

Еще раз об автоматизированном формировании лингвистических баз знаний: технологические аспекты интеграционного подхода

М.Г. Мальковский, А.В. Субботин

МГУ им. М.В.Ломоносова

malk@cs.msu.su lesha@isd.srcc.msu.su

Данная работа продолжает тему автоматизированного формирования лингвистических баз знаний (ЛБЗ) для систем автоматической обработки текста и речи (ЕЯ-систем), затронутую в [1].

Под формированием ЛБЗ понимается процесс сбора и формализации лингвистической информации, представления ее в виде пригодном для использования в составе ЕЯ-систем и поддержания этой информации в актуальном состоянии.

В [1] отмечалось, что процесс формирования ЛБЗ должен быть автоматизированным. Он не может быть реализован как вручную (из-за больших объемов ЛБЗ), так и в автоматическом режиме (в силу необходимости учета знаний носителей языка и экспертов-лингвистов).

Сбор лингвистической информации осуществляется на основе источников, которые могут быть классифицированы следующим образом:

- Люди[1]:
- эксперты-лингвисты;
- рядовые носители языка (в том числе и эксперты в предметной области).
- Тексты:
- описания языка, созданные специалистами;
- тексты предметной области.
- Существующие ЛБЗ.

Использование всех этих источников необходимо, поскольку ни один из них не дает полной информации о языке:

- Хотя результаты лингвистических исследований обладают высокой степенью достоверности, они не покрывают весь спектр языковых явлений, которые необходимо описать в ЛБЗ, поскольку осмысление и описание новых языковых объектов и явлений неизбежно отстает от развития языка. Важным фактором, который также надо учитывать при использовании этих источников, является их субъективность, обусловленная, в частности, влиянием на позицию эксперта-лингвиста принадлежности его к той или иной лингвистической школе.
- В отличие от экспертов-лингвистов, рядовые носители языка непредвзяты в своей оценке языковых явлений. Но при этом, даже у экспертов в предметной области хорошее знание

терминологии может сочетаться с недостаточной лингвистической компетентностью. Рядовой носитель языка зачастую не может правильно (с лингвистической точки зрения) описать, атрибутировать и классифицировать языковые феномены. В связи с этим следует отметить необходимость участия инженера знаний в процессе получения информации от рядовых носителей языка. Инженер знаний здесь выполняет ту же роль, что и при формировании баз знаний для экспертных систем.

- Строгость лингвистических описаний языка зачастую недостаточна для непосредственной формализации. Даже словарь Зализняка [2], который считается одним из наиболее формальных описаний русской морфологии, все же ориентирован на читателя-человека. К описаниям языка также относятся все замечания, сделанные выше относительно информации, полученной от экспертов-лингвистов, поскольку эти описания представляют собой одну из форм воплощения этой информации.
- Тексты предметной области отражают сегодняшний срез языковой реальности. Однако для получения адекватной информации необходимо использовать большие (представительные) выборки текстов.
- Существующие ЛБЗ создавались в различных концептуальных и технических контекстах. Несовместимость обычно обнаруживается практически на любом уровне: от базовой лингвистической модели до кодировки символов.

Для каждого из указанных видов источников нужны свои специфические способы извлечения лингвистической информации. В зависимости от особенностей используемых источников и формируемой ЛБЗ применяются различные методы обработки исходных данных, эвристики извлечения информации и их комбинации, а процесс формирования ЛБЗ является процессом достаточно сложным, трудоемким, насыщенным экспериментами. На сегодняшний день интеграция различных методов и технологических приемов становится необходимым условием успешного решения задачи автоматизированного формирования ЛБЗ. Это подтверждается и опытом, приобретенным авторами в ходе выполнения проектов по созданию конкретных ЛБЗ.

Однако при интеграции методов формирования ЛБЗ возникают следующие трудности:

- Отсутствуют целостные подходы к интеграции различных методов и поддерживающие их лингвистические и математические модели.
- Реализация методов формирования ЛБЗ ведется на базе разных информационных технологий, часто без опоры на стандарты, что существенно затрудняет их интеграцию.

Предлагаемый авторами подход предусматривает создание моделей, на основе которых могут быть интегрированы различные методы автоматизированного формирования ЛБЗ, и описание принципов их компьютерной реализации. Он предполагает:

1. Построение метамодели ЛБЗ (МЕТАМОДЕЛИ), то есть унифицированного средства представления структуры ЛБЗ и процесса ее формирования.
2. Создание системы автоматизированного формирования ЛБЗ (САФЛБЗ), реализующей МЕТАМОДЕЛЬ.

При наличии такой МЕТАМОДЕЛИ и САФЛБЗ интеграция осуществляется следующим образом:

1. Структура формируемой ЛБЗ отображается на МЕТАМОДЕЛЬ.
2. Алгоритмы формирования ЛБЗ отображаются на МЕТАМОДЕЛЬ.
3. Реализации алгоритмов формирования ЛБЗ интегрируются в САФЛБЗ.

Далее будут рассмотрены основные особенности ЛБЗ, и сформулированы требования к МЕТАМОДЕЛИ и САФЛБЗ. Затем будут кратко описана МЕТАМОДЕЛЬ и принципы построения САФЛБЗ.

Особенности лингвистической информации, представленной в ЛБЗ

Неточный характер

Часть компонентов ЛБЗ содержит информацию, которая по своему характеру является неточной, что обусловлено вариативностью и подвижностью границ языковой нормы и статистическим характером отдельных видов информации. Неточность информации, содержащейся в компонентах, относящихся к семантическому и предметно-зависимому уровням ЛБЗ, обусловлена и сложностью процесса формализации описываемых явлений. Рекомендации по проведению такой формализации обычно формулируются в виде описаний на ЕЯ, апеллирующих к языковой интуиции человека, и могут трактоваться по-разному различными специалистами.

Неполнота

Практически непреодолимой причиной неполноты лингвистической информации является открытость и постоянное развитие ЕЯ: появление новых языковых единиц, изменение свойств существующих единиц и правил их сочетаемости. Такая динамика особенно заметна в подъязыках новых предметных областей с неустоявшейся терминологией.

Другой причиной неполноты лингвистической информации является наличие огромного числа нюансов и языковых особенностей отдельных носителей языка, описать и формализовать которые на сегодняшний день не представляется возможным.

Наличие ошибочной информации

Часть информации, содержащейся в ЛБЗ может быть ошибочной. Ошибочная информация отличается от неточной тем, что для неточной информации известно, насколько она может не соответствовать действительности. Ошибочная информация может быть маркирована даже как точная, но в то же время полностью противоречить реальной ситуации.

Ниже указаны основные причины, ведущие к образованию ошибочной информации в ЛБЗ.

- Устаревание информации.
- Ошибки ручного ввода.
- Несогласованности при формировании ЛБЗ различными экспертами.
- Ошибки автоматизированного формирования.

Влияние указанных источников потенциальных ошибок на качество ЛБЗ может быть ослаблено, однако полностью нейтрализовать его невозможно.

Особенности ЛБЗ как информационного ресурса

Рассматривая ЛБЗ как информационный ресурс, можно выделить следующие ее особенности:

- Большой объем содержащейся информации.
- Сложная структура информация.
- Интенсивность доступа к данным.
- Поддержка адаптивности.

Основные требования к МЕТАМОДЕЛИ

На основе рассмотренных особенностей ЛБЗ сформулируем требования к МЕТАМОДЕЛИ.

- Возможность представления неточной информации.
- Поддержка сложных структур данных и ограничений целостности.
- Гибкость - МЕТАМОДЕЛЬ должна позволять изменять схему ЛБЗ и процесса ее формирования с минимально возможными затратами.
- Реализуемость - МЕТАМОДЕЛЬ должна быть реализуема на базе таких технологий, которые позволят удовлетворить требования к ЛБЗ как к информационному ресурсу.

Требования к САФЛБЗ

Из особенностей ЛБЗ следует, что САФЛБЗ является большой информационной системой. Соответственно, к ней применимы требования, предъявляемые к большим информационным системам (ИС), сформулированные, например, в [3].

Перечислим основные из этих требований:

- ИС должна быть способна к эволюционному развитию.
- Необходимо обеспечить расширяемость системы, т.е. возможность добавления новых компонентов в уже существующую ИС.
- ИС должна проектироваться так, чтобы изменение требований к ней не привело к переработке всей системы или большей ее части.
- ИС должна быть открытой системой [4].
- ИС, по возможности, не должна полностью зависеть от тех или иных производителей аппаратных и программных средств.
- Необходимо предусмотреть возможность интеграции со смежными системами и совместимость со старыми компонентами ИС.
- В ИС необходимо обеспечить возможность распределенной обработки информации в гетерогенных вычислительных средах.
- ИС должна быть надежной и отказоустойчивой, т.е. иметь предсказуемую и адекватную реакцию на нештатные ситуации, обеспечивать сохранность информации при сбоях технических средств или ошибках пользователей и восстановление работоспособности в случае отказа оборудования.

Основы МЕТАМОДЕЛИ

В ходе анализа сформулированных выше требований были выбраны следующие основы построения МЕТАМОДЕЛИ:

- объектный подход и язык UML [5];
- нечеткая математика;
- сети Петри.

Объектный подход предоставляет средства для моделирования и работы со сложными структурами данных. Современный объектный подход, воплощенный в языке UML, вобрал в себя средства моделирования отношений, аналогичные с точки зрения выразительности аппарату семантических сетей. В UML также определены модели, предназначенные для описания поведения объектов (State Diagrams), динамики их взаимодействия (Sequence Diagrams, Collaboration Diagrams) и процессов, протекающих в объектных системах (Activity Diagrams). Причем, последний вид моделей позволяет отображать как потоки управления, так и информационные потоки.

Правила интерпретации для Sequence и Collaboration Diagrams являются достаточно простыми (обмен сообщений между объектами) и в стандарте UML описаны строго. Диаграммы состояний имеют реактивную семантику, унаследованную от аппарата конечных автоматов, которая также формально описана. В действующем стандарте UML не определена в точности семантика выполнения моделей действий (Activity Diagram), особенно в части моделирования информационных потоков. Существующего описания вполне достаточно для практических целей, когда модели можно дополнить рядом соглашений по их интерпретации, однако с точки зрения построения формальной модели формирования ЛБЗ этого недостаточно.

Одним из путей решения этой проблемы является интеграция аппарата моделей действий UML с формализмом сетей Петри в рамках средств расширения UML. Такой симбиоз позволяет, сохранив стандартные выразительные средства UML, дополнить их формальной основой, позволяющей осуществлять строгую интерпретацию. К тому же, в аппарате сетей Петри практически отсутствуют возможности по представлению сложно

структурированных данных. Развитая типовая система UML позволит устранить этот недостаток сетей Петри.

Большинство алгоритмов обработки текстов предусматривает анализ разнообразных объектов (слов, грамматических классов, предложений, рубрик, смыслов и т.д.) и связей между ними. Причем, эти объекты достаточно статичны, их набор невелик и слабо отличается в различных методах формирования ЛБЗ, как и ядро структуры этих объектов. Основные отличия в методах формирования состоят в анализируемых связях между объектами, способах извлечения и представления информации. Так, например, связь между рубриками и ключевыми словами может представляться следующим образом:

- матрица рубрик и ключевых слов;
- набор ключевых слов входит в структуру рубрики;
- связь рубрики и ключевых слов представлена совокупностью таблиц.

Причем сила связи на разных этапах формирования рубрикатора, может быть представлена разными способами:

- абсолютной частотой встречаемости в текстах, отнесенных к одной рубрике;
- относительной частотой встречаемости, в текстах отнесенных к одной рубрике;
- в виде нейронной сети, обученной определять, насколько сильно данное ключевое слово связано с рубрикой.

Как было показано в [6], эти методы представления могут быть транслированы на язык нечеткой математики. При этом связи могут быть представлены нечеткими отношениями, предикатами и правилами, а последовательность преобразований этих отношений - как процесс нечеткого вывода.

Статистические наблюдения, позволяющие оценить частотность или вероятность того или иного события, могут быть также представлены в виде нечеткого предиката или отношения над множеством исследуемых объектов. Нейронная сеть, позволяющая оценивать числовые соотношения, также может быть инкапсулирована в виде системы нечетких правил.

Подчеркнем, что здесь речь не идет об отображении внутренних особенностей построения умозаключений с использованием всех указанных формализмов на нечеткую математику. Предполагается только “оборачивать” методы обработки текстов в нечеткую форму, на базе которой можно уже строить выводы с использованием информации, добытой этими методами.

В UML нет средств описания нечетких фактов и правил, однако этот язык расширяем. Поэтому выразительные средства нечеткой математики могут быть интегрированы в UML с помощью механизмов его расширения.

Нечеткий вывод является наиболее мощным средством преобразования и обработки нечеткой информации. Однако существующие алгоритмы его реализации (метод нечетких резолюций) активно используют механизм возвратов. Поскольку рассматриваемая система является распределенной, применение подобного механизма нечеткого вывода в качестве основного средства обработки информации может оказаться крайне неэффективным.

Нечеткий вывод может быть использован не “глобально”, как основной механизм обработки, а “локально”, в рамках отдельных процессов, описанных, например, с помощью диаграмм действий UML (расширенных сетями Петри и средствами обработки нечеткой информации). При этом в рамках описанного процесса, там где это уместно и может быть эффективно использовано, выполняется нечеткий вывод для получения новой информации.

Итак:

- МЕТАМОДЕЛЬ представляет собой расширение метамодели языка UML средствами нечеткой математики и сетей Петри;
- в САФЛБЗ должна быть предусмотрена возможность интеграции различных методов обработки информации: нейронных сетей, статистических методов и т.п., которые будут предоставлять интерфейс в виде, предусмотренном созданной МЕТАМОДЕЛЬЮ.

Для того чтобы сохранить совместимость с UML, МЕТАМОДЕЛЬ построена как профиль UML.

Механизм профилирования предусмотрен в стандарте UML 1.3. Профиль UML, позволяет использовать UML без изменения или расширения метамодели UML. Профиль UML – это предопределенный набор стереотипов, именованных значений, ограничений и элементов графической нотации, которые в совокупности позволяют использовать UML для отображения специфичных особенностей моделируемой предметной области.

МЕТАМОДЕЛЬ строится следующим образом. Выделяется подмножество метамодели UML, необходимое для моделирования ЛБЗ. Это подмножество адаптируется для интеграции с формализмом нечеткой математики с помощью средств расширения языка: введения новых стереотипов (stereotype) и связанных с ними ограничений (constraints) и свойств (tagged values). Также создается каркас моделей ЛБЗ (framework), включающий в себя классы и другие элементы моделей, на базе которых будут строиться модели ЛБЗ. На основе расширенной таким образом метамодели UML с использованием каркаса создается профиль, на основе которого и осуществляется моделирование конкретных ЛБЗ.

Базовые технологии САФЛБЗ

На основе требований к системе автоматизированного формирования ЛБЗ постараемся обосновать выбор технологий для ее создания.

САФЛБЗ должна обеспечить хранение и доступ к большим объемам информации. При этом, поскольку речь идет как о создаваемой ЛБЗ, так и об источниках лингвистической информации (например, текстах, существующих словарях), для этой информации характерно разнообразие представлений и методов организации доступа к ней.

Так, например, тезаурус может быть представлен в виде бинарного файла в собственном закрытом формате и библиотек без исходных текстов, откомпилированных для конкретной программно-аппаратной платформы. Такое представление является довольно типичным для тезаурусов, поставляемых с коммерческими системами.

Другим примером представления информации может являться использование механизмов какой-либо СУБД. Более того, методы обработки этой информации могут быть реализованы в виде хранимых процедур этой СУБД на нестандартном диалекте SQL.

Пытаться собрать в единой программе столь разнообразные библиотеки и организовать взаимодействие с различными источниками данных представляется достаточно тяжелым процессом. Более правильным путем решения этой задачи является создание распределенной системы.

Другим обоснованием такого подхода к организации системы может служить естественная распределенность источников информации, расположенных на различных вычислительных узлах в локальной или глобальной сети. Требование распределенности системы также выражается в необходимости предоставления удаленного доступа инженерам знаний (лингвистам-экспертам) к системе для корректировки и анализа информации, поскольку физическое перемещение данных часто является затруднительным.

Поскольку МЕТАМОДЕЛЬ основана на объектном подходе логично будет использовать для построения системы объектные технологии. Другой причиной использования объектных технологий является то, что на сегодняшний день объектный подход вобрал в себя все средства более ранних (структурно-функционального и модульного) подходов и поддержан достаточно развитыми и проверенными на практике технологиями.

Исходя из распределенности и «объектности» системы наиболее подходящими технологиями будут являться так называемые технологии распределенных объектов [7].

Технологии распределенных объектов, продвигаемые Object Management Group (OMG) основаны на серии CORBA-стандартов (CORBA, CORBAServices, CORBAfacilities) [8]. Отличием технологии CORBA является изначальная ориентация на интеграцию программных систем и компонентов, написанных на различных языках программирования и работающих на различных программно-аппаратных платформах. В отличие от технологии Java для CORBA не зафиксирован единственный язык программирования, а определен независимый от языка программирования язык спецификации интерфейсов компонентов распределенной системы - CORBA IDL (Interface Definition Language). OMG определены отображения с этого языка на различные языки программирования (в частности, C, C++, Java), что позволяет организовывать взаимодействие компонентов распределенной системы, написанных на этих языках посредством ORB (Object Request Broker).

Стандарты CORBAServices специфицируют сервисы, позволяющие организовывать высокоуровневое взаимодействие в распределенной системе и их CORBA IDL интерфейсы. Среди этих сервисов представлены сервисы именования, передачи сообщений, поддержки распределенных транзакций.

Стандарты CORBAfacilities посвящены более высокоуровневым и менее универсальным сервисам, так называемым «общим средствам» (Common Facilities). Примером такого сервиса является сервис Workflow, который будет рассмотрен ниже.

Семейство технологий распределенных объектов, продвигаемых корпорацией Microsoft (DCOM, ActiveX) во многом аналогичны CORBA, но обладают рядом существенных недостатков [9]:

- Эти технологии изначально ориентированы на работу в среде Microsoft Windows и перенос их на другие платформы крайне затруднен.
- В DCOM компонент распределенной системы представляет собой не объект, а фабрику объектов, вследствие чего объектная модель DCOM фактически не поддерживает понятие состояния объекта.
- Программное обеспечение, реализующее данную технологию, фактически зависит от одного производителя.
- Ограничены возможности по использованию языков программирования для реализации программных компонентов.

Таким образом, наиболее подходящей базовой технологией для создания САФЛБЗ является технология CORBA.

Архитектура САФЛБЗ

Поскольку, как было отмечено выше, информация о языке может быть представлена в различном виде, необходима инкапсуляция особенностей источников данных. Это достигается за счет создания CORBA-сервера, обеспечивающего доступ к данным на основе унифицированного интерфейса.

Этот сервер организуется таким образом, чтобы путем встраивания в него различных классов можно было организовать доступ к данным произвольной структуры, хранящимся в произвольном источнике данных.

Процесс формирования ЛБЗ обычно требует длительных экспериментов, поэтому необходимо иметь возможность достаточно «легковесного» изменения логики их выполнения.

В связи с этим процесс формирования ЛБЗ лучше представлять в декларативном виде, а также иметь возможность изменять ход процесса во время его выполнения. Это достигается с помощью так называемой технологии Workflow [10].

Концепция Workflow разрабатывается в рамках WfMC [10]. В семействе стандартов OMG CORBAfacilities, присутствует горизонтальное общее средство Workflow Management Facility [11], для которого определен стандартный набор IDL интерфейсов для управления потоками работ.

Технология Workflow предусматривает язык заданий для различных исполнителей, то есть описание процесса указывает, какие действия и кто должен выполнить в такой момент.

В связи с этим целесообразно структурировать отдельные операции по формированию ЛБЗ в виде совокупностей методов, реализуемых отдельными компонентами, которые распределены по различным вычислительным узлам.

Так, например, при извлечении тезаурусных связей осуществляется обработка большого количества информации, связанная с несколькими проходами по тексту. Соответственно, реализацию методов обработки текста целесообразно было бы разместить около источника данных, чтобы избежать многочисленных распределенных вызовов.

Некоторые компоненты, например, осуществляющие анализ текстовых связей (нечеткий вывод) с целью выявления тезаурусных отношений, требуют больших вычислительных ресурсов и им для работы необходимо небольшое количество информации. Эти компоненты целесообразно располагать в отдельных процессах и, возможно, на отдельных вычислительных узлах.

Целесообразно отделить пользовательский интерфейс от функциональной логики и доступа к данным системы. Подсистема пользовательского интерфейса может быть построена на основе Internet и Java.

В связи с вышесказанным предлагается следующая архитектура САФЛБЗ:

InformationResource – сервер доступа к данным в терминах сети объектов, связанных нечеткими отношениями. Этот сервер позволяет динамически загружать компоненты (в виде динамических библиотек):

- реализующие доступ к различным классам объектов;

- реализующие обработку данных (те, которые целесообразно размещать в непосредственной близости от данных).

Workflow – реализует ядро OMG Workflow Facility и позволяет динамически изменять и загружать новые описания процессов.

FuzzyInference – подсистема нечеткого вывода. Может реализовываться, например, путем «обертывания» (wrapping) [12] в CORBA IDL интерфейс интерпретатора FPROLOG.

AppServer – подсистема, позволяющая размещать компоненты, обрабатывающие информацию и управлять ими. В качестве AppServer может использоваться какой-либо из CORBA совместимых промышленных серверов приложений.

Все компоненты обрабатывающие информацию, обеспечивающие доступ к объектам, а также описания процессов хранятся в ComponentRepository, предоставляющем унифицированный интерфейс для доступа к ним.

В системе поддерживается как минимум два вида пользовательского интерфейса: интерфейс для эксперта-лингвиста и рядового носителя языка (User Interface) и интерфейс для администратора/инженера знаний (Engineer Interface).

На данный момент создан прототип САФЛБЗ.

Литература

Мальковский М.Г., Абрамов В.Г., Субботин А.В. Об автоматизированном формировании лингвистических баз знаний // Труды Международного семинара Диалог'98 по компьютерной лингвистике и ее приложениям. Т.2 - Казань, 1998.

Зализняк А.А. Грамматический словарь русского языка - М.: Русский язык, 1977.

Сухомлин А.В. Методологический базис открытых систем // Открытые системы, 1996, №4 3.
Ахтырченко К.В., Леонтьев В.В. Распределенные объектные технологии в информационных системах // СУБД, №5-6, 1997 (http://www.osp.ru/dbms/1997/05_06/52.htm).

(<http://www.osp.ru/os/1996/04/48.htm>).

OMG Unified Modeling Language Specification, Version 1.3, June 1999.

Мальковский М.Г., Абрамов В.Г., Пильщиков В.Н., Субботин А.В. Программно-информационное обеспечение обработки текста в интегрированных информационных системах. Научно-технический отчет по Проекту № 037.01.178.23 Подпрограммы «Информатизация России», ВНИЦентр Инв.№ 02.9.90 000109, 1998.

Robert Orfail, Dan Hakrey, Jeri Edwards. The Essential Distributed Objects Survival Guide - John Wiley&Sons Inc., 1996.

998. Pope. The CORBA Reference Guide: Understanding the Common Object Request Broker Architecture - Addison-Wesley, 1998.

999. J. Edwards. 3-Tier Client/Server At Work - John Wiley&Sons Inc., 1997.

The Workflow Reference Model // Workflow Management Coalition document WFMC-TC-1003, 19-Jan-95, ver. 1.1 ([http:// www.aiim.org/wfmc/standards/docs/ tc003v11.pdf](http://www.aiim.org/wfmc/standards/docs/tc003v11.pdf)).

OMG BODTF RFP#2 Submission Workflow Management Facility, Draft Revised Submission, May, 1998 // OMG document bom/98-05-04.

Thomas J. Mowbray, Ron Zahavi. The Essential CORBA: Systems Integration Using Distributed Objects - John Wiley & Sons, Inc., 1995

[1] Получение информации от экспертов-лингвистов и рядовых носителей языка может вестись как в непосредственном диалоге с ЭВМ (on-line), так и в пакетном режиме (off-line), когда некоторой программной системой обрабатываются протоколы интервью.