Парадигмы генерации ЕЯ текстов в инструментальной среде DEMLinG

М. В. Болдасов
Московский Государственный Университет им. М.В. Ломоносова, факультет ВМиК, каф. АЯ

Ключевые слова: автоматическая обработка текста, генерация текста на ЕЯ, многоязыковая генерация, интерлингвистическое представление, грамматика, этапы генерации, запрос к базе данных

В статье описывается идеология системы создания, поддержки и выполнения грамматик многоязыковой генерации DEMLinG. Рассматриваются основные достижения мировой науки в области построения генераторов, и излагаются особенности применения этих принципов в системе DEMLinG. В качестве примера реализации генераторов в рамках данной системы рассматривается семейство генераторов QGen, решающих задачу контроля пользователя за правильностью понимания ЕЯ интерфейсом к БД системы InBASE запроса к базе данных, сформулированного пользователем.

1. Введение

Задача Естественно Языковой Генерации (ЕЯГ) (Natural Language Generation (NLG)) – предоставить компьютерным системам средство взаимодействия с пользователем на ЕЯ. Такое средство может быть оценено как естественное требование к информационным системам, учитывая возрастающее число сложных компьютерных систем, нуждающихся в взаимодействии с пользователем, которыми все чаще становятся неспециалисты в информационных технологиях. NLG часто используются также для представления информации пользователям или для частичной автоматизации создания типовой документации, такой как техническая документация, справочной сообщений и документации и т.д. Поэтому понятен постоянно растущий интерес к системам генерации и большое число исследований за рубежом в этой области.

Система DEMLinG (Development Environment for MultiLingual Generators), начала разрабатываться в процессе решения задачи генерации запроса из его OQL-like представления в системе InBASE (http://www.inbase.artint.ru/) [2,3]. В данной статье она рассматривается как универсальная среда для разработки, поддержки и выполнения грамматик многоязыковой генерации. DEMLinG поддерживает создание генераторов, работающих в узких предметных областях с ограниченными лексиконом, грамматическими и семантическими вариациями.

Примером применения системы DEMLinG является семейство генераторов QGen [1,2], решающее вопрос генерации ЕЯ текста запроса к базам данных по его SQL-like представлению. Генератор QGen предназначен для интеграции в ЕЯ интерфейс к базам данных InBASE, в которой он обеспечит контроль пользователя за правильностью понимания сформулированного им запроса к базе данных.

Настоящая статья описывает идеологию создания генераторов, заложенную в разработанной системе. Работа состоит из двух разделов. В разделе 2 выделяются характерные особенности процесса генерации и дается краткий обзор сегодняшнего

состояния дел в области построения генераторов. В разделе 3 рассматриваются особенности создания генераторов в системе DEMLinG и отражение основных тенденций построения процесса генерации в подходах, применяемых в системе DEMLinG.

2. Основные особенности MLG подхода

Системы генерации обычно обрабатывают некоторое нелингвистическое представление информации, преобразуя ее в текст на ЕЯ. При обработке обычно используются знания о языке и о Модели Предметной Области (МПО). На выходе системы генерации – текст, который пользователь может читать, содержащий, возможно, дополнительное форматирование и графические вставки.

В статье рассматриваются системы многоязыковой генерации (МЯГ) текста на Естественном Языке (ЕЯ). МЯГ - достаточно новая область в направлении генерации. Первая коммерческая система МЯГ появилась а Канаде в начале 90'ых90'ых и предназначена для генерации морских сводок погоды на английском и французском языках. Многоязыковая генерация рассматривается как развитие сразу двух областей компьютерной лингвистики: Машинного Перевода (МТ) и ЕЯГ. Отсюда и некоторые отличия в подходах к построению систем МЯГ и ЕЯГ. Перечислим основные особенности МЯГ подхода.

Для многоязыковой генерации характерно входное интерлингвистическое представление, основанное на инвариантах языка, т.е. структурах, интерпретируемых в различных языках. Это требование продиктовано необходимостью генерировать тексты сразу на нескольких языках из одного входного представления. Поэтому входное представление для систем многоязыковой генерации должно быть максимально абстрагировано от особенностей отдельного языка. Идея использовать интерлингвистическое представление была заимствована из систем МТ (interlingua based systems) [4]. Такие системы используют код Interlingua как промежуточный носитель информации в процессе генерации.

Языковые знания (в дальнейшем также именуемые грамматикой), представлены словарем и правилами, которые соединяют слова и словосочетания синтаксическими и семантическими связями во словосочетания и предложения. Такой выбор обусловлен адекватностью предлагаемой модели описываемым лингвистическим феноменам.

Правила состоят в общем случае из трех частей: образец, дополнительные условия применения и действие. Образец позволяет соотнести объекты, используемые правилом с обрабатываемой структурой. Условия описывают дополнительные требования, определенные контекстом, которые должны быть удовлетворены при сопоставлении. Действие описывает возможные преобразования части входной структуры, соотнесенной с образцом.

Словарь перечисляет соответствия сущностей структуры содержания лингвистическим ресурсам, таким как содержательные слова и грамматические связи. Системы отображения структур имеют выбор между использованием первой подходящей подстановки, или осуществлением поиска подстановки, максимизирующей некий критерий оптимальности (например, длина лексического эквивалента) [5]. Проблема лексикализации –

основная проблема МЯГ, в связи с явлением языкового отклонения – разницы между языковыми понятиями в разных ЕЯ.

В последнее время в моделировании ЕЯ грамматик сформировалась тенденция описания языковых структур через описание ограничений (licensing rules) на сформированность (well-formedness) частей выражения [6]. Одно из обязательных составляющих грамматик ограничений (или унификационных грамматик) – это формальное описание грамматических единиц (слов, словосочетаний, предложений) с помощью наборов пар атрибут-значение, называемых свойствами. Наборы свойств могут образовывать вложенные структуры или быть недоопределенными. Формализмы, основанные на ограничениях, обычно поддерживают операцию объединения и проверки грамматической информации, называемую *унификацией*[6,7]. В генерации механизм унификации используется для проверки и поглощения свойств данными из лингвистических ресурсов.

Сила унификационного подхода заключается в предложенном им механизме проектирования грамматик. Использование этого подхода сильно сокращает временные затраты по созданию широкомасштабных грамматик. Так, например, в то время как создание больших аннотированных грамматик структуры фразы занимает в среднем 8 - 12 лет, создание унификационных грамматик требует временых затрат порядка четырех лет (CLE [21], TDL [19]). Почти все современные исследования, включающие разработку грамматики, используют унификационный механизм.

Метод свойств развивает идею каскадного метода (рекурсивные подстановки обобщенных шаблонов для формирования дерева разбора) заменой шаблонов как единиц представления на классы - типизированные сущности, обладающие набором свойств (пар тип-значение). Метод свойств предполагает, что любой параметр текста, обладающий способностью изменяться, объявляется свойством, а конкретные значения, определяющие изменения, называются значениями свойств. Свойства отображают такие параметры как варианты порядка слов, время, наклонение, тип предложения и т.д. Процесс генерации представляет собой последовательное накопление свойств в классах, пока генерируемая структура классов не сможет считаться сформированной (ограничение на «well-formedness» - в классах накоплены наборы свойств, которые полностью определяют текст). Затем, накопленные свойства переводятся в результирующий текст.

Некоторые подходы используют систему наследования типов (HPSG [17], TSF [18], TDL [19], ALE [20]). Группы свойств типизируются, типы частично упорядочиваются в древесную структуру. Типы иерархически определяют для каждого типа, из которого наследуются остальные типы свойств, какие типы и значения может принимать это свойство, и какие другие типы могут сочетаться с ним посредством унификации. Если система свойств допускает сложные свойства, рекурсивное вложение свойств может быть ограничено рекурсивным определением типа. В принципе, любая грамматика может быть описана через рекурсивное определение типов. В своей крайности этот подход применяется для полного лингвистического вывода (анализ и генерация) в системах HPSG [17] и TSF [18].

В большинстве приложений языковой технологии грамматика, кодируются отдельно от обрабатывающего компонента. Такой подход продиктован удобством декларативного задания знаний с точки зрения адаптивности строимой системы и переносимости создаваемых грамматических ресурсов между различными вычислительными моделями, а также возможностью совместного использования одних грамматических ресурсов анализатором и генератором текстов на ЕЯ. Поэтому подход отделения лингвистических

знаний от вычислительного элемента и их написание в декларативной унификационной форме присущ большинству современных систем генерации.

Основная особенность алгоритма генерации – это разделение языково-зависимой и языково-независимой частей алгоритма генерации. Такое требование необходимо для сокращения временных затрат на добавление нового языка, поддерживаемого построенным генератором. Модульность реализации также улучшает адаптивность системы к возможным изменениям в предметной области и языке, и поэтому присуща большинству современных систем генерации.

Еще одна особенность построения систем MLG – это требование конвейерного представления процесса генерации [5]. Это требование отчасти продиктовано предыдущим и означает последовательное выполнение фаз генерации процессом, возможно с редкой обратной связью (feedback). Взаимная интеграция различных этапов генерации усложняет понятность процесса генерации и механизм его контроля. Как результат – усложнение процесса создания ресурсов генератора. Backtracking необходим только в случае локальной неопределенности как отсутствия глобальной информации [8]. В процессе генерации обычно выделяются четыре фазы [5, 9, 10]:

- Макропланорование (macroplanning). Определяет содержание и структуру документа состоит из двух подпроцессов, работающих в тесном взаимодействии: определение глубинного содержания (deep content determination) и структурирование документа. Первый подпроцесс определяет, какая информация будет участвовать в генерируемом тексте. Процесс определения глубинного содержания независим от языка, на котором выражается генерируемый текст, но более зависим от прикладной задачи, чем остальные этапы. Второй подпроцесс принимает решения, как и в каком порядке выделенные порции информации должны быть сгруппированы в документе.
- Микропланирование (microplanning). Решает, какие слова, грамматические структуры и т.д. будут использоваться для выражения на ЕЯ содержания сформированного на предыдущем этапе. Механизм сопоставления основывается на использовании словаря [5], который перечисляет соответствия сущностей структуры содержания лингвистическим ресурсам, таким как содержательные слова и грамматические связи. На этапе микропланирования также решаются такие задачи как:
  - Выделение топиков для оформления гипертекста на основе построенной структуры содержания [24]
  - Установка абстрактной грамматической структуры высказывания.
  - Поддержка стилевых ограничений (обеспечивается, например, в системе Drafter [11]) - изменение абстрактной грамматической структуры высказывания в зависимости заданного стиля текста (например, некоторые издательства явно оговаривают ограничения на использование грамматических конструкций и лексики).
  - Агрегирование – разбиение структуры содержания на последовательность структур предложений [12].
- Поверхностная реализация (surface generation). Переводит абстрактное представление, построенное на предыдущем этапе, в текст на ЕЯ. Реализует порядок слов, осуществляет морфологическое согласование между членами грамматических групп, вставляет вспомогательные средства для выражения выбранных грамматических групп (например,

слова, указывающие на время предложения), выбирает морфологическую форму для содержательных слов, адекватную сделанным грамматическим выборам.

- Форматирование (структурная реализация). Окончательный сбор предложений из иерархической структуры в текст. Простановка заглавных букв, знаков разделения предложений, абзацев и других структурных составляющих текста, а также оформление гипертекстовых ссылок и прочих знаков форматирования текста.

Характерной особенностью многоязыковой генерации является выделение промежуточных представлений между перечисленными этапами [12]. Так, на выходе из блока макропланирования образуется спецификация смыслового содержания генерируемого текста – семантическая сеть. Примитивы этой сети имеют концептуальную природу, а не лингвистическую (обычно они представлены объектами МПО), что обеспечивает языковую независимость выходного представления. Результатом работы блока микропланирования является абстрактное лингвистическое представление, включающее полнозначные слова и семантические и глубинно-синтаксические связи. Например, SPL-представление [13], формализмы задания глубинного синтаксиса, основанные на функциональной унификации или теории Смысл-Текст[25]. В общем случае выходная структура этапа микропланирования может иметь один из следующих видов [9]:

- орфографическая строка – готовый фрагмент текста
- готовый текст – нужно только орфографирование (например, простановка заглавной буквы в начале предложения)
- абстрактная синтаксическая структура – основывается на синтаксических ролях составляющих структуры
- лексикализованный падежный фрейм – основывается на семантических ролях составляющих структуры

Выходом блока поверхностной реализации обычно являются строки предложений, однако если в схеме генерации явно выделяется блок форматирования, то выход из блока поверхностной реализации – это полностью реализованная грамматическая структура.

3. Особенности построения генераторов в системе DEMLinG

Система DEMLinG создавалась как средство создания, поддержки и выполнения грамматик многоязыковой генерации.

Традиционная проблема систем генерации связана с производительностью грамматик. Когда широкомасштабные грамматики достигают определенного размера, их поддержка, расширение и повторное использование значительно усложняются. В результате такие системы могут быть достаточно эффективными в некоторых предметных областях, однако они страдают недостаточной производительностью, необходимой, например, для таких приложений как интерактивные системы. Увеличение производительности грамматик напрямую связано с уменьшением области поиска правильного перехода между информационными состояниями [14, 12].

Поставленная проблема может решаться несколькими способами: изменением основополагающих парадигм создания широкомасштабных грамматик (например, внесением модуляризации и конвейерности), использованием статистических оценочных функций как основополагающих для принятия решений [14], а также использованием грамматик

ориентированных на конкретную задачу [14], в которых грамматики характеризуются сравнительно слабым набором грамматических описаний.

При создании системы DEMLinG было принято решение реализовывать последний способ решения поставленной проблемы. Это означает, что DEMLinG поддерживает создание генераторов, работающих в узких предметных областях с ограниченным лексиконом, грамматическими и семантическими вариациями.

3.1. Представления данных для генераторов системы DEMLinG, входное представление

Входное представление, с которым работают генераторы, созданные в системе DEMLinG – это произвольное XML-представление, которое интерпретируется как иерархическое представление знаний в терминах иерархии семантической сети. Сеть строится из именованных узлов, нагруженных свойствами (парами тип-значение), связанных неименованными связями. На определение свойств накладывается ограничение уникальности их типов в пределах одного узла Общей концепции принципов построения входной структуры пока еще не выработано. Однако предполагается, что оно должно быть языково-независимым и описывать структуру знаний плюс цель дискурса, определенную в корневом узле структуры, или структуру формируемого текста (структуру содержания текста).

Для реализованного генератора QGen входное представление данных – это структурное представление SQL-like запроса, генерируемого системой [1]. Пример запроса пользователя приведен на рис. 1.

```
SELECT count(Employee.Marital state)
FROM Employee
WHERE (Employee.Marital state<>'married') AND
      (Employee.Sex='f')
```

Рис. 1: Текстовое представление ЕЯ запроса пользователя «*How many single women work in the company?»*, преобразованное системой InBASE.

Такое входное представление не несет в себе языково-ориентированной информации, так как объекты этого запроса (Например: Employee, Marital state) – объекты МПО системы InBASE. Оно описывает лишь структуру запроса пользователя, помещенную в шаблон SELECT-FROM-WHERE.

В теории построения грамматик известны два наиболее популярных подхода к представлению обрабатываемых данных: представление данных в виде непосредственных составляющих, и в виде деревьев зависимостей [6]. Представление в терминах непосредственных составляющих позволяет описывать данные с точки зрения их структурной декомпозиции, например, через образование составных лингвистических конструкций из набора более простых линейно непересекающихся отрезков, которые называются непосредственными составляющими (immediate constituents) этой конструкции. Второй метод предполагает построение дерева из бинарных отношений непосредственного

подчинения зависимого слова (modifier) главному (head). В системе DEMLinG механизмы обработки данных построены с расчетом на использование наиболее популярного в последнее время в Западном лингвистическом сообществе механизма непосредственных составляющих. На это представление настроены механизмы распространения свойств, поддерживающие такие процессы, как согласование, распространение контекстной информации, реализация грамматической группы, поддержка абстракции главного члена правила.

Процесс генерации описывается в системе DEMLinG как последовательное преобразование входной структуры данных к реализованной грамматической группе, к которой в конце генерации применяется процедура сбора лексической информации для формирования результирующего текста. Все промежуточные представления, возникающие между различными этапами генерации, подчиняются тем же законам, что и входное представление (см. выше). Промежуточные представления различаются только характерными особенностями структуры, свойственными каждому отдельному представлению.

3.2. Ресурсы генераторов системы DEMLinG

Языковые ресурсы генератора делятся на ресурсы планирования, словарь и ресурсы реализации. Подобное разделение соответствует последним тенденциям развития подхода описания многоязыковых генераторов, описанным в разделе 2. Каждый из перечисленных типов ресурсов реализуется на своем интерпретируемом языке. Таким образом, поддерживается требование разделения кодирования лингвистических знаний и обрабатывающего компонента в системах генерации. Поддержка разных языков для реализации различных этапов генерации обусловлена особенностями преобразований выполняемых на каждой фазе.

Вариация выразительных возможностей языков необходима для упрощения конструкций описания лингвистических аспектов на разных этапах и для частичного контроля соблюдения стиля разработки генераторов, поддерживаемого системой DEMLinG. Под стилем разработки генераторов здесь понимается поддержка определенной структурной декомпозиции ресурсов генерации, т.е. поддержка разделения между этапами (см. ниже) и контроль над реализацией внутри каждого этапа действий, характерных этому этапу.

Ресурсы планирования и реализации представляют собой наборы продукционных правил, состоящих из узла применения (соответствует образцу в описании обычной структуры правил, представленном в разделе 2), дополнительных условий применения и действий. Дополнительные условия проверяют наличие определенных узлов в структуре, наличие заданных свойств в узлах структуры, проверяют значения этих свойств. Действия направлены на осуществление структурных преобразований части обрабатываемого представления, ограниченной узлом применения, и модификацию свойств ее узлов.

Различия в языках представления ресурсов планирования и реализации мотивированы качественной разницей в процессах, описываемых этими ресурсами. Язык описания ресурсов планирования предназначен в первую очередь для описания масштабных структурных трансформаций данных. Поэтому в этом языке представлен мощный механизм сопоставления образца с элементами структуры данных и реализованы операции трансформации, работающие с образцами (перемещение, добавление, удаление, упорядочение и другие). Набор операций расширен специальными операциями,

характерными для реализуемой схемы генерации. К таким операциям относятся операции выделения предложений, сопоставления вершины иерархии данных с грамматической группой и назначение свойств грамматических ролей членам грамматической группы, а также операции распространения контекстной информации об объемлющих грамматических группах по дереву.

Язык описания ресурсов реализации работает только с грамматическими группами, осуществляя их реализацию. Механизм сопоставления развит там значительно слабее: правила описывают грамматические группы как одноуровневую структуру, состоящую из членов грамматической группы и родительского узла, отождествляемого с этой группой (групповой узел). Правила могут добавлять новые члены группы и упорядочивать члены группы, модифицировать наборы их свойств. Набор операций этапа реализации также расширен специальными операциями, характерными для реализуемой схемы генерации. Среди таких операций можно выделить операцию согласования и механизм наследования морфологических свойств, назначенных групповому узлу, главным членом группы.

В соответствии со сказанным в разделе 2, словарь системы DEMLinG перечисляет соответствия сущностей структуры содержания, выраженных свойствами узлов иерархии данных, лингвистическим ресурсам, таким как содержательные слова и словосочетания, грамматические связи. Конструктивно, словарь разработан также в виде набора продукционных правил.

Подобно правилам планирования и реализации, правила словаря состоят из трех частей: сопоставляемого свойства (соответствует образцу в описании обычной структуры правил, представленном в разделе 2), дополнительных условий применения и действий. Дополнительные условия выражают ограничения на окружение сопоставляемого свойства. Там могут перечисляться требования к определенности других свойств в обрабатываемом узле, а также ограничения на их значения. Действия сопоставляют свойству левой части правила лексическое свойство (лексему) или грамматическую структуру, в которую отображается обрабатываемый узел (соответствует описанию словосочетания в грамматике), а также дополнительные свойства, ограничивающие реализацию выбранной лексемы или словосочетания. Поскольку выбор определенного лексического эквивалента понятию МПО в общем случае влияет на грамматическую структуру высказывания, словарем поддерживается также еще один специальный вид действия – передача свойства из словаря от обрабатываемой вершины вверх по грамматической структуре высказывания узлу, ассоциированному с грамматической группой, в которой участвует обрабатываемая вершина.

Применение правил ресурсов генерации осуществляется системой DEMLinG последовательным применением выделенных наборов правил к иерархии данных. Применение каждого набора происходит обходом иерархии данных сверху вниз и слева направо. Применение свода правил к узлу структуры означает последовательный перебор правил свода с попыткой применить каждое из них к этому узлу. Последовательность применения выделенных наборов правил определяется сценарием работы генератора, задаваемым внешним ресурсом генератора.

3.3. Особенности представления свойств в представлениях данных генераторов системы DEMLinG

Чтобы система имела возможность по-разному обрабатывать свойства разной природы, свойства одинаковой природы объединяются в системе DEMLinG в группы. Различаются следующие группы свойств:

- Лексические свойства. Эта группа включает в себя все свойства типа lex. Например: lex:and
- Морфологические свойства: включает в себя свойства нескольких морфологических типов, таких как *PartOfSpeech*, *Gender*, *Number*, *Case*, *Person*, *Tense* etc. Область возможных значений морфологических свойств ограничена конкретным морфологическим типом. Конкретный набор морфологических свойств для каждого языка определяется морфологическим модулем (см. в разделе 3.4).
- Ролевые свойства: специальный тип свойств, задающий грамматические роли членам грамматической группы (см. описание ресурсов реализации). Такие свойства назначаются на этапе планирования, а используются для сопоставления с членами реализуемой грамматической группы на этапе реализации.
- Свойства описания грамматического контекста: включают в себя все свойства, тип которых определяется как IsIn<имя грамматической группы>:<числовое значение>. Механизм распространения свойств грамматического контекста рассматривается ниже. Числовое значение отражает удаленность грамматической группы, на которую указывает контекстное свойство, от узла, в котором оно определено.
- Свойства открытой группы: включает в себя все остальные свойства.

3.4. Способ модуляризации процесса генерации в системе DEMLinG

Процесс генерации в системе DEMLinG делится на пять этапов, каждый из которых описывается своим выделенным набором правил. Этапы выполняются последовательно без обратных связей.

1) Первый этап – это построение структуры содержания. На этом этапе планируется будущая структура высказывания. Этот этап зависим от структуры знаний. На нем происходит также некоторая унификация входных данных. Таким образом, возможное изменение синтаксиса входных данных влечет за собой изменения только в этой части грамматики. Первый этап представлен в обобщенной структуре процесса генерации, описанной в разделе 2, как этап макропланирования, и описывается правилами ресурсов планирования.

   В системе QGen на этом этапе происходит отсеивание избыточных данных, которые не будут участвовать в будущем высказывании, а также выравнивание входной структуры под единое концептуальное представление.

2) На втором этапе происходит формирование пред-грамматических групп: узлам дерева назначаются свойства обобщенных грамматических групп, сверху вниз распространяется информация грамматического контекста. На этом этапе также происходит агрегирование (разбение структуры содержания на последовательность структур предложений), и поддержка стилевых ограничений (см. раздел 2 - микропланирование). Второй этап

составляет часть этапа микропланирования в обобщенной структуре процесса генерации, описанной в разделе 2, и тоже описывается ресурсами планирования.

3) Третий этап называется этапом лексикализации. На этом этапе происходит отображение объектов МПО, участвующих во входном представлении, в лексические единицы. Здесь также уточняются выбранные грамматические группы на основе сделанного лексического выбора. Третий этап составляет часть этапа микропланирования в обобщенной структуре процесса генерации, описанной в разделе 2, и описывается ресурсами словаря.
4) На четвертом этапе происходит уточнение грамматической структуры будущего высказывания, членам грамматических групп назначаются их грамматические роли. На этом этапе также происходит инициализация удаленных зависимостей (анафорических ссылок). Четвертый этап составляет часть этапа микропланирования в обобщенной структуре процесса генерации, описанной в разделе 2, и описывается ресурсами планирования.
5) Пятый этап – реализация определенных на предыдущих этапах грамматических групп. Она включает согласование между членами грамматических групп, вставку вспомогательных слов, вычисление морфологических свойств в лексических единицах структуры, однозначно определяющих их морфологическую реализацию, простановку знаков пунктуации. Пятый этап представлен в обобщенной структуре процесса генерации, описанной в разделе 2, как этап поверхностной реализации, и описывается ресурсами реализации.

Предложенное деление в целом соответствует последним тенденциям в области построения многоязыковых генераторов (см. раздел 2) и поддерживает концепцию разделения языково-зависимой и интерлингвистической частей ресурсов генератора. Последние статьи в этой области и личный опыт создания генераторов автором (автор участвовал в создании генератора AGILE [23], основанного на использовании среды создания тактических многоязыковых генераторов KPML [15]) говорит о необходимости выделения процесса лексического сопоставления как отдельного этапа на стадии планирования. Такое выделение, а также общая предложенная структура генерации являются причиной выдвигаемых требований адаптивности к смене языка и модели предметной области. Так при смене модели предметной области возникает необходимость проведения радикальных изменений только в словаре, а при смене языка результирующего текста, радикально меняется словарь, и адаптируются модули генератора, соответствующие этапам уточнения грамматической структуры и реализации.

Для обеспечения многоязыковости генератора назначение множества морфологических свойств и построение словоформы были вынесены в отдельный внешний для системы модуль, подсоединяемый к процессу генерации с помощью COM-интерфейса. Такой подход позволяет использовать уже готовые морфологические компоненты, написанные для других проектов и подсоединять их к генератору через реализацию требуемых для генератора операций на любом универсальном языке программирования, поддерживающим механизм COM. Так, например, для реализации русской морфологии использовался морфологический компонент, реализованный в фирме Диалинг, адаптированный для задачи генерации через написание COM-объекта на языке программирования C++ (Генератор реализован на языке Java).

Формализм задания грамматик, предлагаемый системой DEMLinG, позволяет также описать морфологию ЕЯ простым перечислением всех используемых словоформ в словаре, задавая морфологические свойства, необходимые для реализации, как дополнительные

условия применения правил словаря. Такой подход вполне подходит для описания морфологических феноменов для языков со слабым словоизменением (например, английский). Тогда морфологический блок будет описывать только набор морфологических свойств. Всю остальную работу берет на себя словарь.

3.5. Основополагающие принципы распределения признаков в системе DEMLinG

При использовании генератором механизма свойств как основного механизма обработки входной структуры данных, основу процесса генерации составляет процесс накопления свойств – ограничений на реализацию узлов структуры (подробности см. раздел 2, метод свойств). Поэтому принципы распределения признаков в структуре данных являются основным вопросом при построении системы создания генераторов. В проведенном описании системы DEMLinG уже упоминались использованные принципы распределения признаков. Но перечислим их еще раз отдельно, явным образом. При реализации ресурсов генератора принимались во внимание принципы распределения признаков в структуре данных, сформулированные в работах, описывающих формализм грамматики обобщенных составляющих GPSG (General Phrase Structure Grammar) [22]. Ниже приводится перечисление этих признаков с дополнительными разъяснениями об их использовании в грамматиках GPSG и интерпретация этих признаков для системы DEMLinG:

1) *Head Feature Convention (HFC)* – метод описания признаков у составляющих данной группы – группа head-признаков родительской категории в точности совпадает с head-признаками главной дочерней категории. Для системы DEMLinG принцип был интерпретирован как соглашение о наследовании морфологических свойств главным членом грамматической группы. Такое наследование используется на этапе реализации.
2) *Foot Feature Principle (FFP)* – обеспечивает механизм распространения информации по синтаксическому дереву снизу вверх для некоторых признаков. В результате появляется возможность задания в КС-терминах таких феноменов как удаленные зависимости. Для системы DEMLinG принцип был проинтерпретирован как назначение уникальных идентификаторов некоторым узлам структуры и дальнейшее связывание по назначенному идентификатору с этим узлом другого узла структуры по некоторой совокупности признаков. Используется для инициализации удаленной связи на этапе уточнения грамматической структуры.
3) *Control Agreement Principle (CAP)* – описывает в общей форме согласование, т.е. тот факт, что некоторые признаки дочерних категорий, объединяемых в одну родительскую категорию должны совпадать. Для системы DEMLinG принцип был проинтерпретирован как морфологическое согласование. Может быть использовано на любом этапе, реализуется как свойство данных. Активно используется на этапе реализации.

Для системы DEMLinG перечисленные принципы расширяются еще двумя:

4) Передача грамматическим группам признаков-ограничений на реализацию снизу вверх от вершин, преобразованных словарем, – прямых или косвенных участников этих групп. Используется на этапе лексикализации для уточнения выбранных грамматических групп в зависимости от сделанного лексического выбора.

5) Распространение контекстной информации по членам грамматических групп. Используется на этапе формирования пред-грамматических групп для предоставления информации о грамматическом включении узлов при выборе сопоставления с грамматической группой, и лексических единиц при осуществлении лексического выбора.

4. Направления дальнейших исследований

К настоящему времени разработана рабочая версия системы поддержки генераторов DEMLinG, а также прототип генераторов семейства QGen, реализованный для решения задачи перефразирования запроса пользователя к базе данных, описывающих сотрудников предприятия.

Планируется проведение массовой апробации построенного генератора, среди студентов одного из Московских вузов, а также открыв доступ к нему в сети Интернет. Как результат такой деятельности планируется дальнейшее уточнение модели генерации, модификация генератора и среды его разработки.

Планируется также эксперимент по адаптации генератора к другой предметной области – генерации на ЕЯ ответов пользователю из БД. Такой эксперимент позволит провести некоторое обобщение структуры входной информации, а также лучше изучить вопрос генерации связанного текста, проработку вопроса создания структуры дискурса.

В связи с планируемой задачей обобщения структуры входного представления, планируется добавление к схеме генерации блока препроцессора – интерфейса, адаптирующего имеющиеся данные к структуре знаний, обрабатываемой генератором.

Еще одна серьезная проблема, практически не затронутая в данной работе, это форматирование выходного текста. Способ презентации информации пользователю часто играет определяющую роль в успехе программного продукта. Для решения этого вопроса схему генерации также планируется расширить выделением блока постпроцессора, осуществляющего форматирование текста на основе реализованных структур предложений.

Литература:

1) Boldasov M.V., Sokolova E.G., QGen - Generation Module for the Register Restricted InBASE System // In: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, N 2588, Springer-Verlag, 2003, pp. 465—476
2) Болдасов М.В., Соколова Е.Г. InBASE: технология построения ЕЯ интерфейсов к базам данных // *Труды Международного семинара Диалог'2002 по компьютерной лингвистике, Том 2,* Протвино, Июнь 2002, С. 49-60.
3) Жигалов В.А., Соколова Е.Г. InBASE: технология построения ЕЯ интерфейсов к базам данных // *Труды Международного семинара Диалог'2001 по компьютерной лингвистике, Том 2,* Аксаково, Июнь 2000, С. 123-135.
4) Nirenburg, Sergei. 1987. Machine Translation: Theoretical and Methodological Issues. Cambridge University Press, Cambridge.

5) E. Reiter, Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible? // In Proceedings of the 7th International Workshop on Natural Language Generation, pages 163-170, Kennebunkport, Maine, 1994.

6) Шаров С. Средства компьютерного представления лингвистической информации// ITTC, TTC 2, 20.02.2000, URL: http://www.kcn.ru/tat_en/science/ittc/vol000/002/

7) Hans Uszkoreit, Annie Zaenen. Grammar Formalisms. In: Survey of the State of the Art in Human Language Technology , Cambridge University Press ISBN 0-521-59277-1, pp. 116-118

8) Takako Aikawa, Generation for Multilingual MT, URL: http://www.research.microsoft.com/nlp/publications/generation_for_multilingual_MT.rtf

9) Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.

10) Eduard Hovy. Language Generation In: Survey of the State of the Art in Human Language Technology , Cambridge University Press ISBN 0-521-59277-1, pp. 161 – 169

11) Paris, C., Vander Linden, K., Fischer, M., Hartley, A., Pemberton, L., Power, R. and Scott, D. (1995) `A Support Tool for Writing Multilingual Instructions', *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pp. 1398-1404, Montreal, Canada.

12) Ljungberg A., An Information State Approach to Aggregation in Multilingual Generation // M.Sc. Artificial Intelligence: Natural Language Processing Division of Informatics University of Edinburg 2001

13) Robert Kasper. A flexible interface for linking applications to Penman's sentence generator. In Proceedings of the DARPA Speech and Natural Language Workshop Philadelphia, PA, February 1989.

14) Fernando Pereira. Sentence Modeling and Parsing // In: Survey of the State of the Art in Human Language Technology , Cambridge University Press ISBN 0-521-59277-1, pp. 130-140

15) John Bateman; KPML Development Environment – multilingual linguistic resource development and sentence generation. Release 1.1 January 1997 // Institut fuer integrierte Publikations- und Informationssysteme (IPSI), German Centre for Information Technology (GMD), Dolivostr. 15, Darmstadt Germany. URL: http://www.darmstadt.gmd.de/publish/komet/kpml.html

16) Bateman, J. A, Thimas Kamps et al. (2000) DArtBio system: constructive text, diagram and layout generation for informational presentation URL: http://acl.ldc.upenn.edu/J/J01/J01-3004.pdf

17) Pollard, C. and Sag, I. (1994). Head-driven Phrase Structure Grammar. Center for the Study of Language and Information (CSLI) Lecture Notes. Stanford University Press and University of Chicago Press.

18) Emele, M. and Zajac, R. (1990). Typed unification grammars. In Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh, Pennsylvania. Association for Computational Linguistics.

19) Krieger, H.-U. and Schaefer, U. (1994). TDL---a type description language of HPSG. Technical report, Deutsches Forschungszentrum fËur KËunstliche Intelligenz GmbH, SaarbrËucken, Germany.

20) Carpenter, B. (1992). The Logic of Typed Feature Structures, volume 32 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

21) Alshawi, H., editor (1992). The Core Language Engine. MIT Press, Cambridge, Massachusetts.

22) Gazdar, G., Klein, E., Pullum, G., and Sag, I. Generalized Phrase Structure Grammar. Oxford: Basil Blackwell, 1985.

23) AGILE Project, INCO COPERNICUS Modelling Lexical Resources in KPML for Generating Instructions in Slavic Languages, PL961104,1999. URL: http://www.itri.brighton.ac.uk/projects/agile/

24) K. Bontcheva and G. Angelova. Planning and Generating Hypertext Documentation. In: Proceedings of the Workshop "Gaps and Bridges in Natural Language Genaration" (W11), European Conference on Artificial Intelligence ECAI-96, Budapest, Hungary, August 1996, pp. 25 - 28.

25) Мельчук И.А. Опыт теории лингвистических моделей Смысл-Текст. М.: Наука, 1974.