



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ТЕХНОЛОГИИ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ, ОСНОВАННЫЕ НА ИНТЕГРИРОВАННОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Загорулько Ю.А.

*Институт систем информатики им. А.П. Ершова Сибирского отделения
Российской академии наук, г. Новосибирск, Россия*

zagor@iis.nsk.su

В докладе рассматриваются технологии разработки интеллектуальных систем на основе интегрированной модели представления знаний, которая объединяет взаимодополняющие друг друга методы и средства представления и обработки знаний следующих моделей: онтологической, сетевой, продукционной, недоопределенной и императивной. Благодаря тому, что интеграция указанных методов и средств выполнена на основе объектно-ориентированного подхода, интегрированная модель позволяет представлять одновременно декларативные и процедурные знания, а также знания о вычислениях. Кроме того, такой способ интеграции позволяет добиться эффективной реализации интеллектуальных систем.

Ключевые слова: технология разработки интеллектуальных систем; интегрированная модель представления знаний; онтология; специализированная оболочка интеллектуальной системы.

ВВЕДЕНИЕ

Как показывает практика, при создании интеллектуальной системы (ИС) самым ответственным и трудоемким процессом является построение ее базы знаний (БЗ) [Гаврилова и др., 2001], которая, в конечном счете, определяет полезность и качество всей системы. В связи с этим возникает насущная потребность в моделях представления знаний и построенных на их основе средствах создания баз знаний, обладающих не только высокой выразительностью и гибкостью, но и являющихся достаточно простыми и понятными в использовании не только инженерам знаний – специалистам по представлению знаний, но и экспертам – специалистам в конкретных предметных областях (ПО).

В рамках работ по искусственному интеллекту накоплен широкий спектр методов и средств представления и обработки знаний. Так, семантические сети [Лозовский, 1984; Лозовский, 1990] используются как универсальная память для хранения любой информации, которую можно представить в терминах объектов и отношений между ними. Продукционные системы [Поспелов, 1990] рассматриваются как удобное средство для выражения знаний экспертов в виде множества правил вида «ЕСЛИ-ТО» и организации логического вывода. Фреймы [Минский, 1978],

представляющие объектно-ориентированный подход в искусственном интеллекте, служат как для повышения уровня представления знаний, так и для обеспечения возможности совместного использования декларативных и процедурных знаний. В рамках парадигмы программирования в ограничениях [Ушаков и др., 2000; Apt, 2000] знания можно задавать в виде системы ограничений на значения параметров моделируемых объектов. Аппарат недоопределенных типов данных [Нариньяни, 1986] и разработанный на его основе метод недоопределенных вычислительных моделей [Нариньяни и др., 1998] дает возможность оперировать неточно заданными значениями и частично описанными объектами. Большим потенциалом обладают языки логического программирования [Братко, 2004], позволяющие описывать сложные процессы обработки информации в декларативном стиле. В последнее время при разработке ИС успешно применяются дескриптивные логики [Baader et al., 2003] и построенные на их основе языки и системы представления знаний [OWL, 2004]. Широкую популярность у разработчиков баз знаний получили онтологии [Guarino, 1998; Гаврилова и др., 2006], с помощью которых можно достаточно просто представить согласованную систему понятий моделируемой области знаний.

Как правило, существующие инструментальные системы и оболочки ИС снабжены каким-то одним

из упомянутых выше средств представления знаний, в то время как при создании прикладных ИС часто возникает необходимость их комплексного использования, поскольку ни одно из них, взятое в отдельности, не может обеспечить в полном объеме потребностей разработки реальной прикладной системы. В связи с этим инструментальные средства разработки ИС должны включать различные взаимодополняющие друг друга методы и средства представления и обработки знаний. При этом возможны разные способы интеграции таких средств – начиная с интеграции на уровне подсистем и компонентов ИС и кончая интеграцией на уровне единого формализма (модели).

Хотя первый способ интеграции проще в реализации, более предпочтительным является второй способ интеграции, при котором все необходимые методы и средства объединяются в одном формализме (модели и/или языке представления знаний). Такая интеграция не только создает предпосылки для построения мощных баз знаний (за счет совместного использования различных взаимодополняющих друг друга методов и средств), но и обеспечивает концептуальное единство создаваемых баз знаний (все компоненты БЗ становятся связанными общими понятиями). На основе такого формализма могут быть созданы инструментальные средства, с помощью которых эксперты смогут сами строить базу знаний ИС в полном объеме, не прибегая к помощи посредников – инженеров знаний и программистов. Благодаря этому значительно упрощается разработка базы знаний ИС и повышается ее качество. Кроме того, интеграция всех требуемых средств в рамках одной модели (языка) создает предпосылки для создания на ее основе эффективных прикладных ИС, так как устраняются неизбежные расходы ресурсов на организацию взаимодействия программных систем, реализующих отдельные методы и средства представления знаний.

Таким образом, использование в качестве базиса ИС модели, объединяющей различные взаимодополняющие методы и средства представления знаний, позволяет удовлетворить большинство требований, выдвигаемых к технологиям создания ИС в [Голенков и др., 2011], в частности, обеспечивает семантическое представление знаний, полностью абстрагированное от особенностей технической реализации интеллектуальной системы, минимизирует участие программистов в разработке ИС, позволяет экспертам в ПО принимать непосредственное участие в разработке ИС, тем самым расширяя круг разработчиков ИС и делая технологии их создания массовыми.

В данной работе будут описаны интегрированная модель представления знаний, построенные на ее основе инструментальные средства, а также основанные на этом инструментарию технологии разработки ИС, ориентированные на экспертов.

1. Интегрированная модель представления знаний

1.1. Требования к модели представления знаний

Для того чтобы модель представления знаний была пригодна для разработки широкого класса интеллектуальных систем, она должна обеспечивать представление всех видов знаний, необходимых для разработки и функционирования ИС.

Прежде всего, необходимо представлять знания о предметной области, в рамках которой будет функционировать система. Сюда входят знания об основных сущностях ПО (понятиях и объектах), а также знания о том, каким образом эти сущности связаны между собой. К последним относятся знания об отношениях, непосредственно связывающих понятия, а также более сложные виды знаний, представляющие различного рода зависимости между понятиями ПО, как логические, так и функциональные. В частности, при разработке большого класса ИС возникает необходимость в представлении знаний о ПО в виде правил «ЕСЛИ-ТО». Также довольно часто возникает потребность в представлении знаний в виде ограничений на значения параметров, описывающих некоторый объект или систему объектов.

Для лучшего структурирования ПО и обеспечения более лаконичного ее описания важно иметь возможность выстраивания понятий предметной области в иерархию «общее-частное» и поддержки наследования свойств по этой иерархии.

Базы знаний большинства ИС также содержат конкретные знания из предметной области (предметные знания), которые представляются в виде экземпляров понятий (конкретных объектов) и связей между ними – экземпляров отношений или ограничений, заданных на значениях атрибутов экземпляров понятий.

Другим важным видом знаний, которые нужно представлять в ИС, являются знания о решаемых в рамках моделируемой ПО задачах и способах их решения (методах и алгоритмах). Эти знания характеризуют проблемную область (ПрО) интеллектуальной системы. Такого рода знания могут быть как декларативными, так и процедурными. К первому типу относятся знания, описывающие пространство решаемых ИС задач, включая разбиение задач на подзадачи и связь подзадач с методами их решения, ко второму – знания, представляющие как методы решения задач, так и конкретные алгоритмы.

Кроме знаний, которые постоянно присутствуют в БЗ, необходимо представлять знания, описывающие фрагмент действительности (ситуацию), который задает контекст и входные данные для задач, решаемых ИС. Такие знания, как и предметные знания, обычно представляются в виде экземпляров понятий и отношений и/или

ограничений, заданных на значениях атрибутов экземпляров понятий.

При описании предметных знаний, контекста и входных данных задачи довольно часто возникает необходимость представлять знания об объектах, имеющих неточные значения атрибутов, путем задания оценок таких значений, например, в виде множеств возможных значений и/или ограничений на значения. Для представления таких знаний необходимо обеспечить возможность использования в экземплярах понятий атрибутов, имеющих недоопределенные значения, и задания экземпляров ограничений, связывающих такие атрибуты.

Рассмотренные выше виды знаний, необходимые для полноценного функционирования ИС, не могут быть представлены средствами одной модели. В связи с этим возникает необходимость в модели представления знаний, которая обладает набором методов и средств, достаточным для представления в ИС всех перечисленных выше видов знаний.

1.2. Состав интегрированной модели представления знаний

Исходя из рассмотренных выше требований, предлагается интегрированная модель представления знаний (ИМПЗ-модель), которая объединяет в рамках одного формализма различные взаимодополняющие друг друга средства и методы представления и обработки знаний следующих моделей – онтологической, сетевой, продукционной, императивной и недоопределенной. (Последняя включает аппарат недоопределенных типов данных и разработанный на его основе метод недоопределенных вычислительных моделей, который, как показано в [Нариньяни и др., 1998], фактически является универсальным методом программирования в ограничениях.)

Интегрированная модель представления знаний описывается шестеркой:

$$\langle ONT, SN, FN, PS, IM, GM \rangle \quad (1)$$

где *ONT* – прикладная онтология, задающая описания основных сущностей (понятий и отношений) предметной и проблемной областей приложения в виде классов объектов и отношений,

SN – объектно-ориентированная семантическая сеть (ОО-сеть), свойства объектов и отношений которой определяются онтологией *ONT*,

FN – функциональная сеть (Ф-сеть), включающая экземпляры ограничений, заданные на значениях атрибутов объектов ОО-сети *SN*,

PS – система правил-продукций, описанных в терминах классов и отношений онтологии *ONT* и операций над их экземплярами,

IM – средства императивного программирования,

GM – общий механизм вывода/обработки

знаний и данных, представленных в ИМПЗ-модели, объединяющий механизмы вывода/обработки каждой из интегрируемых моделей.

В качестве каркаса (базиса) интегрированной модели представления знаний выступает прикладная онтология, которая, являясь основным средством описания ПО и ПрО, определяет главные сущности (классы объектов и отношений между ними), которые затем используются остальными средствами модели.

Формально прикладная онтология *ONT* может быть представлена следующим образом:

$$\langle C, R_{in}, R_C, T, D, A_C, A_R, Cntr, P_R, R_{rev} \rangle \quad (2)$$

где $C = \{C_1, \dots, C_m\}$ – конечное непустое множество классов, описывающих понятия некоторых ПО и ПрО;

R_{in} , где $R_{in} \subseteq C \times C$ – строгий частичный порядок на множестве классов C , задающий отношение наследования;

$R_C = \{R_1, \dots, R_n\}$, $R_i \subseteq C \times C$ – конечное множество бинарных отношений, заданных на классах C ;

$T = \{T_1, \dots, T_t\}$ – множество типов данных;

$D = \{d_1, \dots, d_p\}$ – множество доменов, таких что $d_i = \{s_1, \dots, s_k\}$, где s_i – значение некоторого простого типа из T ;

$A_C = \{a_1, \dots, a_r\}$, $a_j \subseteq C \times (T \cup D)$ – конечное множество атрибутов, описывающих свойства классов C ;

$A_R = \{a_1, \dots, a_v\}$, $a_k \subseteq R \times (T \cup D)$ – конечное множество атрибутов, описывающих дополнительные свойства отношений R_C ;

Cntr – множество ограничений, заданных на атрибутах классов, т.е. логических выражений вида $Exp_i(e_{i1}, \dots, e_{iw})$, где $e_{ik} \in A_i$ или $e_{ik} \in T_j$, т.е. e_{ik} либо имя атрибута, либо константа;

P – множество математических свойств (симметричность, рефлексивность, транзитивность и т.д.), которые могут быть заданы для отношений R_C ;

R_{rev} , где $R_{rev} \subseteq R \times R$ – множество пар взаимнообратных отношений из R_C .

Прикладная онтология *ONT*, задавая комплексное описание понятий и отношений моделируемых ПО и ПрО в виде классов объектов и отношений, инкапсулирующих в себе семантические свойства и ограничения на свои атрибуты и аргументы, тем самым определяет структуру семантической и функциональной сети, с

помощью которых представляются предметные знания и знания о задачах, а также задает множество сущностей, в терминах которых описываются продукционные правила.

С помощью отношения R_m на классах онтологии S может быть задана иерархия «общее-частное», по которой организуется наследование свойств (включая атрибуты, отношения и ограничения) вышестоящих классов нижестоящими.

Благодаря включению в ИМПЗ-модель аппарата недоопределенных типов данных, все простые типы данных, входящие в T , имеют недоопределенные расширения, обеспечивающие возможность представления недоопределенных значений (н-значений) в виде интервала или множества допустимых значений и оперирования ими. Это позволяет, в частности, задавать атрибутам объектов ОО-сети н-значения.

Функциональные зависимости между атрибутами сущностей $a_j \in A_C$ могут задаваться в описаниях классов объектов и отношений в виде ограничений $cntr_i \in Cntr$, связывающих значения атрибутов объектов. (В описаниях бинарных отношений ограничения задаются на атрибутах их аргументов.) Такие ограничения представляют собой множество логических выражений (формул), связывающих арифметическими, теоретико-множественными и логическими операциями значения атрибутов одного или нескольких объектов. Следует заметить, что благодаря включению в ИМПЗ-модель метода недоопределенных моделей с помощью ограничений задаются не только условия корректности значений связанных ими атрибутов, но и функциональные интерпретации каждого из этих значений относительно других, что обеспечивает потенциальную возможность вычисления (уточнения) неизвестных (неточных) значений атрибутов через известные (или более точные).

В зависимости от потребностей представления знаний о ПО любому бинарному отношению $R_k \in R$ может быть «приписано» одно или несколько не противоречащих друг другу математических свойств $P_k \in P$ из следующего набора: рефлексивность, симметричность, транзитивность, антирефлексивность, антисимметричность, а также задано обратное отношение.

Конкретные знания о ПО, как и знания о моделируемом в ИС фрагменте действительности или контексте задачи представляются в объектно-ориентированной семантической сети SN , включающей множество экземпляров классов (объектов) и отношений. Для представления этого вида знаний также служат экземпляры ограничений, накладываемых на атрибуты объектов ОО-сети и

хранимые в функциональной сети FN , связанной с ОО-сетью через атрибуты объектов.

Для уточнения значений слотов объектов, ограничения на которые представлены в функциональной сети, используется метод недоопределенных вычислительных моделей.

Описание предметных знаний в виде правил «ЕСЛИ-ТО» средствами системы продукционных правил PS является естественным и удобным для многих областей. В интегрированной модели система продукционных правил работает над семантической сетью, состав понятий и отношений которой определяется онтологией. В связи с этим посылка и заключение любого правила могут описываться в терминах классов и отношений онтологии. Это позволяет не только удобно представлять экспертные знания в виде правил «ЕСЛИ-ТО», но и строить довольно мощные по своим изобразительным и операционным возможностям правила и с их помощью описывать на достаточно высоком уровне довольно сложные процессы вывода и обработки информации.

Знания о пространстве задач могут быть представлены в прикладной онтологии, где (1) описываются типы задач и методов, предназначенных для их решения, и (2) задается разбиение задач на подзадачи и связь задач с методами их решения (знания второго вида также могут быть представлены средствами ОО-сети). При этом сами методы решения могут быть представлены продукционными правилами, объединенными в группы и/или модули. Для этих целей в модель включаются средства структурирования множества правил и управления им, такие как операторы активации и деактивации групп правил, которые могут сочетаться с императивными средствами управления.

Для поддержки эффективной реализации методов и алгоритмов, а также обеспечения создания полноценных прикладных ИС в ИМПЗ-модель включены традиционные для императивной модели средства. Они применяются для разработки алгоритмов в случаях, когда их реализация другими средствами ИМПЗ-модели (например, продукционной модели) была бы слишком неэффективной по времени и/или затратной по памяти, а также для реализации численных вычислений и обеспечения интерфейсных возможностей ИС.

В дополнение к традиционным императивным средствам, таким как процедуры и функции, операторы присваивания, ветвления, цикла и т.п., ИМПЗ-модель включает механизм альтернатив, который является мощным средством поддержки процессов вывода и обработки информации в условиях, когда неточно заданы не только объекты, но и отношения между ними. Данный механизм реализуется оператором альтернативы, который позволяет задавать и перебирать альтернативы (варианты вычислений), обеспечивая при

возникновении противоречия во время обработки очередной из них переход к следующей альтернативе.

Интеграция указанных методов и средств выполняется на основе объектно-ориентированного подхода. При этом:

- понятия и отношения описываемых ПО и ПрО представляются классами объектов и отношений онтологии *ONT*, инкапсулирующими в себе семантические свойства и ограничения на свои атрибуты и аргументы;
- объекты (экземпляры классов) и экземпляры отношений представляются в семантической сети *SN*, структура и свойства которой определены в онтологии;
- ограничения *Cntr* на значения атрибутов объектов представляются в функциональной сети *FN*, связанной через эти атрибуты с семантической сетью *SN*;
- все необходимые процессы вывода и обработки информации (реализующие процесс решения задачи) осуществляются системой продукционных правил *PS*, работающей над семантической сетью, механизмом удовлетворения ограничений, реализуемым в функциональной сети *FN*, а также средствами императивного программирования *IM*.

Взаимосвязи интегрируемых моделей показаны на рисунке 1. Здесь видно, что сетевая и продукционная модели используют сущности, определенные в онтологической модели. В частности, структура и свойства семантической сети определяются в онтологии, а посылки и заключения продукционных правил описываются в терминах классов и отношений онтологии. В свою очередь, средства оперирования, предоставляемые онтологической моделью, используют средства оперирования сетевой модели (для того, чтобы привести ОО-сеть в соответствие с онтологией). Средства оперирования продукционной модели используют и функционал сетевой модели (для сопоставления посылок правил с ОО-сетью и выполнения действий над ней, определенных в заключении правил), и средства оперирования, заданные в императивной модели. Кроме того, сетевая модель использует средства оперирования недоопределенной модели, а последняя – сущности, определенные в сетевой модели.

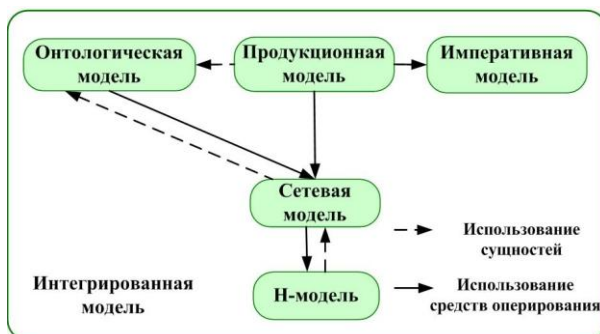


Рисунок 1 – Взаимодействие интегрируемых моделей

1.3. Схема реализации интегрированной модели

ИМПЗ-модель поддерживает представление и решение множества задач, которые в зависимости от назначения прикладной ИС могут быть условно разделены на вспомогательные (подготовительные), выполняемые на этапе настройки системы, и основные – ориентированные на конечного пользователя или получение конечного результата. Однако, такое разделение является условным, так как одна и та же задача может быть для одного приложения вспомогательной, а для другого – основной. В связи с этим в этом разделе рассматривается общий механизм решения задач независимо от их места в «общем разделении труда» в прикладной ИС.

Решение всех задач в ИМПЗ-модели обеспечивается виртуальной машиной (ВМ), реализующей общий механизм вывода/обработки знаний и данных *GM*. Данную виртуальную машину будем называть **главной виртуальной машиной** (ГВМ), потому что она управляет всеми процессами обработки знаний и данных и включает как команды, отвечающие за обработку сущностей онтологической, продукционной, сетевой и недоопределенной моделей, так и команды, реализующие императивные операторы.

ГВМ может быть представлена в виде иерархии виртуальных машин (рисунок 2), каждая из которых обеспечивает обработку определенного вида знаний и данных.

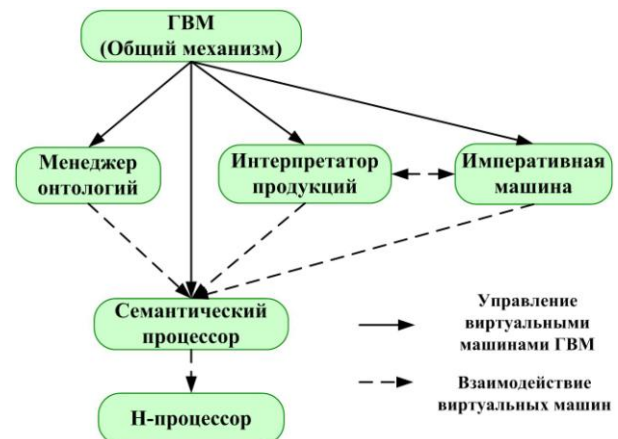


Рисунок 2 – Иерархия виртуальных машин

Выполнение команд создания и редактирования прикладной онтологии обеспечивается виртуальной машиной, которую мы будем называть **менеджером онтологий**. При этом следует иметь в виду, что редактирование онтологии не сводится к простому добавлению, удалению или модификации ее элементов. Семантика операций, выполняемых менеджером онтологий, должна гарантировать логическую целостность новой версии онтологии. Кроме того, если в приложении уже имеется непустая семантическая сеть, то после выполнения операций над онтологией она должна оставаться согласованной с онтологией, т.е. содержать в своем

составе экземпляры только тех классов и отношений и с такими свойствами, какие есть в онтологии.

Команды, обеспечивающие весь комплекс операций над семантической сетью, включая создание, удаление и редактирование объектов и экземпляров отношений, а также поиск фрагментов сети, удовлетворяющих заданному образцу, реализованы в **семантическом процессоре**. При этом нужно учитывать, что семантическая сеть тесно связана с функциональной сетью. В связи с этим при создании пустой ОО-сети создается пустая Ф-сеть, и выполнение операций над объектами и экземплярами отношений ОО-сети при наличии у последних ограничений на атрибуты и/или аргументы влечет за собой выполнение соответствующих операций над Ф-сетью.

Весь спектр операций над функциональной сетью, включая создание, модификацию и интерпретацию ограничений, обеспечивается виртуальным потоковым процессором, который в дальнейшем мы будем называть **Н-процессором**.

Интерпретатор продукций отвечает за работу системы продукционных правил, реализуя процессы вывода/обработки информации, представленные в виде структурированного множества правил.

«Императивная машина» обеспечивает выполнение команд обработки данных и управления, реализуемых в императивной парадигме программирования, а также поддерживает механизм альтернатив.

Взаимодействие виртуальных машин обеспечивается путем включения команд одних машин в реализации команд других. При этом часть команд рассматривается в качестве вызовов процедур или фрагментов, составленных из команд этой же или другой машины. Так, команды «императивной машины», интерпретатора продукций и менеджера онтологий включают вызовы команд семантического процессора, команды семантического процессора обращаются к командам Н-процессора, а команды интерпретатора продукций могут включать команды императивной машины.

Таким образом, процесс решения задачи в ИМПЗ-модели разворачивается в последовательность команд, исполняемую разными виртуальными машинами. Эта последовательность определяется типом приложения и не должна противоречить схеме взаимодействия виртуальных машин, представленной на Рис.2. В частности, в соответствии с этой схемой напрямую к Н-процессору может обращаться только семантический процессор, а к семантическому процессору могут обращаться все остальные ВМ, за исключением Н-процессора.

Построенная таким образом интегрированная модель предоставляет средства для комплексного описания как декларативных, так и вычислительных свойств понятий и отношений некоторой области

знаний, а также средства оперирования их экземплярами в продукционном стиле. Доступ к указанным средствам ИМПЗ-модели осуществляется посредством языка представления и обработки знаний (ЯПОЗ), синтаксические конструкции которого отображаются в команды и данные описанных выше виртуальных машин. На этом языке можно естественным образом описывать и решать задачи, в том числе, требующие сочетания логического вывода и вычислений над неточно заданными значениями.

Язык ЯПОЗ включает два типа средств – средства для описания предметной области (декларативных знаний) и средства для спецификации приложений (процедурных знаний), предназначенных для использования в рамках описанной предметной области. Первая группа средств позволяет описывать ПО в виде классов объектов и отношений онтологии, а также конструировать необходимые для этого новые типы данных и домены. Вторая группа средств предназначена для описания функциональности приложения, главным образом, в виде системы продукционных правил, работающей над семантической сетью, задавать на них структуры и порядок их активации. Кроме того, с целью поддержки разработки полноценных приложений в ЯПОЗ включены традиционные средства императивного программирования, а также средства, обеспечивающие разработку эргономичных пользовательских интерфейсов.

2. Инструментальные средства разработки ИС, основанные на интегрированной модели

Для ускорения и упрощения процесса разработки интеллектуальных систем широко применяется такой класс инструментальных систем, как оболочки ИС [Попов и др., 1996], представляющие собой ИС с пустой или частично заполненной базой знаний. Достоинством оболочек является то, что разработка прикладных ИС на их основе не требует участия программистов, так как наполнение базы знаний входит в компетенцию инженеров знаний и экспертов.

На основе средств, предоставляемых ИМПЗ-моделью, могут быть построены оболочки, предназначенные для разработки различных типов систем, основанных на знаниях, – от оболочек информационных систем, обеспечивающих пользователю содержательный доступ к знаниям и данным определенной области знаний, до оболочек экспертных систем и систем поддержки принятия решений. В зависимости от функциональности и назначения оболочка ИС включает в качестве своего программного ядра ту или иную конфигурацию виртуальных машин, а в качестве средств настройки ИС на ПО – соответствующий программному ядру набор средств редактирования базы знаний.

В рамках данной работы рассматриваются два

типа таких оболочек – оболочка систем, основанных на знаниях (СОЗ), и оболочка порталов знаний.

2.1. Оболочка системы, основанной на знаниях

Оболочка систем, основанных на знаниях, предназначена для построения интеллектуальных систем традиционного типа, т.е. таких СОЗ, за процессы вывода и обработки информации в которых в основном отвечает система продукций. В то же время благодаря тому, что такие СОЗ базируются на ИМПЗ-модели, они способны решать задачи требующие сочетания логического вывода и вычислений над неточно заданными значениями.

Оболочка СОЗ (рисунок.3) включает типовую СОЗ с пустой базой знаний и средства спецификации БЗ.

В типовую СОЗ входят следующие компоненты: интерфейс, база знаний и программное ядро, обеспечивающее реализацию процессов вывода и обработки информации в создаваемых СОЗ.

Интерфейс обеспечивает диалог (взаимодействие) полученной системы с конечным пользователем или внешними подсистемами. СОЗ может включать интерфейсы двух типов – стандартный и специализированный.

Стандартный интерфейс – это абстрактный интерфейс, который обеспечивает основные виды взаимодействия с СОЗ – ввод данных и получение результатов работы. Такое взаимодействие осуществляется на «языке семантической сети», посредством помещения в основную память программного ядра экземпляров объектов и отношений семантической сети или их образцов. При этом дальнейшая интерпретация введенных и полученных в результате работы СОЗ данных (например, на входные и выходные) возлагается на внешние системы, использующие данную СОЗ.

Специализированный интерфейс строится под конкретный тип приложения – для этого в языке ЯПОЗ есть специальные средства и библиотеки. При этом обмен данными с программным ядром также осуществляется через его основную память.

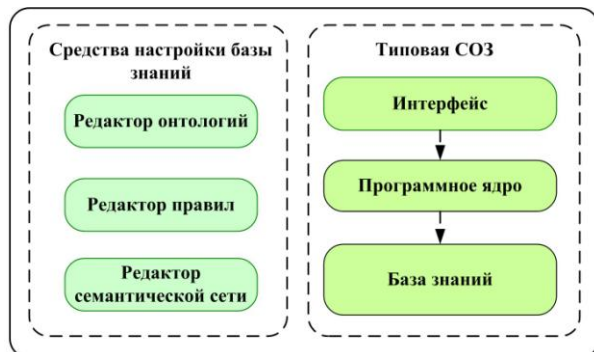


Рисунок 3 – Оболочка СОЗ традиционного типа

Программное ядро СОЗ включает интерпретатор продукционных правил, «императивную машину», семантический процессор и Н-процессор (порядок

взаимодействия между модулями ядра показан на рисунке 5).

Базу знаний прикладной СОЗ (рисунок 4) составляет онтология ПО и ПрО, для работы (моделирования) в которых предназначена СОЗ, семантическая сеть, функциональная сеть, а также система продукций, определяющая основные процессы вывода и обработки информации в СОЗ.

Связанные между собой общими данными семантическая и функциональная сети играют роль основной памяти системы. В семантической сети представлены декларативные знания об объектах и отношениях между ними, в функциональной сети – вычислительные отношения (ограничения), заданные на атрибутах объектов. Вид и свойства этих сетей определяется понятиями и отношениями, введенными инженером знаний на этапе конструирования онтологии.

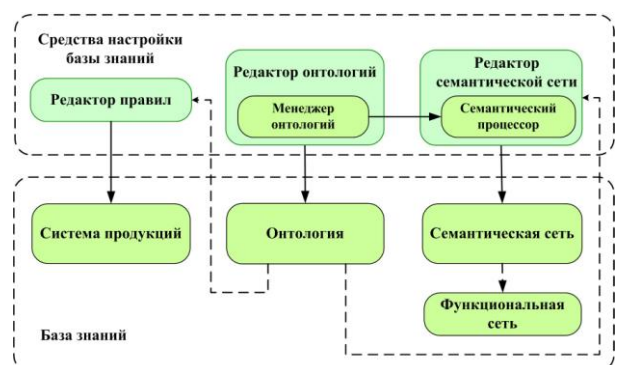


Рисунок 4 – База знаний СОЗ и средства ее настройки

Средства настройки БЗ включают редактор онтологий, предоставляющий удобный интерфейс к менеджеру онтологий, редактор семантической сети, обеспечивающий интерфейс к семантическому процессору, с помощью которого создается начальное наполнение семантической сети, задающее контекст решения задачи или исходное состояние объекта (системы) управления, а также редактор правил, с помощью которого с использованием сущностей, определенных в онтологии, создаются продукционные правила.



Рисунок 5 – Система виртуальных машин оболочки СОЗ традиционного типа

В целом оболочка СОЗ включает все виды рассмотренных выше виртуальных машин (рисунок 5), а также описанные выше редакторы. Кроме того,

в оболочке вводятся определенные ограничения на последовательность применения ВМ (т.е. на ВМ задается частичный порядок).

Возможны следующие последовательности применения виртуальных машин: 1) редактор онтологий, менеджер онтологий; 2) редактор правил; 3) редактор семантических сетей, семантический процессор, Н-процессор; 4) интерпретатор продукций либо семантический процессор, Н-процессор, либо «императивная машина», семантический процессор, Н-процессор. При этом номер на стрелках, выходящих из ГВМ (см. рисунок 5), определяет порядок, в котором эти последовательности могут выполняться.

В соответствии с приведенной схемой типовой сценарий работы с оболочкой следующий:

1. С помощью редактора онтологий создается онтология.
2. С помощью редактора правил с использованием понятий и отношений, определенных в онтологии, описываются и объединяются в группы и модули продукционные правила, которые затем компилируются и загружаются во внутреннюю память системы.
3. С помощью редактора семантической сети создается начальное наполнение семантической сети (фрагмент действительности), задающее контекст решения задачи или исходное состояние объекта (системы) управления.
4. Функционирование системы в рабочем режиме, инициируемое запуском системы продукций.

На последнем этапе в процессе обработки фрагмента действительности формируется результат решения задачи, который представляется в виде экземпляров классов и отношений, также помещаемых во фрагмент действительности.

СОЗ, построенная на основе предложенной оболочки, может быть либо самостоятельной прикладной ИС, либо выступать в качестве интеллектуального ядра другой системы.

2.2. Оболочка портала знаний

Интегрированная модель представления знаний может быть применена не только для создания широкого класса систем, основанных на знаниях, но и для решения другой важной задачи – проблемы эффективного доступа к знаниям и информационным ресурсам, представленным в сети Интернет. Для этих целей строится оболочка портала знаний (ПЗ), которая позволяет создавать специализированные системы – интернет-порталы, обеспечивающие интеграцию и систематизацию знаний и информационных ресурсов по требуемой тематике, а также содержательный доступ к ним.

Информационная модель такого портала базируется на интегрированной модели

представления знаний, что позволяет представлять в нем разнородные знания и данные. Для удобства разработки в информационной модели выделяется два уровня – первый уровень составляет онтология портала знаний, задающая понятия и существующие между ними отношения моделируемой области знаний, включая типологию соответствующих ей информационных ресурсов, а второй – объектно-ориентированная семантическая сеть, содержащая информационные объекты (экземпляры понятий онтологии), связанные отношениями, определенными в онтологии. На основе данной информационной модели организуются удобная навигация по знаниям и информационным ресурсам, интегрированным в портале знаний, и содержательный поиск требующихся информационных объектов.

Оболочка портала знаний (рисунок 6) представляет собой типовой ПЗ и включает пользовательский интерфейс, базу знаний, программное ядро, состоящее из менеджера онтологий, семантического процессора и Н-процессора, а также средства настройки БЗ.

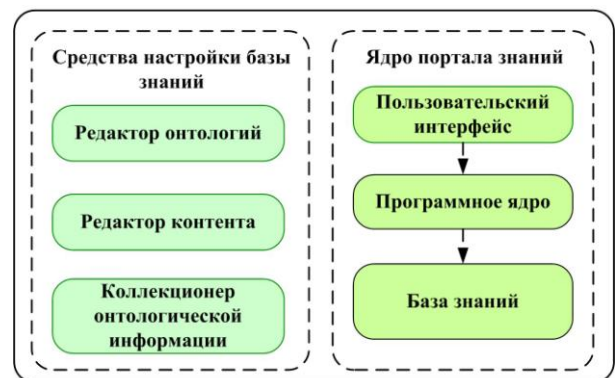


Рисунок 6 – Оболочка портала знаний

Управляемый онтологией пользовательский интерфейс, представляет собой веб-приложение, обеспечивающее навигацию по знаниям и информационным ресурсам, представленным в онтологии и контенте портала знаний, а также содержательный поиск требующейся информации. При выполнении своих команд пользовательский интерфейс обращается к менеджеру онтологий и семантическому процессору, «поставляющим» ему необходимые данные.

Редактор контента предоставляет более удобные и разнообразные средства для управления контентом портала знаний, чем редактор семантической сети, используемый в оболочках СОЗ. При работе он опирается на онтологию и использует команды поиска и редактирования, реализуемые в семантическом процессоре.

Вся работа по созданию конкретного портала знаний состоит в задании необходимых предметных знаний, т.е. онтологии области знаний и соответствующих ей информационных объектов и связей между ними, составляющих контент ПЗ, а также настройке пользовательского веб-интерфейса.

Для этих целей в оболочку ПЗ включены редакторы онтологий и контента, предназначенные для разработки и сопровождения соответственно онтологии и контента ПЗ.

Для повышения степени автоматизации разработки контента ПЗ оболочка предоставляет настраиваемые программные средства для сбора онтологической информации (расширенных метаданных) об информационных ресурсах и публикациях по тематике портала и согласованного пополнения ими контента портала знаний. Эти средства представлены коллекционером онтологической информации (коллекционер ОИ), который при выполнении своих функций использует знания, представленные в онтологии.

Заметим, что указанные редакторы и коллекционер ОИ непосредственно входят в состав портала знаний и, благодаря этому, могут использоваться не только на этапе его создания, но и во время эксплуатации, обеспечивая поддержку его системы знаний в актуальном состоянии.

Оболочка ПЗ включает следующие виртуальные машины (рисунок 7) – редактор онтологий, менеджер онтологий, семантический процессор, Н-процессор, редакторы онтологий и контента, а также коллекционер онтологической информации.

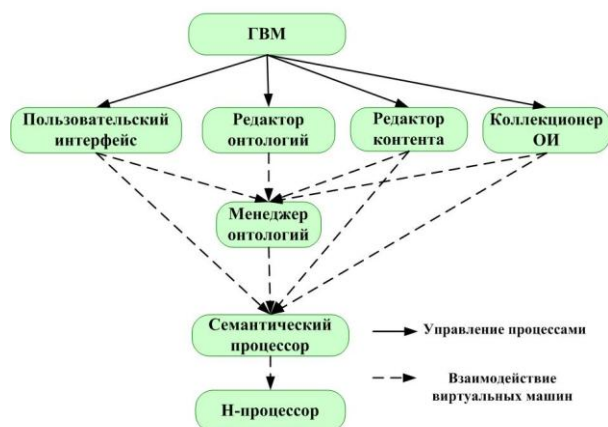


Рисунок 7 – Система виртуальных машин оболочки СОЗ традиционного типа

Возможны следующие последовательности применения ВМ оболочки ПЗ: 1) редактор онтологий, менеджер онтологий, семантический процессор, Н-процессор; 2) редактор контента, семантический процессор, Н-процессор; 3) коллекционер ОИ, семантический процессор, Н-процессор; 4) пользовательский интерфейс, семантический процессор.

Первая последовательность применения виртуальных машин соответствует деятельности по созданию и редактированию онтологии, вторая – деятельности по управлению контентом портала, третья – деятельности по автоматическому пополнению контента портала, четвертая – использованию портала знаний пользователем.

В отличие от оболочки СОЗ в оболочке ПЗ не вводится ограничений на порядок последовательностей применения ВМ, в частности, пользователь может редактировать контент и онтологию в любом порядке.

3. ТЕХНОЛОГИИ РАЗРАБОТКИ ИС, ОРИЕНТИРОВАННЫЕ НА ЭКСПЕРТОВ

Рассмотренные выше программные оболочки являются эффективными средствами построения интеллектуальных систем указанного типа. Однако есть еще одна нерешенная проблема, которая состоит в том, что оболочка ИС, даже основанная на описанной выше интегрированной модели представления знаний, является все еще трудной для непосредственного использования экспертом, так как содержит только универсальные средства представления знаний, которые далеки от области знаний эксперта и поэтому требуют участия в разработке ИС инженеров знаний.

В настоящее время наблюдается тенденция все большего вовлечения экспертов в разработку ИС с целью ее ускорения. Более того, существуют целые классы приложений, использующие большие объемы предметных знаний, при создании которых желательно возложить разработку и сопровождение их базы знаний целиком на эксперта. К таким приложениям можно отнести, например, медицинские экспертные системы, порталы научных знаний, электронные тезаурусы и т.п.

Представляется очевидным, что чем меньше будет разрыв между знаниями и представлениями эксперта и средствами представления знаний инструментальной системы, тем легче ими будет пользоваться эксперту. Поэтому специализация оболочки ИС на определенный класс областей знаний и задач, т.е. включение в ее состав знаниевых компонентов, содержащих базовый набор понятий, относящихся к заданному классу областей знаний, а также программных компонентов, поддерживающих требуемую в ИС функциональность, сделало бы ее более понятной экспертам и пригодной для использования ими без привлечения инженеров знаний [Артемьева, 2008].

На описанной выше идее специализированных оболочек строится концепция технологии разработки СОЗ и порталов знаний, ориентированная на экспертов. В соответствии с этой концепцией общая архитектура инструментальной поддержки разработки ИС выглядит, как показано на рисунке.8.

Базис инструментальной поддержки составляет базирующаяся на ИМПЗ-модели и языке ЯПОЗ интегрированная система представления и обработки знаний, которая представляет собой набор языковых и программных средств, реализующих интегрированную модель.

Следующий уровень включает оболочки, предназначенные для построения прикладных

интеллектуальных систем (СОЗ) и порталов знаний. Оболочки этого уровня предоставляют универсальные средства для построения систем данного типа и предназначены для инженеров знаний.



Рисунок 8 – Общая архитектура инструментальной поддержки разработки ИС

Самый верхний уровень образуют специализированные оболочки, ориентированные на определенные области знаний и классы задач, и, благодаря этому, пригодные для использования экспертами. Причем специализированные оболочки строятся инженерами знаний с помощью средств разработки баз знаний, предоставляемых универсальными оболочками. В соответствии с этим специализированная оболочка СОЗ включает базовые онтологии, на основе которых строится онтология области знаний СОЗ и онтология решаемых ею задач, а также набор специализированных продукционных модулей и библиотек процедур и функций (на языке ЯПОЗ), реализующих решение базовых задач.

Создание конкретной СОЗ на основе такой оболочки заключается в построении онтологии области знаний СОЗ путем расширения и развития базовых онтологий, построении системы продукций путем создания дополнительных продукционных модулей, реализующих задачи, специфичные для разрабатываемой СОЗ, а также наполнении декларативной части базы знаний (семантической и функциональной сети) необходимым набором экземпляров объектов и отношений, представляющих универсальные факты или начальное состояние объекта (системы) управления.

Примерами специализированных оболочек СОЗ, созданных на базе универсальной оболочки СОЗ,

являются оболочка СОЗ медицинской диагностики и оболочка СОЗ диагностики и мониторинга технологической инфраструктуры предприятия [Zagorulko et al., 2010]. Первая из этих оболочек включает общую онтологию медицинской диагностики и лечения, которая содержит понятия и отношения, общие для всех диагностических медицинских систем. Вторая оболочка включает верхний уровень онтологии технологической инфраструктуры предприятия, а также библиотеку процедур и функций для анализа временных рядов, которые могут использоваться в правилах, осуществляющих мониторинг и диагностику технологической инфраструктуры предприятия.

Аналогичным образом строятся и используются специализированные оболочки порталов знаний. Каждая такая оболочка представляет собой типовой портал знаний с набором базовых онтологий, описывающих основные сущности некоторого класса областей знаний, и пустым контентом. Такие специализированные оболочки строятся инженерами знаний путем дополнения универсальной оболочки соответствующим набором базовых онтологий.

Разработка конкретного портала знаний с помощью специализированной оболочки заключается, главным образом, в настройке его на требуемую область знаний, которая состоит в построении онтологии области знаний портала на основе имеющихся в оболочке базовых онтологий и заполнении контента портала под управлением построенной онтологии. Эту работу уже могут выполнять сами эксперты без привлечения программистов и инженеров знаний.

Средствами оболочки портала знаний были построены две специализированные оболочки – оболочка для разработки порталов научных знаний (ПНЗ-оболочка) [Загорулько, 2009] и оболочка для создания многоязычных тезаурусов (Т-оболочка) [Загорулько и др., 2012].

Эти оболочки отличаются, главным образом, набором базовых онтологий. Так, в ПНЗ-оболочку включены две базовые онтологии – онтология научного знания, содержащая базовые понятия, задающие структуры для описания области знаний портала, и онтология научной деятельности, включающая базовые понятия, служащие для описания различных аспектов научно-исследовательской деятельности. В то время как Т-оболочка включает онтологию представления тезауруса, основанную на стандартах построения многоязычных информационно-поисковых тезаурусов.

Следует заметить, что специализированные оболочки могут быть отправной точкой для построения еще более специализированных оболочек. Например, на основе ПНЗ-оболочки можно построить оболочку для разработки порталов знаний, предназначенных для поддержки инновационной деятельности, расширив набор

базовых онтологий ПНЗ-оболочки онтологией производственной и инновационной деятельности.

ЗАКЛЮЧЕНИЕ

В качестве базиса для представления знаний в ИС была предложена модель, интегрирующая взаимодополняющие друг друга методы и средства представления и обработки знаний следующих моделей: онтологической, сетевой, продукционной, недоопределенной и императивной.

Интеграция указанных методов и средств была выполнена на основе объектно-ориентированного подхода. В соответствии с этим подходом понятия и отношения предметной и проблемной областей представляются классами объектов и отношений, инкапсулирующими в себе семантические свойства и ограничения на свои атрибуты и аргументы, объекты этих классов и экземпляры отношений представляются в семантической сети, структура и свойства которой определены в онтологии, ограничения на значения атрибутов объектов – в функциональной сети, связанной через эти атрибуты с семантической сетью, а все необходимые процессы обработки информации (реализующие процесс решения задачи) осуществляются работающей над семантической сетью системой продукционных правил, механизмом удовлетворения ограничений, реализуемым в функциональной сети, а также средствами императивного программирования.

Благодаря такому способу интеграции ИМПЗ-модель позволяет представлять одновременно декларативные и процедурные знания, а также знания о вычислениях (т.е. знания, позволяющие уточнять или находить точные значения параметров описываемого объекта или ситуации). Кроме того, такой способ интеграции позволяет добиться эффективной реализации ИС и сделать доступными все интегрируемые средства в рамках одного языка представления и обработки знаний.

Набор языковых и программных средств, реализующих ИМПЗ-модель, образует интегрированную систему представления и обработки знаний, которая послужила базисом для инструментальных средств разработки интеллектуальных систем. В частности на основе этой системы были построены оболочка СОЗ традиционного типа [Загорюлько и др., 1995] и оболочка портала знаний [Загорюлько и др., 2009]. Обе оболочки могут использоваться инженерами знаний для разработки СОЗ и порталов знаний без привлечения программистов.

Несмотря на успешное применение указанных оболочек для разработки прикладных ИС, следует заметить, что они являются довольно трудными для освоения и непосредственного использования экспертами, так как содержат только универсальные средства представления знаний, далекие от областей знаний экспертов. В связи с этим технологию создания ИС, ориентированную на экспертов, было

решено базировать на специализированных оболочках, настроенных на определенные области знаний и классы задач. При этом сами специализированные оболочки могут создаваться инженерами знаний путем частичного заполнения баз знаний соответствующих универсальных оболочек с использованием имеющихся в ней средств описания предметных знаний.

Работа выполнена при финансовой поддержке Президиума РАН (интеграционный проект СО РАН № 15/10 «Математические и методологические аспекты интеллектуальных информационных систем»).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Артемяева, 2008] Артемяева, И.Л. Специализированные оболочки интеллектуальных систем для сложно-структурированных предметных областей / И.Л.Артемяева // Труды 11-й национальной конференции по искусственному интеллекту с международным участием КИИ-2008 (г.Дубна, Россия). – М: ЛЕНАНД, 2008. –Т.1. – С. 95-103.
- [Братко. 2004] Братко, И. Алгоритмы искусственного интеллекта на языке Prolog / И. Братко. –М.: Издательский дом «Вильямс», 2004. – 637 С.
- [Гаврилова и др., 2001] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2001.
- [Гаврилова и др., 2006] Гаврилова, Т.А. Модели и методы формирования онтологий / Т.А. Гаврилова, Д.В. Кудрявцев, В.А. Горовой // Научно-технические ведомости СПбГПУ, № 4, 2006. – С.21-28.
- [Голенков и др., 2011] Голенков, В.В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н.А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2011): мат. Междунар. научн.-техн. конф. – Минск: БГУИР, 2011. – С. 21-58.
- [Загорюлько и др., 1995] Загорюлько, Ю.А. Интегрированная технологическая среда для создания систем обработки знаний / Ю.А. Загорюлько, И.Г. Попов, В.В. Щипунов // Известия РАН. Теория и системы управления. –1995. – № 5. – С. 210–213
- [Загорюлько, 2009] Загорюлько, Ю.А. Технология разработки порталов научных знаний / Ю.А. Загорюлько // Программные продукты и системы. – 2009. – № 4. –С.25-29.
- [Загорюлько и др., 2009] Загорюлько, Ю.А. Информационная модель портала научных знаний / Ю.А. Загорюлько, О.И. Боровикова // Информационные технологии. – 2009. – № 12. – С.2–7.
- [Загорюлько и др., 2012] Загорюлько, Ю.А. Программная оболочка для построения многоязычных тезаурусов предметных областей, ориентированная на экспертов / Ю.А. Загорюлько, О.И. Боровикова // Труды 13-й национальной конференции по искусственному интеллекту с международным участием КИИ-2012. – Белгород: Изд-во БГТУ, 2012. –Т.4. –С. 76-83.
- [Минский. 1978] Минский, М. Структура для представления знания / М. Минский // Психология машинного зрения. – М.: Мир. – 1978. – С. 249-338.
- [Нариньяни, 1986] Нариньяни, А.С. Недоопределенность в системах представления и обработки знаний / А.С. Нариньяни // Изв. АН СССР. Техн. кибернетика. – 1986. – №5. – С. 3-28.
- [Нариньяни и др., 1998] Нариньяни, А.С. Программирование в ограничениях и недоопределённые модели / А.С. Нариньяни, В.В. Телерман, Д.М. Ушаков, И.Е. Швецов // Информационные технологии. 1998. – №7. – С. 13-22.
- [Лозовский, 1984] Лозовский, В.С. Семантические сети / В. С. Лозовский // Представление знаний в человеко-машинных и робототехнических системах. – М.: ВИНТИ, 1984. –Т.А. – С. 84-121.
- [Лозовский, 1990] Лозовский, В.С. Сетевые модели / В. С. Лозовский // Искусственный интеллект. В 3-х кн. Кн.2. Модели и

методы: Справочник / Под. Ред. Д.А. Поспелова. – М.: Радио и связь, 1990. – С. 28-49.

[Попов и др., 1996] Попов, Э.В. Статические и динамические экспертные системы: Учеб. пособие. / Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. – М.: Финансы и статистика, 1996. – 320 С.

[Поспелов, 1990] Поспелов, Д.А. Продукционные модели / Д. А. Поспелов // Искусственный интеллект. В 3-х кн. Кн.2. Модели и методы: Справочник / Под. Ред. Д.А. Поспелова. – М.: Радио и связь, 1990. – С. 49-56.

[Ушаков и др., 2000] Ушаков, Д.М. Системы программирования в ограничениях (обзор) / Д.М. Ушаков, В.В. Телерман // Системная информатика: Сб. науч. тр. Под. Ред. И.В. Поттосина. Вып.7: Проблемы теории и методологии создания параллельных и распределенных систем. – Новосибирск: Наука, 2000. – С. 275-310.

[Apt, 2000] Apt K. R. Principles of Constraint Programming / K.R. Apt. –New York: Cambridge University Press, 2003. 407 P.

[Baader et al., 2003] Baader, F. The Description Logic Handbook: Theory, Implementation, Applications / F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider – Cambridge, 2003. 574 P.

[Guarino, 1998] Guarino, N. Formal Ontology in Information Systems / N. Guarino // Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. – Amsterdam: IOS Press, 1998. – P. 3-15.

[OWL, 2004] OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004: [Electronic resource]. – 2004. – The University of Sheffield, 1995-2011. – Mode of access: <http://www.w3.org/TR/owl-guide/> – Date of access: 12.11.2012.

[Zagorulko et al., 2010] Zagorulko, Yury. An Approach to Development of the Decision Support System for Enterprise with Complex Technological Infrastructure / Yury Zagorulko, Galina Zagorulko. // Bulletin of NCC – Issue 31.– 2010.–P. 195-207.

TECHNOLOGIES OF DEVELOPMENT OF INTELLIGENT SYSTEMS BASED ON THE INTEGRATED KNOWLEDGE REPRESENTATION MODEL

Zagorulko Yu.A.

*A.P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of
Sciences, Novosibirsk, Russia*

zagor@iis.nsk.su

The paper discusses technologies of development of intelligent systems (IS) based on the integrated knowledge representation model. The integrated model combines methods and means of knowledge representation and processing of ontological, network, production, subdefinite and imperative models. The knowledge base development tools built on the basis of this model are possessed of high expressiveness and flexibility and at the same time easy-to-use for both knowledge engineers and experts. These tools are foundation of intelligent system development technology oriented to experts.

INTRODUCTION

The most labor-consuming and crucial stage of intelligent system development is the building of its knowledge base, therefore there exists a vital need in knowledge representation model and knowledge base development tools built on their bases which are possessed of high expressiveness and flexibility and at the same time easy-to-use for both knowledge engineers and experts.

We propose the integrated knowledge representation model as such model and tools based on this model as such tools. Use of the model combining various mutually complementary methods and means of knowledge representation and processing as basis of intelligent system allows to meet the main requirements which are setting to technologies of development of IS.

MAIN PART

We have conducted comprehensive investigation of all kinds of knowledge which must be represented in intelligent systems. On the basis of results of this investigation a new knowledge representation model was developed. This model which was named the integrated knowledge representation model combines in one formalism methods and means of knowledge representation and processing of ontological, network, production, subdefinite and imperative models. Applied ontology is used as basis of the integrated model. This ontology, being a main means for description of subject and problem domains, defines principal entities (classes of objects and relations between them) which are later used other means of the model.

The integrated model built in such way provides facilities for complex description both declarative and procedural properties of concepts and relations of certain knowledge area as well as facilities for operation of their instances.

Several program shells intended for development of various types of system based on knowledge was designed on the basis of the integrated model. These are shell of knowledge based system and shell of knowledge portal. In turn on the basis of these universal shells the specialized shells which can be used by experts without resorting to the help of programmers and knowledge engineers can developed.

In accordance the suggested technology of IS development the specialized shells are built by knowledge engineers which perform a partial filling of knowledge bases of corresponding universal shells using for this aim knowledge base development tools of these shells.

CONCLUSION

Though shell of knowledge based system and shell of knowledge portal were successfully used for development of set of applied intelligent system, they are hard-to-use for experts. Therefore we have decided to base technology of IS development on specialized shells. On the base of the shell of knowledge based system was developed two specialized shells. First of them is a shell of medical diagnostics and cure, second is a shell of system of diagnostics and operational monitoring of the technological infrastructure of an industrial enterprise. Using the knowledge portal shell two specialized shells was built: shell of scientific knowledge portal and shell for development of multilingual electronic thesaurus.