

ИНСТРУМЕНТАЛЬНАЯ СРЕДА XG#: ОНТОЛОГИЧЕСКИЙ ПОДХОД К РАЗРАБОТКЕ ИНТЕГРИРОВАННЫХ ОБОЛОЧЕК ЭКСПЕРТНЫХ СИСТЕМ

Михаил Никулин

Аннотация: В статье описывается подход к реализации инструментальной среды XG# для создания оболочек экспортных систем с интегрированным представлением знаний. Основной особенностью среды XG# является использование онтологического подхода как для представления прикладных знаний уровня пользователя-эксперта, так и базовых примитивов, выступающих в качестве каркаса ядра инструментальной среды. В частности, онтологии используются для представления входного языка системы, что позволяет не только адаптировать входной язык под пользователя-эксперта, но и интегрировать различные средства представления знаний посредством внесения дополнений и изменений в исходную грамматику. Предлагаемый подход позволяет вносить изменения в грамматику входного языка без переписывания уже существующего кода системы, а лишь путем добавления и регистрации новых библиотек, реализующих требуемую функциональность.

Keywords: экспертные системы, метазнания, онтологический подход, проектирование и разработка информационных систем.

ACM Classification Keywords: I.2 Artificial Intelligence: I.2.4 Knowledge Representation Formalisms and Methods – Representation languages; I.2.5 Programming Languages and Software – Expert system tools and techniques

Введение

Современные оболочки экспертных систем (ЭС), особенно универсальные оболочки ЭС, являются весьма дорогостоящим продуктом, а среди оболочек, наиболее подходящих для целей обучения и, в тоже время, являющихся практически значимыми при разработке реальных экспертных систем, отсутствуют средства, позволяющие адекватно учитывать специфику проблемной области и стиль рассуждений конкретного эксперта, его предпочтений с точки зрения требований к языку представления знаний (ЯПЗ). Поэтому представляется весьма актуальной задача [Churpina, 2005] разработки современной объектно-ориентированной инструментальной среды создания оболочек ЭС с учетом требований, предъявляемых пользователями к средствам представления знаний.

Данная статья посвящена описанию концепции построения и реализации системы XG#, которая направлена на решение указанной выше проблемы. В основе разрабатываемой инструментальной среды лежит специальный *интерфейс разработчика*, позволяющий в наглядной форме с использованием визуальных средств так называемого *инспектора грамматик* [Nikulín, 2004] вносить изменения в описание синтаксиса языка представления знаний, его семантики, а также определять прагматический аспект языковой конструкции относительно его интерпретации в терминах базовых примитивов системы. Онтология, описывающая входной язык системы, является изменяемой частью онтологической базы метазнаний, называемой *метасхемой XG#*, описывающей поведение системы. Компоненты среды XG# реализуются в терминах базовых примитивов ядра и способны автоматически настраиваться на содержимое метасхемы. Предлагаемый подход делает возможным эволюционное развитие системы и

позволяет наращивать ее возможности путем подключения новых компонент, предназначенных, например, для интеграции различных средств представления знаний (продукций, фреймов и др.).

Архитектура XG#

Обобщенная архитектура инструментальной среды XG# показана на рис. 1. Центральной компонентой является метасхема, которая содержит:

- онтологию базовых примитивов (знания системы о самой себе);
- онтологию входного языка системы (синтаксис, семантика, прагматика);
- тиражируемые проблемно-ориентированные онтологии понятий.

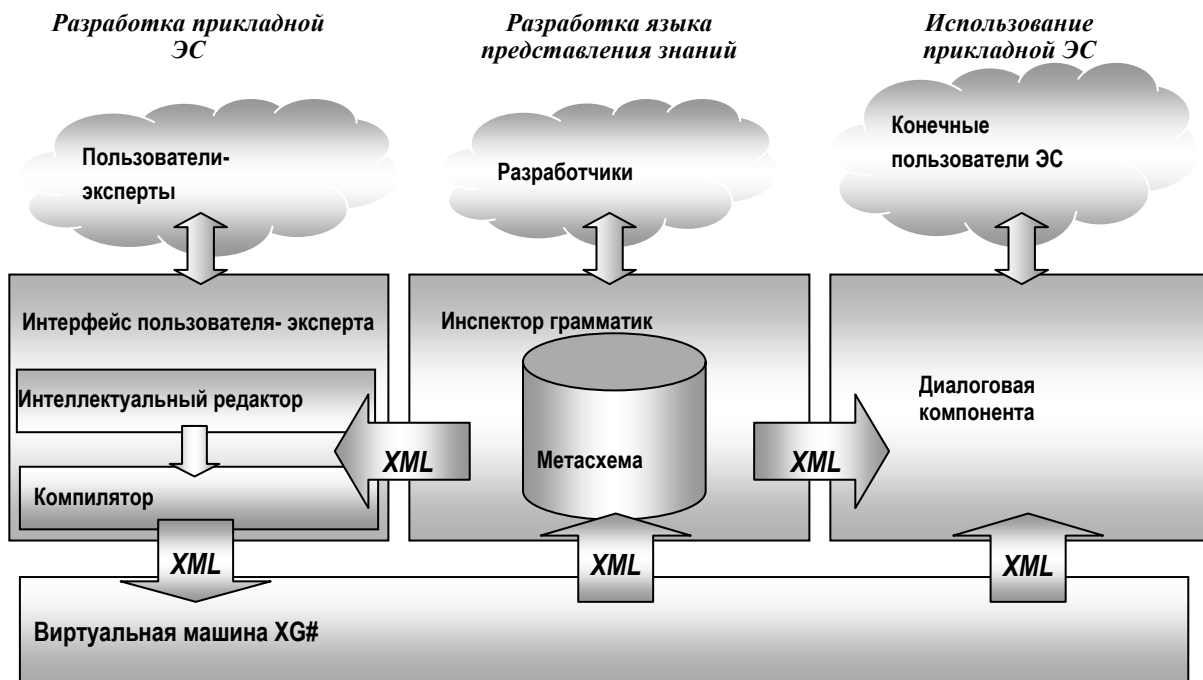


Рис.1. Архитектура XG#

Часть метазнаний системы, а именно, *системные метазнания*, автоматически формируется самим ядром виртуальной машины XG#. Для создания и пополнения той части базы метазнаний, которая описывает язык представления знаний разрабатываемой оболочки ЭС, создан специальный визуальный интерфейс разработчика, так называемый "Инспектор грамматик". *Инспектор грамматик* позволяет в наглядной форме описывать и динамически изменять лексику, синтаксис, семантику и правила генерации объектного кода языка представления знаний.

Таким образом, инструментальная среда XG# – это своего рода *универсальная оболочка экспертных систем, проблемной областью которой является создание специализированных оболочек ЭС*, учитывающих специфику проблемной области, с определенным, возможно, интегрированным средством представления знаний и механизмом функционирования. Универсальность достигается не за счет того, что в систему изначально зашито в интегрированном виде все основные средства представления знаний с поддержкой соответствующих механизмов логического вывода, а за счет возможности динамического внесения изменений в грамматику языка представления знаний.

Компилятор системы XG# получает необходимую информацию о структуре языка из метасхемы. Основной принцип работы компилятора – *раздельное проведение основных фаз трансляции исходного текста программы*. Фаза *синтаксического анализа* формирует, согласно метасхеме, дерево

синтаксического разбора программы либо выдает сообщения об обнаруженных ошибках. На фазе *семантического разбора* происходит анализ семантической корректности дерева синтаксического разбора, согласно правилам, определенным в метасхеме. *Генератор кода* на входе получает семантически правильное дерево синтаксического разбора программы в виде XML и преобразует его с помощью XSL к XML объектного кода виртуальной машины.

Интеллектуальный редактор XG# предназначен для редактирования правил объектного уровня и метаправил уровня предметной области. Интеллектуальным редактор назван потому, что он способен настраиваться на изменения во входном языке системы и поддерживает современный уровень IntelliSense (подсветка синтаксиса, выдача подсказок пользователю). Это стало возможным благодаря тому, что информацию о грамматике конкретного входного языка он берет непосредственно из метасхемы. Кроме того, для генерации контекстных подсказок в процессе редактирования исходного текста программы интеллектуальный редактор использует компилятор для неполного грамматического разбора фрагментов текста. Это дает возможность, например, вывести адекватный текущему контексту редактирования список допустимых конструкций или набор допустимых значений из домена соответствующей переменной в ссылке продукции.

Основное назначение виртуальной машины – выполнение объектного кода. Ядро виртуальной машины способно выполнять базовые арифметические операции, операции по приведению типов и т.п. Все дополнительные компоненты (механизмы логического вывода, библиотеки функций) реализованы в виде отдельных модулей, которые подключаются к ядру системы через специальный шлюз. Для библиотек с компонентами, в свою очередь, также определен интерфейс, посредством которого ядро "опознаёт" и подключает соответствующие модули.

Базовые примитивы XG#

В настоящее время понятие «онтология» является одним из наиболее часто используемых понятий. Термин «онтология» применяется в различных контекстах, в которых ему приписывается различный смысл. Учитывая специфику решаемых в данной работе задач, будем считать, что онтология – это точная спецификация некоторой области, которая включает в себя словарь терминов (понятий) предметной области и множество связей между ними (типа «элемент-класс», «часть-целое»), которые описывают, как эти термины соотносятся между собой в конкретной предметной области. Фактически в данном случае *онтология – это иерархическая понятийная основа рассматриваемой предметной области*, для которой разработана информационная система.

Онтология базовых примитивов ядра XG# основывается на простейших понятиях «сущность» и «связь». Все объекты системы представляются *множеством взаимосвязанных сущностей*. Объект «связь» может выступать как в роли атрибута, связывая при этом сущность и значение, так и в роли отношения, связывая две или более сущности. Каждая сущность описывается онтологией, состоящей из других сущностей. Таким образом, сущности выступают в роли «классов» или в роли «экземпляров». Роль «класса» для сущности типа класс играют особые сущности «классы», называемые «мета-классы». Мета-классы являются самоописывающимися сущностями, выступающими в качестве экземпляров и классов для самих себя одновременно. Такое замыкание онтологии символизирует количество осмысленных метауровней и исключает необходимость бесконечного описания каждой мета-сущности на новом мета-уровне. Отношения сущностей и их атрибуты описываются специальными сущностями-экземплярами называемыми «связь». Сущность «связь» описывает арность и классы связываемых сущностей посредством других связей с соответствующими классами сущностей. Поскольку каждая связь описывается соответствующей ей сущностью «связь», в конечном итоге найдется сущность «связь»,

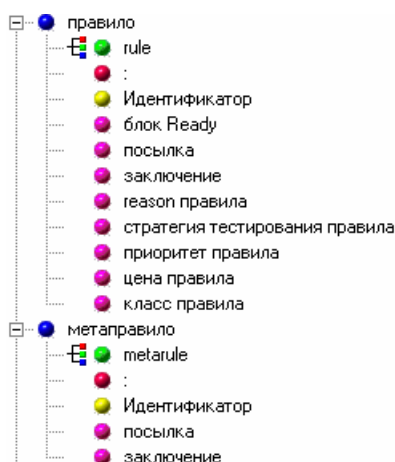


Рис. 3. Грамматика продукционного мета-правила

Таким образом, модификация онтологии входного языка XG# позволила без переписывания уже существующего кода компонент системы расширить функциональность механизма логического вывода. Учитывая ориентацию системы на цели обучения, наличие инспектора грамматик (визуального инструмента для явного представления и редактирования онтологии входного языка), служит целям более глубокого осмысления студентами способов использования метазнаний.

Заключение

В данной статье был рассмотрен подход к реализации инструментальной среды создания оболочек экспертных систем XG#, в которую была заложена возможность динамического изменения грамматики входного языка системы, с целью обеспечения адаптируемости системы под потребности пользователя-эксперта, проблемную область, для целей обучения, а также с целью её дальнейшей расширяемости. Таким образом, описанная в данной работе архитектура настраиваемого компилятора позволила разработать среду-оболочку для создания оболочек, входным языком которых могут быть не только языки представления знаний в общем случае, но и широкий спектр других строго типизированных языков, грамматика которых удовлетворяет требованиям, описанным в работе [Nikulin, 2004].

Библиографический список

- [Nikulin, 2004] М.Б. Никулин, А.В. Жигалов. Концепция построения и реализации оболочек экспертных систем на базе метазнаний. В сб. тр. V Всерос. научно-практ. конф. молодых ученых, аспирантов, студентов "Молодежь. Образование. Экономика". Ч.4. Ярославль: Изд-во "Ремдер", 2004. С. 81-86.
- [Churpina, 2005] С.И. Чуприна, М.Б. Никулин. Инструментальная среда XG#: разработка проблемно-ориентированных оболочек экспертных систем с интегрированным представлением знаний // Успехи современного естествознания. №5, 2005. Материалы XXXII Междунар. конф. IT+S&E'2005 "Информационные технологии в науке, образовании, телекоммуникации и бизнесе". Украина, Гурзуф, 2005. С. 105-107.

Сведения об авторе

Михаил Никулин – Пермский государственный университет, аспирант кафедры математического обеспечения вычислительных систем; Россия, г. Пермь, 614990, ул. Букирева, д. 15; e-mail: mihmix@mail.ru