повторения, в противном случае выполняется переформулирование подсказки;
- (3) Ситуация «Запрос помощи», когда пользователь не знает, как ответить на вопрос и просит пояснить. В этом случае узел взаимодействия реактивируется и ассоциируемая подсказка помощи выполняется вместо основной подсказки;
- (4) Ситуация «Нет ввода», когда пользователь не подает высказываний.
- (5) Ситуация «Нет соответствия», когда пользователь отвечает, но из его высказывания нельзя извлечь ничего полезного.

В ситуациях (4), (5) узел реактивируется и подсказка «Нет ввода/ Нет соответствия» вставляется перед повтором основной подсказки. Для того чтобы избежать заикливания, необходимо накладывать ограничения на число повторений активации узла.

Помимо перечисленных стратегий локальный менеджер диалога в зависимости от решаемой проблемы может использовать следующие стратегии исправления непонимания:

- общая помощь по использованию системы;
- контекстно-зависимые примеры;
- отказ системы от вопроса и дальнейший переход по диалогу;
- повтор вопроса;
- система просит пользователя повторить, что он сказал;
- система просит перефразировать высказывание;
- система просит сказать громче;
- система просит использовать короткие высказывания;
- начать диалог заново;
- прекратить диалог и повесить трубку.

Использование локального менеджера диалога позволяет системе выявить неинформативные или бессмысленные высказывания на первом этапе диалогового взаимодействия. Очевидно, что эффективность работы локального менеджера будет всецело зависеть от грамотно подобранных стратегий.

### *4. Стратегии глобального менеджера диалога*

Глобальное управление на основе правил выполняется следованием строго заданных дополнительных стратегий:

- Стратегия определения следующего шага взаимодействия;
- Стратегия управления тупиковым диалогом используется в ситуации, где нет решения, соответствующего пользовательскому запросу;
- Стратегия подтверждения позволяет системе проверить достоверность значений, полученных при взаимодействии;
- Стратегия завершения диалога определяет, когда взаимодействие с пользователем должно быть завешено;
- Стратегия устранения несоответствий используется в ситуациях, в которых есть, по крайней мере, два несовместимых значения, полученных от пользователя.

Все эти стратегии используются в работе глобального менеджера диалога независимо от приложения. Наша задача состоит в том, чтобы разработать вышеуказанные стратегии и оптимизировать порядок включения этих стратегий в процесс управления диалогом.

#### 4.1. Стратегия определения следующего шага взаимодействия

Предложенная стратегия состоит из следующих четырех шагов:

1. Извлечение некоторых канонических значений, полученных от пользователя при текущем взаимодействии;
2. Добавление полученных значений к набору ранее извлеченных значений атрибутов слотов фрейма задачи таким образом, чтобы в нем остались только значения совместимые с текущим набором значений;
3. Обобщение атрибутов, для которых имплицитно получены канонические значения.
4. Активизация. Идентифицируется следующий открытый атрибут, т.е. первый атрибут, не имеющий значения, и активизируется ассоциируемое с ним действие.

Рассмотрим пример (1) взаимодействия.

(1) Пример.

**S1:** Чего бы вы хотели?

**U1:** Посмотреть рейсы на Варшаву?

**S2:** Назовите дату, когда вы хотели бы вылететь в Варшаву?

Здесь *S* – система, *U* – пользователь. Стратегия для этого взаимодействия работает следующим образом.

1. Извлечение: из *U1* для фрейма 1 получили «Пункт назначения»= «Варшава», «Пункт отправления» был определен по умолчанию, поскольку справочная система располагается в Санкт-Петербурге.

2. Добавление: Вносим значения, извлеченные в пункте 1 во фрейм 1, как это показано в таблице 6.

Имя слота	Значение слота	Способ подтверждения значения слота	Присоединенная процедура
Пункт отправления	Санкт-Петербург	Не подтвержден	Стратегия подтверждения
Пункт прибытия	Варшава	Не подтвержден	Стратегия подтверждения
Число			
Месяц			
Класс (эконом, бизнес)			

Таблица 3. Добавление во фрейм 1 «Рейс» полученных значений

3. Обобщение: На данный момент во фрейме получены значения атрибутов слотов «Пункт отправления» и «Пункт назначения».

4. Активизация: следующий «открытый» - неизвестный атрибут «Число» (значения атрибута не получены), следовательно, активизируется ассоциируемое с ним действие - задается вопрос *S2*.

#### 4.2. Стратегия подтверждения

Стратегия подтверждения это процедура, используемая для получения подтверждения от пользователя значений, которые были получены системой. Существует три типа подтверждений:

*Явное подтверждение* выполняет подтверждение полученных значений путем задачи прямого вопроса пользователю. Явное подтверждение требует простых ответов типа «Да» или «Нет». Например, когда система

спросила: «Вы сказали, что хотите билет в Санкт-Петербург, правильно?»).

*Скрытое подтверждение* содержит неявный запрос на подтверждение полученной ранее информации, а также запрос новой информации. Это делается путем повторения информации, извлеченной из предыдущего сообщения. При этом ожидается, что агент инстинктивно заметит неправильную информацию и поправит ее или, в случае отсутствия ошибок, сообщит новые данные. Например, необходимо проверить введенные данные о рейсе «Пункт назначения»= «Варшава», «Пункт отправления»= «Санкт-Петербург». Скрытое подтверждение SConf() будет выглядеть следующим образом:

SConf(«Пункт отправления», «Пункт назначения»)=«Когда бы вы хотели вылететь из Петербурга в Москву?»

*Окончательное подтверждение* – по окончании диалога или поддиалога, система просит подтвердить все переменные, полученные в ходе диалога. В нашем случае использование окончательного подтверждения является избыточным и будет только снижать ощущение естественности диалога.

#### 4.3. Стратегия управления тупиковым диалогом

Эта стратегия требуется в случае, если в результате выполнения подзадачи решение не найдено. Для разрешения данной ситуации, мы используем стратегию релаксации:

1. Если в результате обращения к информационной базе данных было получено некоторое число решений, совместимых с полученными значениями, то выдаем все мягкие решения пользователю и просим его выбрать желаемое решение. В противном случае выбираем атрибут, соответствующий не нулевому числу мягких решений.
2. Удаляем значение, ассоциируемое с выбранным атрибутом, и активизируем узел «да/нет», чтобы получить решение пользователя об изменении значения данного атрибута.
3. Если пользователь согласен с решением, то активируется следующее взаимодействие по правилу стандартной активации.
4. Если пользователь отвергает все возможности релаксации, то диалог обрывается.

Рассмотрим процесс релаксации на примере (2):

(2) *Пример.*

S1: *Что бы вы хотели?*

U1: *Посмотреть рейсы в Амстердам на 13 мая.*

*На этом этапе диалога система получила канонические значения «Амстердам», «13» и «Мая». Если система не нашла рейсов на Москву на 13 число, то возникает тупиковая ситуация. Для ее разрешения вначале смягчается поиск по «Числу», и система выдает список других канонических значений пользователю, предлагая выбрать из тех, что есть, либо выдает сообщение:*

:S2: *По вашему запросу не найдено соответствующего решения. Хотите ли посмотреть другие даты, ближайšie к 13 мая?*

*В противном случае система выполняет процесс подтверждения отказа от поиска по прежним параметрам.*

S2: *По вашему запросу не найдено соответствующего решения. Хотите ли пересмотреть дату?*

#### 4.4. Стратегия устранения несоответствий

Эта стратегия нужна в случае, когда пользователь предоставляет несовместимые значения для одного или нескольких атрибутов. В случае возникновения такого несоответствия новое значение используется поверх старого. В случае повторения несоответствия запускается стратегия управления тупиковым диалогом.

Если пользователь вначале определил «Москва» для слота «Пункт назначения» и позже определил «Варшава», тем самым, внося несоответствие. В таких ситуациях используются подсказки несоответствия, например: «Я не уверен какой именно пункт назначения вам нужен : Москва или Варшава. Не могли бы вы повторить?»

В случае нескольких сходных несоответствий, обрабатывается только одно, остальные новые значения удаляются. Используемые подсказки несоответствия генерируются автоматически по шаблонам («Я не уверен какой \$attributeName вы назвали: \$attributeValue1 or...») которые содержат ссылки на используемые атрибуты имени и значения.

#### **4.5. Стратегия завершения диалога**

В ходе диалога эффективнее представить на выбор пользователя найденное ограниченное число решений в виде списка решений, вместо того, чтобы пытаться продолжить диалог для пояснения запроса пользователя и снижения числа решений до единственного. Если число доступных решений существенно мало, то система показывает все возможные решения пользователю и предлагает выбрать, вместо того, чтобы вводить дополнительные критерии выбора. Чтобы применить такой способ, было установлено ограничение на число текущих решений, так называемый порог завершения.

Стратегия завершения соответствует остановке диалога и переключает на явный выбор решения, если текущее число решений меньше или равно порогу завершения. Как только выбор сделан, задача завершается, и диалог обрывается.

### **5. Заключение**

На сегодняшний день нами разработаны и созданы отдельные диалоговые стратегии в виде модулей, подключаемых в ходе взаимодействия. Данные стратегии отработаны на подзадаче представленной фреймом 1 «Рейс». Для развития данной модели на данный момент нами разрабатывается модель скрытого сопровождения диалога с использованием методики Гудвина (Wizard of Oz) [8], которая позволит нам получить достаточное количество реальных диалоговых данных, для построения реальной модели речевого диалога.

#### **Список литературы**

1. Allen J., Byron D., Dzikovska M., Ferguson G., Galescu L., Stent A. “Towards Conversational Human-Computer Interaction“ // AI Magazine M. P.: 2001, V.22, №4, pp. 27-37.
2. Allen .F., Guinn C.I., Horvitz E. Mixed-Initiative Interaction. // IEEE Intelligent Systems. 1999. Vol. 14(5), pp.14-23.
3. Seneff S. and Polifroni J. “Dialogue Management in the Mercury Flight Reservation System“ // Proc. Dialogue Workshop. ANLP-NAACL, Seattle: April 2000, V3, pp.11-16.
4. Pellom B., Ward W., Hansen J., Hacıoglu K., and Zhang J., Yu X., and Pradhan S. “University of Colorado Dialog Systems for Travel and Navigation” // Proceedings of the 2001 Human Language Technology Conference (HLT-2001), San Diego, CA, USA: March 2001. <http://cslr.colorado.edu/~spradhan/publications/hlt2001.pdf>
5. Griol D., Torres F., Hurtado L., Grau S., Garcia F., Sanchis E., Segarra E. A dialog system for the DIHANA Project. // 11-th International Conference SPECOM’2006, St. Petersburg: “Anatolia”, 2006, pp. 131-136.
6. Lamel L., Rosset S., Gauvain J.L., Bennacef S., Garnier-Rizet M., and Prouts B.. The LIMSI ARISE System. // Speech Communication, Aug 2000, 31(4):339-354.
7. Rajman M., Rajman A., Seydoux F., and Trutnev A. Assessing the usability of a dialogue management system designed in the framework of a rapid dialogue prototyping methodology. // First ISCA Tutorial Research Workshop on Auditory Quality of Systems, Herne: Akademie Mont-Cenis, April 2003.
8. Ронжин А.Л., Леонтьева А.Б. Применение техники «Гудвин» для моделирования человеко-машинного взаимодействия Известия вузов. Приборостроение // С-Пб.: 2006. Т. 49, № 11. С. 70-75.