

- [Emelyanov et al., 1995] Emelyanov V.V., Ovsyannikov M.V., Yasinovsky S.I. An AI-based Method and Tool for Discrete Manufacturing Systems Simulation and Real-time Control, IEPM'95, International Conference on Industrial Engineering and Production Management, Marrakech April 4-7, (1) 1995, p.322-332.
- [Hill, 1996] Hill David R.C. Object-Oriented Analysis and Simulation, Addison-Wesley, 1996.
- [Kusiak, 1988] Artificial Intelligence. Implications for CIM. Edited by A. Kusiak. JFS (Publications) Ltd, UK, 1988.
- [Laurière, 1987] Laurière Jean-Louis Intelligence Artificielle. Resolution de problèmes par l'Homme et la Machine, Eyrolles, Paris, 1987.
- [Pritsker, 1984] Pritsker A. Alan B. Introduction to Simulation and SLAM II. Systems Publishing Corporation West Lafayette, Indiana, 1984.
- [Емельянов и др., 1998] Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО. -М.: АНВИК, 1998.

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ИНСТРУМЕНТАЛЬНОГО КОМПЛЕКСА АТ-ТЕХНОЛОГИЯ ДЛЯ ПОДДЕРЖКИ ПОСТРОЕНИЯ ИНТЕГРИРОВАННЫХ ЭКСПЕРТНЫХ СИСТЕМ (версия MS Windows)

С. В. Пышагин¹, Г. В. Рыбина¹, В. В. Смирнов¹

В работе обсуждаются отдельные аспекты программной реализации инструментального комплекса АТ-ТЕХНОЛОГИЯ (версия MS Windows), разработанного в лаборатории "Системы искусственного интеллекта" кафедры Кибернетики МИФИ. Комплекс АТ-ТЕХНОЛОГИЯ будет представлен на Выставке программных средств ИИ в рамках КИИ-98.

1. Введение

В работе [Рыбина и др., 1997] был детально описан инструментальный комплекс АТ-ТЕХНОЛОГИЯ (версия MS Windows), предназначенный для компьютерного построения прикладных интегрированных экспертных систем (ИЭС) в статических проблемных областях. Комплекс реализует задачно-ориентированную методологию построения ИЭС [Рыбина и

¹ 115409, Москва, Каширское шоссе 31, МИФИ, Кафедра Кибернетики,
galina@ailab.mephi.ru

др., 1997] и представляет собой взаимосвязанную совокупность средств автоматизации проектирования ИЭС на всех этапах жизненного цикла (ЖЦ) с единым управлением проектом по созданию ИЭС в соответствии с поставленными задачами, набором имеющихся программных средств (ПС), конкретной моделью ЖЦ создания программного обеспечения ИЭС.

Программная реализация комплекса выполнена с использованием объектной модели Delphi 3.0 (фирмы Borland), в соответствии с чем его архитектура является набором взаимосвязанных компонентов, которые могут быть представлены в виде следующих основных групп:

- управляющее ядро, включающее средство управления проектами, планировщик и банк проектов (совокупность баз данных (БД) проектов);
- библиотека базовых классов, состоящих из стандартных и специально разработанных классов и средств их регистрации;
- библиотека ПС, включающая совокупность редакторов диаграмм, диалоговых форм, свойств объектов, гипертекста; средства тестирования проекта и его частей; ПС наполнения словарей, извлечения знаний, выбора решателя, разработки БД, выбора пакетов прикладных программ (ППП), формирования моделей обучаемого и обучения и др.

Реализация всех компонентов комплекса и технология построения прототипов прикладных ИЭС основаны на специально разработанных базовых классах, в связи с чем рассмотрим некоторые особенности реализации и использования базовых классов и библиотеки ПС текущей версии комплекса АТ-ТЕХНОЛОГИЯ.

2. Особенности реализации базовых классов, предназначенных для создания прототипа ИЭС

Проект разработки прикладной ИЭС, создаваемый на основе средств комплекса АТ-ТЕХНОЛОГИЯ, включает следующие основные группы данных:

- БД проекта, содержащую результаты анализа требований пользователя на разработку ИЭС [Калянов, 1997], а также общее описание проекта на языке AT-Visual, формируемое на этапе общего проектирования, и детальное описание проекта на языке AT-Script, формируемое на этапах общего и детального проектирования;
- Данные о зарегистрированных компонентах проекта, применяемых при их повторном использовании.

Перечислим базовые классы текущей версии комплекса АТ-ТЕХНОЛОГИЯ:

- Базовые классы, описанные в VCL-библиотеках Delphi (например, классы TButton, TGrid, TDataSource, TTable, TEdit, и т.п.). Объекты этих классов формируют множество базовых объектов комплекса. Для

регистрация базовых классов применена специальная функция регистрации RegisterClass (Class: TClass);

- Базовые классы, описанные в разработанных компонентах комплекса АТ-ТЕХНОЛОГИЯ, зарегистрированные аналогично стандартным классам Delphi (Таб. 1.).

Таблица 1.

Класс	Описание
TAtObject	Суперкласс всех объектов БД проекта
TAtDataObject	Класс объектов, содержащих указатели на источники данных
TAtControlObject	Класс объектов, содержащих управляющую информацию
TAtRelation	Суперкласс объектов-связей
TAtFlow	Класс объектов, определяющих потоки
TAtDataFlow	Класс объектов, определяющих поток данных
TAtControlFlow	Класс объектов, содержащих описание потоков управления
TAtComponentTool	Суперкласс для всех классов ИС
TAtContainerTool	ИС разработки сложных объектов
TInterfaceForm	Суперкласс интерфейсных форм
TMain	Класс - основная форма проекта
TObjectInspector	Класс - инспектор объектов, служащий для просмотра и редактирования свойств выделенного объекта
TToolsForm	Класс - панель инструментов редактирования
TAboutBox	Класс - информационное окно
TPlanner	Класс, осуществляющий поддержку планирования разработки
TRepository	Класс, инкапсулирующий все механизмы работы с БД проекта

Проиллюстрируем, каким образом на основе объектов класса TAtObject организована БД проекта. БД проекта имеет несколько уровней иерархии, поэтому ее можно представить в виде ориентированного графа $G=(\Theta, R, H, L)$, множество узлов которого $\{\Theta_i\} \subset \Theta$ - множество объектов базовых или новых классов; множество дуг $\{r_j\} \subset R$ - множество связей между объектами; H - функция, связывающая элементы множеств Θ и R ; L - множество уровней $\{L_k\} \subset L$, показывающих детализацию объектов множества Θ .

Используются следующие типы связей r_j :

r_1 - показывает наследование свойств; r_2 - показывает отношение часть-целое между различными уровнями детализации объекта БД; r_3 - показывает передачу данных, которая в общем случае может указывать на передачу нескольких объектов БД; r_4 - указывает на передачу управления от одного объекта к другому.

Фрагмент такого графа выглядит следующим образом (рис. 1.):

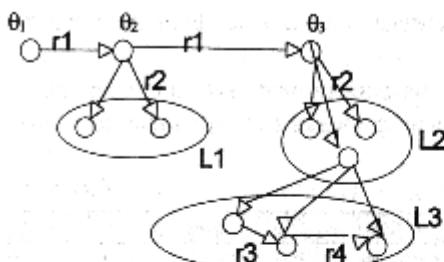


Рис. 1. Структура БД проекта (фрагмент)

сов (наследников от базовых классов), объектов, функций, методов и программных средств комплекса АТ-ТЕХНОЛОГИЯ, которые после регистрации становятся доступными на любом уровне детализации БД проекта, что, в отличие от MS DOS версии [Рыбина и др., 1997], позволяет ускорить процесс интеграции ПС в составе комплекса. Рассмотрим особенности такой регистрации.

Регистрация новых классов выполняется путем вызова функции RegisterClass(class: TClass). Перечислим некоторые из зарегистрированных классов:

TAtDialogFormEditor - класс редактора диалоговых форм;

TAtDiagramEditor - класс редактора диаграмм;

TAtKnowlegeElicit - класс ПС извлечения знаний;

TAtKBEdition - класс редактора базы знаний;

TAtTableEditor - класс редактора таблицы БД и др.

Новые объекты становятся доступными за счет применения функции регистрации RegisterObject (object: TControl). В качестве таких объектов могут выступать интерфейсные формы, диалоговые окна, и т.п.

Доступ к новым функциям обеспечивается после выполнения функции регистрации RegisterFunction (function: TFunction, FunctionName:string, ArgsType: array of PTypeInfo, ReturnType: PTypeInfo). Например: RegisterFunction (_LenStr_, 'ДлинаСтроки', [TypeInfo(string)], TypeInfo(Integer)), означает что функция _LenStr_ стала доступной в языке AtScript. Для получения ссылки на структуру данных, применяется функция TypeInfo(<тип>). Расширение набора доступных методов осуществляется путем вызова функции RegisterMethod (class: TClass, method: TMethod, FunctionName: string, ArgsType: array of PTypeInfo, ReturnType: PTypeInfo).

Следует отметить, что в отличие от MS DOS - версии комплекса АТ-ТЕХНОЛОГИЯ, в версии MS Windows предусмотрена возможность постоянного расширения библиотеки ПС. Такое расширение набора доступных ПС осуществляется путем вызова функции RegisterTool (Path: String, ComponentTool: TAtComponentTool; AppliedTo: TComponent).

θ_1 - некоторый объект БД проекта;

θ_2 - объект-наследник (тогда θ_1 - объект-прототип);

L₁, L₂ - уровни детализации объектов θ_1 и θ_2 , а r₁ ... r₄ - связи соответствующих типов;

Множество базовых классов прототипа ИЭС расширяется путем регистрации новых классов

Процесс планирования разработки в рамках комплекса АТ-ТЕХНОЛОГИЯ поддерживается Планировщиком - специальным объектом класса Tplanner. Укажем основные методы, определенные на уровне класса:

- создание списка ПС на основе информации о зарегистрированных компонентах, поддерживающих разработку текущей подсистемы, компонента, объекта БД проекта, входящих в состав прототипа ИЭС;
- выбор ПС с использованием формализованных алгоритмов, механизмов вывода, интерактивного взаимодействия с пользователем;
- загрузка, инициализация и выполнение выбранного ПС;
- модификация состояния проекта.

3. Особенности реализации программных средств, входящих в библиотеку комплекса АТ-ТЕХНОЛОГИЯ

3.1. ПС, применяемые на этапе анализа системных требований

На этапе анализа системных требований для редактирования расширенных диаграмм потоков данных (РДПД) [Рыбина и др., 1997] применяются редактор диаграмм (объект класса TAtDiagramEditor) и редактор свойств (объект класса TObjectInspector) следующих визуальных элементов, отображаемых на диаграмме: операция, неформализованная операция, внешняя сущность, накопитель данных, управляющая связь, связь по данным.

Для доопределения свойств отображаемых на диаграмме элементов используются специальные зарегистрированные ПС, расширяющие функциональность редактора диаграмм: средства пополнения словарей, извлечения знаний, определения структуры накопителей. При этом процесс взаимодействия редактора РДПД, редактора свойств визуальных элементов и ПС управляет планировщиком комплекса АТ-ТЕХНОЛОГИЯ.

3.2. ПС, применяемые на этапе общего проектирования прототипа

Редактор миниспецификаций (объект класса TAtProcessEditor) применяется также на этапе общего проектирования и, в общем случае, поддерживает определение миниспецификаций процессов на специальном визуальном языке АТ-Visual. Отметим следующие основные особенности данного языка:

- однородное использование функций и источников данных, представленных в виде объектов диаграммы;
- возможность 2D-представления описания процессов, облегчающее восприятие;
- встроенная (на уровне класса TObject) поддержка иерархического представления процедурных элементов языка, позволяющего описывать

миниспецификации процессов на различных уровнях абстракции;

- возможность определения множества элементов языка для миниспецификаций процессов различных типов (например, для миниспецификаций процессов, описывающих работу с распределенными объектами, используются элементы, поддерживающие COM/DCOM-интерфейсы; работу с БД - ODBC/IDAPI - интерфейсы; для описания миниспецификаций управляющих процессов - управляющий элемент, область данных, группа данных, условие выполнения; для описания сценариев диалога - диалоговая форма, условный элемент, управляющая связь).

Таким образом, на этапах анализа требований и общего проектирования формируются миниспецификации, описывающие РДПД - процессы нижнего уровня на языке AT-Visual, и определяются компоненты, входящие в состав прототипа ИЭС. Благодаря использованию собственных подмножеств элементов языка AT-Visual для описания семантически разнородных процессов удается эффективно реализовать интерпретацию процессов (список которых может пополняться) и локализовать и ограничить набор инструментов.

3.3. ПС, применяемые на этапе детального проектирования

На этапе детального проектирования, используя редактор диалоговых форм (объект класса TAtDialogEditor) и редактор свойств (объект класса TObjectInspector), определяются взаимосвязи между компонентами прототипа ИЭС, описанными на этапе общего проектирования. С помощью этих редакторов реализуются функции создания диалоговых форм, определения интерфейсных элементов, описания взаимосвязей между интерфейсными элементами, описания событий и обработчиков событий, доопределения методов объектов. Редактор диалоговых форм обращается к отдельным зарегистрированным ПС, поддерживающим разработку элементов, привязанных к формам, и методов, описывающих процессы функционирования этих элементов.

Для описания методов объектов разработан язык AT-Script, сокращенный синтаксис которого представлен ниже:

```

<ОписаниеМетодовОбъекта> ::= {<Метод>}
<Метод> ::= 'function' <ИмяМетода> '(' [<ОписПарам>] ')' <Тело>
<ОписПарам> ::= <ОписанПерем> {','<ОписанПерем>}
<ОписанПерем> ::= <ИмяПеременной> ':' <ИмяТипа>
<Тело> ::= <Присваивание> | <Блок>
<Присваивание> ::= [<Операнд>] '=' <Выражение>
<Операнд> ::= <Идент> | <ИмяОбъекта>. <ИмяСвойства> | <ИмяКонтейнера> |
<ИмяОбъекта> | <Const>
<Идент> ::= <ИмяПеременной> | <ИмяТипа>
<Const> ::= Целое | Дробное | Строкा
<Выражение> ::= <Выражение> '+' <Т> | <Выражение> '-' <Т> | <Т>
<Т> ::= <Т> '*' <Ф> | <Т> '/' <Ф> | <Ф>

```

```

<Г> ::= '(' <Выражение> ')' | <ОбщОперанд>
<ОбщОперанд> ::= [<Операнд> ','] <Функция>
<Функция> ::= <Имя> '(' [<Аргументы>] ')'
<Аргументы> ::= <Выражение> ',' <Выражение>
<Блок> ::= <Операция> | 'Г' {<Операция>} '}'
<Операция> ::= <Присваивание> ';' <Цикл> | <Условие> | <Возврат>
<Комментарий> ::= ';' | <ОписанПарам> ';' | <Локальные-переменные>
<Цикл> ::= 'While' <Выражение> 'Do' <Блок> ;;
<Условие> ::= 'If' <Выражениc> 'Then' <Блок> ['Else' <Блок>] ;;
<Возврат> ::= 'Возврат' [<Выражение>] ;;
<Комментарий> ::= '' <Строка>;
<Локальные-переменные> ::= 'Local' <ОписанПарам> ;'

```

ПС библиотеки комплекса используют различные расширения языка AT-Script, опирающиеся на зарегистрированные функции и методы, описанные на языках низкого уровня (например, Object Pascal фирмы Borland Delphi). Например, в качестве расширения AT-Script используется язык описания сценария диалога для извлечения знаний из эксперта. Приведем некоторые функции этого языка:

```

<Операция описания сценария диалога>::=
CREATE_FORM(<ИмяФормы>)
ASK_EXPERT(<ИмяФормы>, <ИмяПеременной>)
CREATE_OBJECT(<ИмяПеременной>, <Имя объекта поля знаний>)
CREATE_ATTRIBUTE(<ИмяПеременной>, <Имя объекта поля знаний>, <Имя атрибута>)
CREATE_RULE(<Имя правила>) | др. операции.

```

Компоненты проекта, даже если они относятся к различным уровням детализации БД проекта, могут быть связаны посредством специальным образом описанных объектов-интерфейсов. Выделены два типа объектов-интерфейсов: ссылки на объекты, расположенные ниже в иерархии "общее-частное" и ссылки-параметры, используемые объектами нижних уровней детализации. После регистрации (раздел 2) интерфейсы становятся опубликованными свойствами класса, "понимаемыми" редактором свойств объектов.

3.4. ПС, применяемые на этапе кодированная и тестирования прототипа

Благодаря особой организации комплекса АТ-ТЕХНОЛОГИЯ описание на языках AT-Visual и AT-Script может быть интерпретировано специальным ПС комплекса, которое поддерживает процесс тестирования текущего прототипа ИЭС и обеспечивает демонстрацию работоспособности прототипа и отдельных его компонент, даже если они еще не полностью спроектированы.

4. Заключение

Комплекс АТ-ТЕХНОЛОГИЯ ориентирован на использование персональных компьютеров типа РС 486 DX2-66 и выше с оперативной памятью не менее 16Мб. Общий объем занимаемого дискового пространства

составляет 8-10 Мб. В числе особенностей реализации комплекса можно выделить обеспечение доступа к разнородным источникам данных, доступа к внешним DLL-модулям, внешним объектам.

Работа поддерживается РФФИ РАН (грант № 98-01-00918).

Л и т е р а т у р а

[Калянов, 1997] Калянов Г.Н. Консалтинг при автоматизации предприятий. Подходы, методы, средства. - М.: СИНТЕГ, 1997.

[Рыбина Г.В., 1997] Рыбина Г.В. Задачно-ориентированная методология автоматизированного построения интегрированных экспертных систем для статических проблемных областей. Известия РАН. Теория и системы управления. 1997, № 5.

[Рыбина Г.В. и др., 1997] Рыбина Г.В., Пышагин С.В., Смирнов В.В., Чабас А.В. Программные средства и технология автоматизированного построения интегрированных экспертных систем. Программные продукты и системы. 1997, № 4.

NeMo+: ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ СИСТЕМА ПРОГРАММИРОВАНИЯ В ОГРАНИЧЕНИЯХ НА ОСНОВЕ НЕДООПРЕДЕЛЕННЫХ МОДЕЛЕЙ

Г. Б. Загорулько¹, В. А. Сидоров¹,
В. В. Телерман¹, Д. М. Ушаков¹

В данной работе описывается система NeMo+, основанная на аппарате недоопределенных моделей. В этой системе интегрируются две парадигмы: программирование в ограничениях и объектно-ориентированное программирование. Приводится краткое описание и упрощенная схема работы системы NeMo+. В докладе обсуждаются особенности языка спецификации моделей. Технология решения задач средствами NeMo+ демонстрируется на примере проектирования кресла водителя легкового автомобиля. В заключении приводятся основные направления дальнейшего развития системы NeMo+.

1. Введение

Система NeMo+ — это объектно-ориентированная среда программирования в ограничениях, предназначенная для решения широкого спектра прикладных задач. Программирование в ограничениях относится к на-

¹ 630090, Новосибирск, пр. Лаврентьева 6, РосНИИ искусственного интеллекта gal@iis.nsk.su, sidorov@iis.nsk.su, telerman@iis.nsk.su, ushakov@iis.nsk.su