

## ПОСТРОЕНИЕ ОБОЛОЧКИ ЭКСПЕРТНЫХ СИСТЕМ ДЛЯ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОЦЕССОВ ПРЕОБРАЗОВАНИЯ РЕСУРСОВ

Аксенов К. А.<sup>1</sup>

<sup>1</sup>ФГАОУ ВПО «Уральский федеральный университет им. первого Президента России Б. Н. Ельцина», Екатеринбург, Россия (620002, Екатеринбург, ул. Мира, 19), e-mail: [antonovaannas@gmail.com](mailto:antonovaannas@gmail.com)

В статье представлены результаты построения оболочки экспертной системы для предметной области процессов преобразования ресурсов. Для данной предметной области в качестве модели представления знаний выбран интегрированный фреймово-продукционный подход. Разработан подход к созданию фрейм-системы на основе реляционной базы данных. Достоинством предложенного решения является то, что проектирование модели предметной области в виде фрейм-системы, построение концептуальной модели предметной области, ввод знаний и данных, механизм логического вывода, и запросы к базе знаний реализуются на языке Transact-SQL, т.е. не потребовалась разработка языка вывода на фреймовой модели. Концептуальная модель предметной области основана на расширении диаграммы классов языка UML. Визуальный конструктор для машины логического вывода был разработан на основе диаграммы последовательности язык UML. Применение реляционных баз данных, фреймового и объектно-ориентированного подхода, продукционных систем, позволило реализовать новую экспертную систему для предметной области процессов преобразования ресурсов.

Ключевые слова: экспертная система, процесс преобразования ресурсов, фрейм, объектно-ориентированный подход, динамические продукционные системы.

## DEVELOPMENT THE EXPERT SYSTEM SHELL FOR PROBLEM DOMAIN OF RESOURCES CONVERSION PROCESSES

Aksyonov K. A.<sup>1</sup>

<sup>1</sup>Ural Federal University n.a. the first president of Russia B. N. Yeltsin, Yekaterinburg, Russia (620002, Yekaterinburg, street Mira, 19), e-mail: [antonovaannas@gmail.com](mailto:antonovaannas@gmail.com)

The article presents results of development the expert system shell for problem domain of resources conversion processes. For knowledge representation this problem domain are selected integration approach base on frame and production systems. Within its framework the problem of transition between the knowledge representation, conceptual model and their technical implementation on relational database level, was solved. This approach allows Transact-SQL language to be used for problem domain models design, data and knowledge input, logical output mechanism implementation. Problem domain conceptual model is based on UML class diagram extension. The visual designer for knowledge base inference machine is implemented in decision search diagram, based on UML sequence diagram. New expert system for resources conversion processes was design base on application relations data base, frames, production systems and object-oriented approach.

Key words: expert system, resources conversion processes, frame, object-oriented approach, dynamic production system.

### Введение

Задача построения оболочки экспертных систем для предметной области процессов преобразования ресурсов связана с разработкой интеллектуальных систем мультиагентного динамического моделирования. К процессам преобразования ресурсов относится большинство окружающих нас процессов: производственные и бизнес-процессы, организационно-технические системы (ОТС). Для учета человеческого фактора и сценариев принятия решений, основанных на знаниях, в настоящее время используется аппарат мультиагентных систем, спецификой которых является наличие сообществ, взаимодействующих агентов (интеллектуальных агентов, ИА), отождествляющихся с лицами, принимающими решения (ЛПР). Результаты решения данной задачи в виде

информационной технологии будут востребованы при реализации мультиагентных интеллектуальных систем (при программировании интеллектуальных агентов со знаниями) при решении практических задач многостадийного принятия решений.

### **Анализ моделей представления знаний для процессов преобразования ресурсов**

Основными объектами мультиагентной модели процесса преобразования ресурсов (МППР) являются [2-3]: операции ( $Op$ ), ресурсы ( $RES$ ), команды управления ( $U$ ), средства ( $MECH$ ), процессы ( $PR$ ), источники ( $Sender$ ) и приемники ресурсов ( $Receiver$ ), перекрестки ( $Junction$ ), параметры ( $P$ ), цели ( $G$ ), сообщения ( $Message$ ), агенты ( $Agent$ ). Описание причинно-следственных связей между элементами преобразования и ресурсами задается объектом «связь» ( $Relation$ ). Для построения ядра моделирующей системы был использован аппарат продукционных систем. Определена структура продукционной системы МППР в виде (1):

$$PS = \langle Rps, Bps, Ips \rangle, \quad (1)$$

где  $Rps = \{RES(t)\} \cup \{MECH(t)\} \cup \{U(t)\} \cup \{G(t)\}$  – текущее состояние ресурсов, средств, команд управления, целей (рабочая память);  $Bps$  – множество правил преобразования ресурсов и действий агентов (база знаний);  $Ips$  – машина вывода, состоящая из планировщика и машины логического вывода по базе знаний (БЗ) агентов.

Алгоритм имитатора состоит из следующих основных этапов: определения текущего момента времени  $SysTime = \min(T_j)$ ,  $j \in RULE$  (где  $T_j$  – время активизации  $j$ -го правила преобразования;  $RULE$  – множество правил преобразования ресурсов); обработки действий агентов; формирования очереди правил преобразования; выполнения правил преобразования и изменения состояния рабочей памяти. Для диагностирования ситуаций и выработки команд управления имитатор обращается к модулю экспертной системы (ЭС) [2–3].

Основные требования, предъявляемые к оболочке ЭС для МППР, следующие:

- 1) ориентация на иерархические процессы преобразования ресурсов;
- 2) решение задачи технико-экономического проектирования (ТЭП) ОТС;
- 4) наличие сообществ ИА, управляющих процессом;
- 5) учет человеческого фактора и сценариев принятия решений, основанных на знаниях;
- 6) поддержка объектно-ориентированного подхода;
- 7) визуальные средства работы с БЗ и визуальный конструктор машины логического вывода (в качестве основы визуального языка может быть взят UML).

В качестве моделей представления знаний о процессах преобразования ресурсов выбран интегрированный подход на основе фреймов и продукций.

Для построения концептуальной модели предметной области (КМПО) и решения задачи сокращения затрат на разработку программного обеспечения, в работе использован

фреймовый подход А.Н.Швецова [5], основанный на совмещении фреймоподобных структур с конструкциями концептуальных графов J. F. Sowa [7–9]. Преимуществами данного подхода является деление на активные и пассивные фреймы и учет поведения объекта.

### **Анализ применимости современных промышленных реляционных БД для создания на их основе фреймовых экспертных систем**

Целью данного раздела является оценка возможности эффективного использования современных промышленных реляционных БД для создания на их основе фреймовых ЭС.

Основоположителем фреймовых систем является М. Минский. Характеристики и функции фреймовых систем, а также фреймовая модель представления знаний достаточно хорошо освещены в работе Х. Уэно и М. Исидзука [4]. В настоящее время механизм представления фрейм-структур в табличной форме описан достаточно широко – это является следствием того, что большинство современных носителей информации в ЭВМ хранят данные в форме таблиц или в форме, легко преобразуемой в табличную. Что ставит перед разработчиками задачу преобразования фрейм-структур к табличной форме, однако в литературе не встречено описания методов применения реляционных баз данных (РБД) для реализации фреймов.

В упрощенном варианте любая фрейм-модель представляет собой классическое дерево, и таким образом, проблема сводится к выбору метода организации хранения древообразной структуры. Наиболее простой моделью является дерево, имеющее связи типа «абстрактное – конкретное», при наличии в каждом узле несколько описаний экземпляров объектов. Такая структура имеет весьма специфические функции, встречается редко, но при этом довольно проста в реализации. При статичной структуре фрейма, очень легко организуется: для каждого шаблона организуется отдельная таблица для хранения данных (каждая строка соответствует экземпляру объекта), нет необходимости создавать дополнительные таблицы, следить за уникальностью имён таблиц, нет ошибок, связанных с целостностью структуры дерева. Использование ограничений (constraints) системы позволяет обеспечивать уникальность всех записей. Использование хранимых процедур позволяет полностью инкапсулировать данные. Однако спектр функций такого дерева будет весьма ограниченным.

Как правило, возникает необходимость в реализации более сложных моделей. При необходимости динамически изменять структуру модели необходимо ввести таблицу описания фреймов. В работе [4] предлагается вести таблицу описания слотов:

Таблица 1

<Имя фрейма>

Имя слота	Указатель наследования	Указатель атрибутов	Значение слота	Демон
Слот 1				
...				
Слот N				

Х. Уэно и М. Исидзука в [4] предлагают вводить подобную таблицу для каждого объекта модели. Это оправдано, если фреймовая система создаётся на устройстве, ориентированном на подобные задачи. При реализации в SQL Server мы столкнёмся со следующими факторами: 1) значения слотов должны быть predeterminedены и однообразны, а не соответствовать указателю атрибутов; 2) исчезает необходимость в указателе атрибутов, информацию о нем можно получить из системных таблиц; 3) хранение демона в таблице невозможно; 4) создавать для каждого объекта новую таблицу нерационально; 5) большое количество данных в таблицах повторяется.

Для реализации на SQL Server преобразуем структуру следующим образом: Выделить отдельную таблицу для описания слотов всего дерева или даже всей системы. Идентифицироваться слот будет в этом случае по своему названию и имени объекта. Так как фрейм-структура предполагает однообразность свойств всех объектов, относящихся к одному фрейму, то можно использовать одну группу записей для всех подобных объектов.

В этом случае необходимо исключить из таблицы поле «Значение слота» и следует выделить таблицу для значений слота отдельно. Поле «демон» можно опустить, создав для таблицы экземпляров фрейма триггеры, а в поле присоединенная процедура можно вписать имя хранимой процедуры (этот вопрос будет рассмотрен далее) и информацию о её параметрах. Таким образом, получим следующие таблицы (таблица 2 и 3):

Таблица 2. Описание слотов (терминалов)

Имя слота	Имя фрейма	<имя дерева>	Тип данных	Указатель наследования	Указатель атрибутов	<Хранимая процедура>	<параметры>
Слот 1	Фрейм1						
Слот 2	-''-						
...							
Слот M	-''-						
Слот 1	Фрейм2						
Слот 2	-''-						
...	-''-						
Слот N	-''-						

Таблица 3. Фрейм

Имя объекта	Слот 1	Слот 2	...	Слот M
Объект 1				
Объект 2				

<В скобках указаны необязательные поля>

Для реализации создания ссылок и формул (как наиболее простое решение для реализации механизма наследования во фрейм-системе) используется следующий механизм: 1) на каждый слот, который должен содержать ссылку или формулу в таблице заводится два поля: поле с текущим значением (имеет формат данных слота) и поле, содержащее ссылку /

формулу; 2) в таблице описания фрейма в поле атрибутов шаблона устанавливается значение, показывающее, содержит слот ссылку или формулу.

Хранение информации о наследовании в дереве реализовано в таблице 4.

Таблица 4. Списки деревьев

<Дерево>	Предок	Потомок
<дерево N>	Фрейм 1	Фрейм 2
<дерево N>	Фрейм 2	Фрейм3
<дерево N>	Фрейм 1	Фрейм5

Принятие подобной структуры для организации фрейма приведет к проблеме инкапсуляции данных. Очевидно, что при работе со статической фрейм-моделью для обработки данных в таблицах создадут хранимые процедуры, но при необходимости внесения изменений в структуру модели это приведет к ошибкам (к повреждению всей создаваемой модели). В этом случае можно предложить два следующих выхода.

Первый заключается в том, что сначала заносятся данные в таблицу описания слотов. Эта операция достаточно простая и для её выполнения можно даже не писать дополнительную встраиваемую процедуру. Затем запускается процедура генерации новой таблицы, её регистрации, изменения данных в таблицах, описывающих структуру дерева. Входными данными для этой процедуры будут имена нового фрейма и его предка.

Второй подход универсален, он позволяет сильно упростить работу пользователя или программиста, создающего приложение на основе БД. Он заключается в том, что сначала запускается процедура, принимающая лишь количество полей в новом фрейме и создающая новую процедуру – процедуру создания фрейма с заданным количеством слотов. Эта процедура и выполняет все действия по созданию и регистрации нового объекта БД, а затем удаляется.

Второй подход реализуется при помощи языка Transact-SQL. В нем предусмотрена возможность использования многократно вложенных операторов **Exec[ute]**, которые позволяют создавать и более сложные конструкции. Также с помощью этого оператора создаются процедуры заполнения фреймов и обработка вносимых в модель объектов.

Выделим несколько групп хранимых процедур, необходимых при работе с моделью:

1. Преобразования структуры модели (создаются вместе с БД и динамически).
2. Редактирования данных в модели (создаются динамически).
3. Преобразования данных (создаются вместе с БД и динамически).
4. Присоединенные к данным (создаются динамически).
5. Контроля и поддержания целостности модели (создаются вместе с БД).

Теперь обратимся к использованию процедур преобразования данных и процедур, присоединенных к данным. Для их учета можно поступить следующим образом: для каждого дерева создадим таблицу следующей структуры (таблица 5).

Таблица 5. Таблица указателей процедур

Идентификатор PR	Имя процедуры

В таблице описания слотов в поле «Хранимая процедура» будем помещать идентификатор хранимой процедуры, а в поле «параметры» ключ таблицы (таблица 6).

Таблица 6. Таблица для реализации демонов

Идентификатор PNR	Параметр 1	Параметр 2	...	Параметр N

Таким образом, для каждой процедуры должна быть своя таблица параметров. Для учета у объектов нескольких процедур введена таблица ключей (таблица 7).

Таблица 7. Таблица ключей

Идентификатор объекта	Идентификатор PR	Идентификатор PNR

### **Применение языка Transact-SQL для реализации функции логического вывода**

В целом можно выделить следующие варианты применения языка Transact-SQL для реализации функции логического вывода в МППР:

- механизм вывода (алгоритм) интеллектуального агента полностью или частично оформлен в виде хранимой процедуры, и каждый раз вызывается планировщиком;
- части правил ИА содержат или запросы к БЗ на языке Transact-SQL или ссылки на хранимые процедуры, которые реализуют функции поиска и/или расчетов (вычислений);
- решение задач поиска и вычислений на фрейм-системе (так как модели реальных мультиагентных систем преобразования ресурсов имеют большую размерность).

### **Применение объектно-ориентированного подхода и языка UML для визуального построителя экспертных систем и конструктора машины логического вывода**

При построении КМПО в качестве основы описания структуры фрейм-концептов использована диаграмма классов языка UML. Дальнейшее описание концептуальных графов (семантики) и наполнение данными полученной КМПО образует базу знаний [1, 3].

Для реализации визуального построителя МЛВ оболочки ЭС предложено использование диаграмм последовательности языка UML [1, 3]. Диаграмма последовательности графически описывает последовательность вызова методов между классами при решении определенной задачи (сценария). Данный подход позволяет

визуально (в виде блок-схемы) описать ход решения задачи – последовательность вызовов процедур (методов / демонов) от одного фрейма к другому рис. 1.

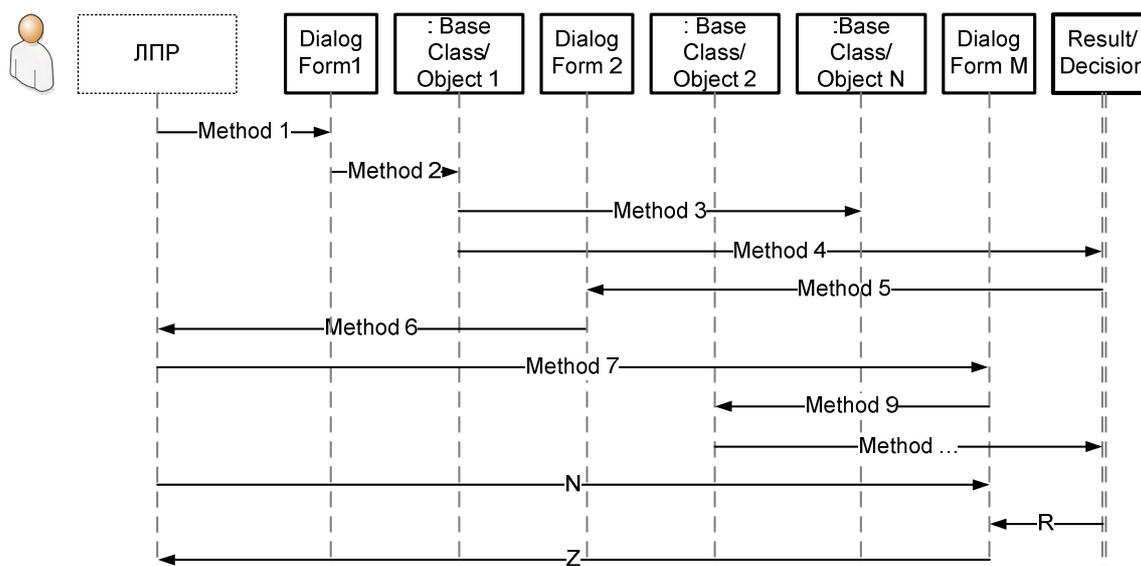


Рис. 1. Диаграмма поиска решения

На рисунке приняты следующие обозначения: *Base Class/Object<sub>i</sub>*,  $i=\{1, \dots, N\}$  – фрейм / экземпляр фрейма предметной области; *Dialog Form<sub>j</sub>*,  $j=\{1, \dots, M\}$  – формы диалога ЭС; *Method<sub>l</sub>*,  $l=\{1, \dots, Z\}$  – методы вывода на сети фреймов и взаимодействия с пользователем; *Result/Decision* – фрейм / экземпляр(ы) фрейма, содержащий результат поиска (решение), полученное путем взаимодействия с ЛПП и выводом на сети фреймов.

### Применение комплекса мультиагентного моделирования BPsim

Комплекс систем поддержки принятия решений BPsim [2–3] со встроенной оболочкой ЭС использовался при решении следующих практических задач: 1) ТЭП мультисервисных сетей связи [1], проектировании информационных систем [6].

### Заключение

Разработан подход к созданию фрейм-системы на основе реляционной БД. Достоинством предложенного решения является то, что проектирование модели предметной области в виде фрейм-системы, построение концептуальной модели предметной области, ввод знаний и данных, механизм логического вывода и запросы к БЗ реализуются на языке Transact-SQL, т.е. не потребовалась разработка языка вывода на фреймовой модели. Данный фактор снижает требования к навыкам системных программистов, аналитиков и инженеров по знаниям, поддерживающих работоспособность данной системы, а также автоматизирует их работу. Использование промышленной БД для хранения данных БЗ позволяет интегрировать данную систему с корпоративной системой предприятия и эффективно применять в ЭС и системах поддержки принятия решений. Реализован визуальный объектно-ориентированный конструктор построения онтологии и построения механизма вывода на знаниях, что позволило реализовать диалоговую ЭС.

### **Список литературы**

1. Аксенов К. А. Интеллектуальная система моделирования «BPsim.MSS» и объектно-структурный метод технико-экономического проектирования мультисервисных сетей связи // Вестник компьютерных и информационных технологий. – № 8. – М., 2010. – С. 19–27.
2. Аксенов К. А. Модель мультиагентного процесса преобразования ресурсов и системный анализ организационно-технических систем // Вестник компьютерных и информационных технологий. – М., 2009. – № 6. – С. 38–45.
3. Аксенов К. А. Теория и практика средств поддержки принятия решений: монография / К. А. Аксенов. – Germany, Saarbrucken: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2011. – 341 p.
4. Представление и использование знаний: пер. с япон. / под ред. Х. Уэно. М. Исидзука. – М.: Мир, 1989. – 220 с.
5. Швецов А. Н. Модели и методы построения корпоративных интеллектуальных систем поддержки принятия решений: Дис. ... д-ра техн. наук: 05.13.01 / А. Н. Швецов. – СПб., 2004. – 461 с.
6. Aksyonov K., Spitsina I., Bykov E., Smoliy E., and Aksyonova O. Computer-supported software development with BPsim products family – integration of multiple approaches // Proceedings of the 2009 IEEE International Conference on Information and Automation (ICIA), 22–25 June 2009, Zhuhai / Macau, China, P. 1532–1536.
7. Sowa J. F. Conceptual graphs for a database interface / J. F. Sowa // IBM Journal of Research and Development. – 1976. – № 20:4. – P. 336–357.
8. Sowa J. F. Conceptual Structures: Information Processing in Mind and Machine / J. F. Sowa. – Reading, M A : Addison. – Wesley, 1984. – 481 p.
9. Sowa J. F. Knowledge Representation : Logical, Philosophical, and Computational Foundations / J. F. Sowa. – Pacific Grove, C A : Brooks/ Cole Publishing Co., 2000. – 594 p.

### **Рецензенты:**

Поршнев Сергей Владимирович, д.т.н., профессор, заведующий кафедрой Радиоэлектроники и информационных систем,, ФГАОУ ВПО «Уральский федеральный университет им. первого Президента России Б. Н. Ельцина», г. Екатеринбург.

Доросинский Леонид Григорьевич, д.т.н., профессор, заведующий кафедрой Информационных технологий, ФГАОУ ВПО «Уральский федеральный университет им. первого Президента России Б. Н. Ельцина», г. Екатеринбург.