

ПОСТРОЕНИЕ СИСТЕМЫ ВЕДЕНИЯ ДИАЛОГА НА ОСНОВЕ РАСШИРЕНИЯ СЕТЕЙ ПЕТРИ

Э. К. Алгазинов, А. А. Жижелев, А. Р. Нехаев

Воронежский государственный университет

Поступила в редакцию 15.10.2011 г.

Аннотация. В работе представлен подход к организации ведения диалога на семантическом уровне голосовой диалоговой системы. На основе уже существующих модификации сетей Петри, таких как динамические цветные сети, реактивные сети, WF-сети, предлагается оригинальное расширение сетей Петри, учитывающее специфику задачи ведения диалога. В работе приведены примеры использования данного подхода в диалоговой системе ввода объявлений.

Ключевые слова: голосовая диалоговая система, ведение диалога, сети Петри.

Annotation. The paper explores dialog management method for the voice dialog system using extension of Petri nets. The proposed extension is based on well-known modifications of Petri nets, such as dynamically colored nets, reactive nets and WF-nets. An ads receiving dialog system is considered as an example.

Keywords: voice dialog system, dialog management, Petri nets.

ВВЕДЕНИЕ

В последнее время все большую популярность приобретают голосовые системы. Это связано с ростом возможностей вычислительной техники и с совершенствованием средств распознавания и синтеза речи. Для английского языка уже в настоящее время существуют промышленные системы распознавания и синтеза речи. Такие системы могут выполнять голосовые команды, задавать вопросы и понимать ответы на них. Однако они не допускают общения в свободной форме: пользователям требуется точно знать синтаксис и семантику голосовых команд, а также варианты корректных ответов на вопросы. Взаимодействие человека с такой голосовой системой резко отличается от диалога человека с человеком в рамках той же предметной области, поскольку диалоговые возможности современных голосовых диалоговых систем являются достаточно скромными. Настоящая работа посвящена исследованию вопросов ведения диалога в голосовой диалоговой системе.

Диалог – это функциональная разновидность языка, реализующаяся в процессе непосредственного общения между собеседниками и состоящая из последовательного чередования реплик. К основным языковым особенностям

диалога относятся: обилие вопросительных и побудительных предложений, наличие повторов и переспросов, синтаксическая неполнота реплик, компенсируемая за счёт предыдущего высказывания.

Под **голосовой диалоговой системой** (ГДС) мы понимаем информационную систему с элементами искусственного интеллекта, участвующую в диалоге в качестве одной из сторон. Будем считать, что **ведение диалога** заключается в следующих способностях системы:

- 1) понимать обращенные к ней реплики,
- 2) формировать ответные реплики в соответствии с логикой и динамикой диалога,
- 3) позволять пользователю достигать своих прагматических целей через диалог.

Последним пунктом мы ограничиваем рассмотрение голосовыми диалоговыми системами оказания услуг (самообслуживания). Все названные пункты тесно связаны между собой: каждый следующий пункт невозможен без успешного выполнения предыдущего.

Ключевой в приведенном списке является задача понимания: без ее успешного решения невозможно приступить к выполнению остальных задач. Проблемы, связанные с пониманием реплик пользователя, можно разделить на две группы: проблемы, обусловленные уровнем развития современных средств распознавания русской речи, и проблемы, обусловленные от-

личиями диалога от монолога и от письменной речи.

К первой группе проблем можно отнести:

- низкое качество распознавания, которое существенно зависит от микрофона и дикции говорящего. В среднем доля ошибок в словах (WER) колеблется от 10 до 30%;

- невозможность восприятия эмоций говорящего, которые проявляются в речи главным образом через интонацию и неречевые звуки (смех, вздохи, хмыканье и др.)

Ко второй группе (подробнее см. [1]):

- эллипсисы – пропуски того, о чем можно догадаться из контекста;

- синтаксическая неполнота (особенно при вопросно-ответном характере диалога): крайне лаконичные, односложные ответы;

- ошибки неподготовленной (спонтанной) речи и связанные с ними исправления и уточнения;

- перебивания;

- повторы [2];

- отсутствие четкой сегментации мыслей (в тексте это пунктуационно выделенные предложения, параграфы).

Вместе с тем, форма диалога имеет ряд преимуществ по сравнению с письменной речью или монологом, благодаря которым подсистема ведения диалога, на наш взгляд, имеет возможность нивелировать описанные выше проблемы:

- возможность запросить (в том числе косвенными методами) у пользователя подтверждение того, что его реплика была понята верно;

- возможность попросить пользователя уточнить и переформулировать;

- возможность задать уточняющий вопрос.

Применяемый нами способ разбора и извлечения семантики из распознанного текста подробно описан в [3], в настоящей работе мы ограничим рассмотрение семантическим уровнем диалога. Для его реализации требуется как минимум два компонента: семантическая модель предметной области и сценарий диалога. Под **сценарием диалога** мы подразумеваем заранее подготовленное описание этапов диалога, ожидаемых реплик пользователя и правил реагирования системы на эти реплики. В разделе «Модель предметной области» мы дадим краткую характеристику и опишем особенности использования фреймовой модели. В разделе «Описание сценария диалога» мы предложим способ описания сценариев на семантическом уровне

на основе расширения сетей Петри. В разделе «Примеры использования» мы приведем сценарии для некоторых типовых диалоговых ситуаций.

В качестве практической иллюстрации мы будем использовать фрагменты диалогов с реализованным авторами прототипом голосовой диалоговой системы приема объявлений о продаже мобильных телефонов. Пользователь называет производителя и модель телефона, а также цену продажи. Для размещения объявления требуется контактный телефон пользователя. Цель системы – получить в ходе диалога с пользователем все перечисленные выше данные и опубликовать по ним объявление в газете.

МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ

В наиболее распространенных голосовых диалоговых системах (VoiceXML, Nuance, RavenClaw и др.) для описания семантики предметной области используются те или иные реализации фреймовой модели представления знаний. Рассмотрим основные черты реализации фреймовой модели, применяемой в нашей системе.

Под **фреймом-экземпляром** будем понимать формализованное описание какого-либо объекта или явления, свойства которого представлены ассоциативным набором слотов и значений. **Фрейм-прототип** – описание класса фреймов-экземпляров (метаописание), состоящего из ассоциативного набора слотов, предусмотренных для данного класса фреймов-экземпляров, и соответствующих им доменов значений. И фрейм-экземпляры, и фреймы-прототипы имеют собственные идентификаторы. **Фреймы-шаблоны** похожи на фреймы-экземпляры, но их слоты заполнены частично, и они не имеют идентификаторов. Такие фреймы отражают имеющиеся в распоряжении системы обрывочные знания и предназначены для итеративного накопления знаний, а также для поиска полных фреймов-экземпляров по их неполному описанию.

При таком подходе предмет диалога (команда, товар и т.д.) представляется в виде фрейм-шаблона, который заполняется по ходу общения с пользователем. Фреймовая модель позволяет избежать ограничений на порядок заполнения, т.е. делает диалоговый сценарий более гибким, позволяя пользователю говорить о том, что он сам считает важным.

Упрощенный пример фреймовой модели для объявления о продаже мобильного телефона показан на рис. 1. Жирным шрифтом отмечены слоты, обязательные для ввода.

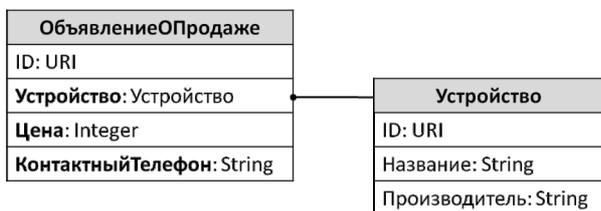


Рис. 1. Пример фреймовой модели

Пользователь может заполнять слоты в произвольном порядке, однако, если он не указал значения для какого-либо из обязательных слотов, система сама спрашивает о нем (здесь и далее используются обозначения «П» – пользователь, «А» – автомат):

П: Продаю телефон нокия тридцать два тридцать.

А: За сколько?

П: За две тысячи.

А: Нокия тридцать два тридцать за две тысячи рублей.

Результатом поиска в базе знаний является множество фреймов-экземпляров, удовлетворяющих заданному фрейму-шаблону. Для более удобной последующей алгоритмической обработки это множество фреймов-экземпляров «сливается» в один фрейм-экземпляр, у которого вместо одного значения каждому слоту соответствует множество уникальных значений для данного слота, которые встретились в исходном множестве фреймов-экземпляров. Такую структуру мы будем называть **нечетким фреймом-экземпляром** (fuzzy frame instance). Представление результатов поиска в таком виде позволяет в случае нескольких найденных вариантов задать уточняющий вопрос с альтернативами:

П: Продам телефон Самсунг.

А: Гэлэкси пятьдесят восемь тридцать или девятьсот восемьдесят дуос?

Не менее продуктивным оказывается использование нечеткого фрейма-экземпляра при поиске противоречий внутри пользовательского высказывания:

П: Я хочу продать Самсунг тридцать два тридцать.

А: Самсунг или Нокия тридцать два тридцать?

Таким образом, фреймовая модель позволяет решить проблемы, связанные с эллипсисами, синтаксической неполнотой и спонтанным характером речи, поскольку не накладывает жестких ограничений на порядок и формат заполнения слотов. Кроме того, она позволяет активно пользоваться преимуществами диалоговой формы речи: формулировать вопросы, находить и исправлять противоречия в высказываниях пользователя. Фреймовая модель позволяет предсказывать содержимое реплик пользователя (что он должен сказать, чтобы заполнить следующий слот, ответить на вопрос системы и т.д.), а следовательно позволяет улучшить распознавание, динамически корректируя языковую модель [4].

ОПИСАНИЕ СЦЕНАРИЯ ДИАЛОГА

Фреймовая модель хорошо подходит для голосового «заполнения форм» в диалоговом режиме, при возможности непосредственно распознать значения слотов в речи пользователя. Однако она не подходит для описания алгоритмов преобразования семантики в случае, когда значение слота само является сложным объектом и не может быть получено сразу и в явном виде. Несмотря на возможность находить и исправлять путем дополнительных вопросов противоречия, с помощью фреймовой модели достаточно сложно организовать работу с уточнениями и исправлениями, исходящими от самого пользователя. Если фрейм, заполняемый в ходе диалога, имеет даже несколько слотов со сложной внутренней структурой, учет и регуляция всех возможных сценариев диалога в рамках фреймовой модели (с привлечением метаданных) затруднительны. Это означает, что комбинация фреймов, для каждого из которых отсутствует жесткий порядок и формат заполнения слотов, может привести к непредсказуемому результату.

В частности, в задаче ввода объявлений, которую мы выбрали в качестве примера, с помощью фреймовой модели затруднительно организовать ввод контактного телефона пользователя. Рассмотрим, как пользователь мог бы продиктовать оператору-человеку номер своего телефона, например, 8-800-395-55-10:

Восемь восемьсот триста девяносто пять пятьдесят пять десять

Восемь восемьсот триста девяносто пять пятьсот пятьдесят один ноль

Восемь восемьдесят ноль три девяносто пять
 пятьдесят пять десять

Восемь восемьсот три девять три пятерки
 один ноль

...

Обычно диктовка длинного телефона разби-
 вается на несколько частей с ожиданием после
 каждой из них подтверждения от оператора:

П: Восемь восемьсот

О: Угу

П: Триста девяносто пять

О: Триста?

П: Да, триста девяносто пять. Пятьдесят
 пять. Записал?

О: Ага

П: Десять

О: Восемь восемьсот триста девяносто пять
 пятьдесят пять десять. Правильно?

П: Да.

Чтобы сохранить преимущества фреймовой
 модели работы с семантикой и устранить недо-
 статки, предлагается использовать ее в сочетании
 с описанием алгоритмических преобразований
 семантики расширенными сетями Петри.

Сеть Петри представляет собой двудольный
 ориентированный граф, состоящий из вершин
 двух типов — позиций и переходов, соединён-
 ных между собой дугами. Вершины одного
 типа не могут быть соединены непосредствен-
 но. В позициях могут размещаться метки (мар-
 керы), способные перемещаться по сети [5].
 Классическая сеть Петри задается как набор

$$N = (P, T, A),$$

где P — множество позиций; T — множество пе-
 реходов, $P \cap T = \emptyset$; $A \subseteq (P \times T) \cup (T \times P)$ —
 множество дуг.

Для моделирования диалоговых сценариев
 мы возьмем за основу расширенный вариант
 сетей Петри, а также введем ряд специфических
 для нашей задачи правил и ограничений. Рас-
 ширяя сеть Петри, мы будем опираться на уже
 существующие их модификации, такие как
 динамические цветные сети [6], реактивные
 сети [7], WF-сети [8].

Зададим расширенную сеть Петри как на-
 бор

$$N = (P, T, A, S, G, F),$$

где S — множество цветов (типов) меток;
 $G = \{(a, c) | a \in A, c : S \rightarrow \{0, 1\}\}$ — множество сто-
 рожевых условий на дуги, позволяющих про-
 хождение по этим дугам только меток опреде-

ленных цветов (в [6] для той же цели вводится
 множество сторожевых переходов — guard tran-
 sitions); $F = \{(t, f) | t \in T, f : S \rightarrow (\{0, 1\}, S)\}$ —
 множество пар переходов и функций, описыва-
 ющих их срабатывание. Для каждого сработав-
 шего перехода соответствующая функция воз-
 вращает количество (0 или 1) и цвет меток,
 получающихся при срабатывании.

Маркировка сети задается как функция,
 возвращающая число меток данного цвета в
 данной позиции $M : P \times S \rightarrow N$.

Для взаимодействия с внешним миром в
 сетях Петри допускается (см. [5]) использо-
 вание входных и выходных позиций сети таких,
 что во входных позициях P_{IN} могут появляться
 метки (входящие данные из внешних систем),
 а из исходящих P_{OUT} — пропадать (исходящие
 во внешние системы данные). Входные и вы-
 ходные позиции будем называть внешними и
 обозначать P_E , а все остальные — внутренними
 P_I . Введем также служебные позиции — старто-
 вую и финальную.

$$P = P_I \cup P_E \cup \{p_{start}, p_{final}\}, \quad (1)$$

$$P_E = P_{IN} \cup P_{OUT}.$$

Аналогично переходы, имеющие хотя бы
 одну входящую дугу из внешней позиции, будем
 называть внешними, а остальные переходы —
 внутренними:

$$T = T_I \cup T_E. \quad (2)$$

Опираясь на эти определения, сформулиру-
 ем следующее ограничение на выполнение на-
 шей сети: сначала запускаются разрешенные
 внутренние переходы сети, а при отсутствии
 разрешенных внутренних — разрешенные вне-
 шние.

$$M \rightarrow M' \Leftrightarrow \exists t_i \in T_I : M \xrightarrow{t_i} M' \vee$$

$$\vee \left((\exists t_e \in T_E : M \xrightarrow{t_e} M') \wedge \right. \quad (3)$$

$$\left. \wedge (\nexists t_i \in T_I : M \xrightarrow{t_i} M') \right).$$

Определим логическую функцию, проверя-
 ющую наличие меток в позиции:

$$e(p) = \begin{cases} 1, & \text{если } \sum_{s \in S} M(p, s) > 0, \\ 0, & \text{если } \sum_{s \in S} M(p, s) = 0. \end{cases} \quad (4)$$

Как уже отмечалось выше, метки в сети мо-
 гут появляться извне в специальных входных
 позициях сети. Для этого используется внешнее
 по отношению к сети правило, которое сраба-
 тывает, когда в сети не остается разрешенных

переходов. Это алгоритм одновременно помещает метки во все входные позиции сети из множества D . Обозначим множество позиций, исходящие дуги из которых ведут в переход t , как $P_{IN}(t)$. Формирование множества активных входных позиций сети выполняется по формуле:

$$D = \left(p_i \mid p_i \in P_{IN} \wedge (\exists t_e \in T_E : p_i \in \cdot t_e) \wedge (\exists p_j \in \cdot t_e : e(p_j)) \right) \quad (5)$$

С точки зрения моделирования диалога, позиции – это диалоговые состояния, в которых может находиться система. Если метки присутствуют более чем в одной позиции, то диалоговая система одновременно находится в нескольких состояниях. Метки – это фреймы-экземпляры, содержащие семантическую информацию. Цвет метки – это идентификатор фрейма-образца, соответствующего данному фрейму-экземпляру. Дуги могут быть помечены цветом меток, для которых разрешено прохождение, этим обеспечивается маршрутизация семантики в сети. Переходы отражают семантические преобразования, которые выполняются системой. Функция срабатывания, связанная с переходом, задает алгоритм семантического преобразования.

Стартовая позиция используется для первоначальной инициализации сети, а также для передачи семантических данных во вложенные сети. Попадание метки в финальную позицию сети сигнализирует о том, что фрагмент диалога, который описывается данной сетью, был завершен.

Во входные позиции сети по ходу диалога помещается семантическая информация, извлекаемая из реплик пользователя. Введенное нами ограничение (3) гарантирует, что семантика, пришедшая раньше, будет полностью обработана до прихода новой семантики. Необходимость ввода правила заполнения входных позиций с выбором только активных (5) продиктована стремлением избежать активации нерелевантных участков сценария диалога и придать большую устойчивость к неверно распознанным фразам.

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

Рассмотрим возможность применения предложенного расширения сетей Петри на примере предметной области подачи объявлений о продаже телефонов. Общий диалоговый сценарий ввода объявления включает в себя различ-

ные диалоговые ситуации, однако он слишком громоздкий для рассмотрения его целиком в качестве иллюстрации. Поэтому мы выделим из него упрощенные мини-сценарии, демонстрирующие применение нашего подхода для наиболее характерных диалоговых ситуаций.

Начнем с самого простого – необходимости отвечать на реплики пользователя, такие как «алло?», «привет!», «спасибо!» и т.п. Диалоговый сценарий, соответствующий такой ситуации, представлен на рис. 2.

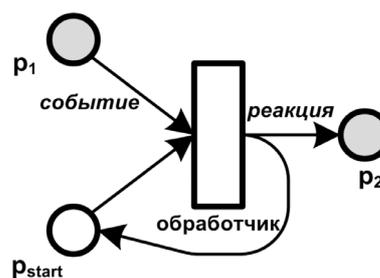


Рис. 2. Сценарий обработки события

Сеть инициализируется маркировкой с единственной меткой в стартовой позиции p_{start} . Переход остается закрытым до тех пор, пока нет метки во входящей позиции p_1 (внешние позиции на этом и других рисунках обозначены серым цветом). Метка в этой позиции появляется при поступлении извне семантического значения реплики пользователя или возникновении диалогового события. Семантическими значениями реплики в данном случае могут быть, например, приветствие, проверка связи, прощание и другие подобные значения реплик, требующих простого ответа. Примеры диалоговых событий: затянувшаяся пауза в диалоге, слишком тихий ответ пользователя и др. И значения реплик, и диалоговые события единообразно представляются в виде фреймов-экземпляров, их тип (т.е. соответствующий идентификатор фрейма-образца) проверяется на соответствие сторожевому условию дуги. В случае если идентификатор фрейма-образца (в терминологии цветных сетей Петри – цвет метки) подошел, переход становится разрешенным и вызывается обработчик, связанный с этим переходом. Обработчик выдает семантику ответа, которая поступает в выходную позицию p_2 и в стартовую позицию p_{start} . Семантика ответа из выходной позиции используется для синтеза ответной реплики пользователю. Копирование

семантики ответа в стартовую позицию позволяет вновь активировать сеть для обработки последующих событий того же типа, если они возникнут.

Пример использования сети на рис.2 для ответа пользователю, проверяющему связь:

П: Алло?

А: Да, я вас слушаю.

Пример реакции на диалоговое событие длительного молчания пользователя – теперь системе самой требуется проверить связь:

П: <молчание>

А: Алло? Я вас не слышу.

Теперь рассмотрим более сложную сеть, предназначенную для заполнения слотов объявления (рис. 3).

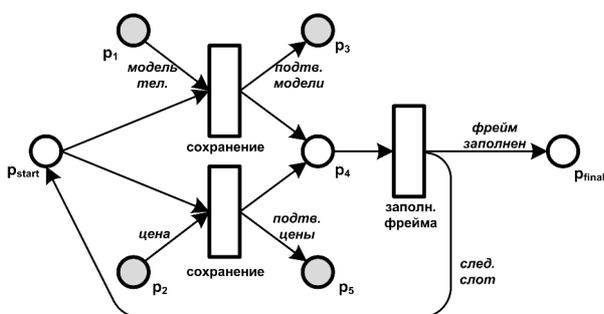


Рис. 3. Сценарий заполнения объявления

Эта сеть имеет две входные позиции – p_1 и p_2 , для модели телефона и цены соответственно. Пользователь может называть значения этих слотов в любом порядке, после ввода каждого значения, система выдает соответствующее подтверждение, озвучивая введенное пользователем значение (выходные позиции p_3 и p_5). Это дает пользователю возможность скорректировать ошибочно названное им или неправильно понятое системой значение. Сеть будет активироваться заново до тех пор, пока фрейм объявления не будет заполнен до конца.

П: Нокия тридцать два тридцать за три тысячи рублей.

А: Найден телефон Нокия Тридцать два тридцать.

А: Цена три тысячи рублей.

Комбинируя вместе сеть для заполнения слотов и обработчик события молчания пользователя (описан выше), можно реализовать напоминание о следующем незаполненном слоте:

П: Мой телефон Самсунг гэлэкси.

А: Найден телефон Самсунг Гэлэкси пятьдесят восемь тридцать.

П: <молчание>

А: Назовите, пожалуйста, цену.

П: Цена две тысячи семьсот рублей.

Следует отметить, что для заполнения фрейма со сколь угодно большим количеством слотов можно построить сеть, аналогичную изображенной на рис. 3 при условии, что значения всех слотов будут из разных доменов. В нашем случае модель телефона – это справочное значение, цена – это число. Если же два и более слотов принимают значения из одного и того же домена, потребуется более сложная структура сети для их различения.

Как мы уже отмечали выше, важными для нас свойствами фреймовой модели являются возможность формулирования альтернативных вопросов и разрешения противоречий. На рис. 4 представлена сеть для работы с этими возможностями.

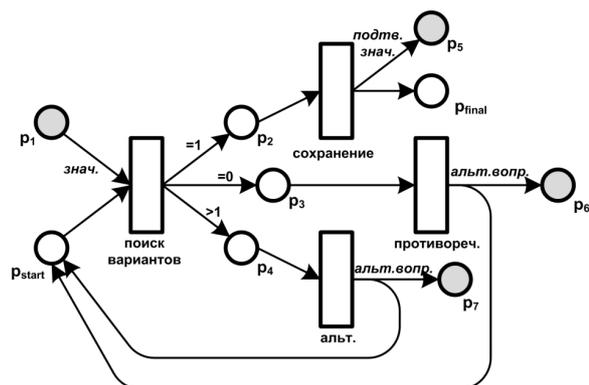


Рис. 4. Сценарий разрешения неоднозначности

Если поиск вариантов вернул один результат, искомое значение найдено. Если более одного результата, формулируется вопрос к пользователю:

П: У меня Элджи.

А: Элджи Джи экс двести или Элджи Кей пи пятьсот.

Если ни одного результата не было найдено, значит высказывание содержало противоречие, которое пользователь должен разрешить:

П: Сони эриксон гэлэкси.

А: Сониэриксон двести пять или Самсунг Гэлэкси пятьдесят восемь тридцать?

Для ввода последовательности чисел используется сеть, представленная на рис. 5. Она сохраняет называемые пользователем чис-

ла, пока пользователь не скажет, что ввод закончен. После этого система воспроизводит всю названную последовательность чисел для подтверждения.

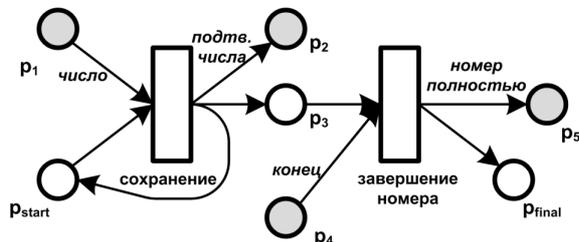


Рис. 5. Сценарий ввода телефона

Пример ввода контактного телефона с использованием такой сети:

П: Восемь. Девятьсот сорок. Триста двадцать шесть. Пятьдесят пять. Восемьдесят семь. Всё.

А: Вы назвали номер: восемь девятьсот сорок триста двадцать шесть пятьдесят пять восемьдесят семь.

Если пользователь делает паузы, система воспроизводит фрагмент последовательности, названный в последней реплике:

П: Восемь.

А: Восемь.

П: Девятьсот сорок.

А: Девятьсот сорок.

П: Триста двадцать шесть.

А: Триста двадцать шесть.

П: Я закончил.

А: Вы назвали номер: восемь девятьсот сорок триста двадцать шесть.

Поскольку распознавание чисел не всегда выдает правильные результаты, пользователю необходимо предоставить возможность исправить неправильно распознанные числа. Сеть для ввода исправления фрагмента последовательности представлена на рис. 6.

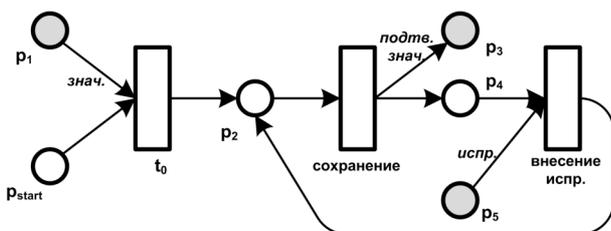


Рис. 6. Сценарий коррекции значения

По умолчанию исправляется последний введенный фрагмент последовательности:

П: Двести тридцать семь... Нет, триста двадцать семь восемьдесят пять.

А: Вы назвали номер: триста двадцать семь восемьдесят пять.

Однако допускается коррекция произвольного фрагмента последовательности, при условии, что он указан явно:

П: Двести тридцать семь... Нет, не двести тридцать, а триста двадцать.

А: Вы назвали номер: триста двадцать семь.

ЗАКЛЮЧЕНИЕ

Благодаря тому, что сети Петри изначально разработаны для моделирования параллельных и асинхронных процессов, они позволяют учесть особенности диалога: асинхронный обмен репликами и параллельное развитие нескольких тем. Движение семантики по сети Петри в виде меток хорошо отражает логику обработки семантики. Формирование составных сетей Петри осуществляется достаточно просто (см. [5]), поэтому диалоговые сценарии на их базе обладают полезными при росте числа поддерживаемых диалоговых ситуаций свойствами модульности и масштабируемости. При реализации системы ведения диалога неизбежно возникает несколько взаимосвязанных семантических слоев, каждый из которых может иметь свой набор входов и выходов. Введенное в данной работе расширение сетей Петри предоставляет большие возможности для регуляции процесса обработки семантики, позволяя формировать и выполнять достаточно сложные диалоговые сценарии.

Предложенный в данной работе подход расширяет возможности ведения диалога в голосовой диалоговой системе, способствуя повышению естественности диалогов.

СПИСОК ЛИТЕРАТУРЫ

1. Якубинский Л.П. О диалогической речи // Л.П.Якубинский. Избранные работы. Язык и его функционирование. – М.: Наука. 1986. – С. 17–58.
2. Шведова Н.Ю. Очерки по синтаксису русской разговорной речи. М.: Изд-во АН СССР, 1960. – 377 с.
3. Алгазинов Э.К., Жижелев А.А., Нехаев А.Р. Анализ распознанной речи с использованием шаблонов // Кибернетика и высокие технологии XXI века: материалы XII Международной научно-технической конференции – Воронеж: Издательско-полиграфический центр ВГУ, 2011. – Том 1. – С. 76–84.

4. *Jurafsky D., Martin, J.H.* Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics, 2nd ed. – USA, New Jersey: Prentice-Hall, 2009. – 988 с.

5. *Питерсон Дж.* Теория сетей Петри и моделирование систем: Пер. с англ. – М.: Мир, 1984. – 264 с.

6. *Everdij M.H.C.* Compositional specification of a multi-agent system by Dynamically Coloured Petri

Nets / Klompstra M.B., Blom H.A.P., Klein Obbink B. // HYBRIDGE WP 9.2 report, National Aerospace Laboratory, 2004, – 31 с.

7. *Eshuis R., Dehnert J.* Reactive Petri nets for workflow modeling // In Proceedings of ICATPN, 2003. – С. 296–315.

8. *Van der Aalst W.M.P.* The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 8(1), 1998 – С. 21–66.

Алгазинов Эдуард Константинович – д. ф.-м. н., профессор кафедры Информационных систем ФКН, ВГУ. Тел. (4732)208-724. Email: algazinov@cs.vsu.ru

Algazinov E. K. – doctor of Physics-math. Sciences, Professor of the dept. of the Information Systems, Voronezh State University, Tel. (4732)208-724

Жижелев Арсений Александрович – к.э.н., ассистент кафедры Информационных систем ФКН, ВГУ. Тел. (4732)208-724. Email: zhizhelev@cs.vsu.ru

Zhizhelev A. A. – candidate of economic sciences, assistant of dept. of the Information Systems, Voronezh State University, Tel. (4732)208-724

Нехаев Антон Романович – ассистент кафедры Информационных систем ФКН, ВГУ. Тел. (4732)208-724. Email: nehaev@gmail.com

Nekhaev A. R. – assistant of dept. of the Information Systems, Voronezh State University, Tel. (4732)208-724