

УДК 681.51.01(035.5)

## ОБЪЕКТНО-РЕЛЯЦИОННАЯ РЕАЛИЗАЦИЯ ИЕРАРХИЧЕСКИХ СИТУАЦИОННЫХ МОДЕЛЕЙ В ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

В. В. МИРОНОВ, А. Н. СИТЧИХИН, Р. Ф. АХМЕТШИН

Факультет информатики и робототехники УГАТУ  
Тел: (3472) 23 78 23 E-mail: mironov@ugatu.ac.ru

Рассматриваются вопросы реализации иерархических ситуационных моделей (ИСМ) в вычислительном комплексе системы управления на основе сочетания объектного и реляционного подходов. Описываются диаграммы классов объектов внешнего представления ИСМ и структура базы данных модели текущего состояния. Приводятся результаты практической реализации ИСМ в среде СУБД Visual FoxPro

*Дискретно-событийные, иерархические, ситуационные, объектные, реляционные модели*

### ВВЕДЕНИЕ

Системное моделирование сложных объектов и процессов основано на использовании различных классов дискретно-событийных моделей: сетей Петри, графов переходов, диаграммы состояний и т. п. Одним из них является класс иерархических ситуационных моделей (ИСМ), позволяющий задавать и интерпретировать в процессе управления иерархически организованную совокупность ситуаций и условий переходов одних ситуаций в другие [1].

Разработанные к настоящему времени версии ИСМ, ориентированные на применение в технических системах (летательных аппаратах), основаны на достаточно простых внешних (на основе проверки сигналов датчиков) и внутренних (на основе проверки текущего состояния модели) предикатах переходов ситуаций.

Распространение ИСМ на задачи моделирования организационного управления и поддержки принятия решений требует использования в качестве внешних предикатов решений, принимаемых человеком. Внутренние предикаты должны учитывать предысторию (ретроспекцию) смены ситуаций. Как предикаты, так и акции должны позволять достаточно просто взаимодействовать с базами данных системы управления. Все это усложняет реализацию ИСМ в вычислительной среде (как и других классов дискретно-событийных моделей, выполняющих те же функции) и требует новых подходов и решений в этой области.

Для преодоления указанных трудностей в работе [2] был предложен класс ретроспективных ИСМ, сохраняющих предысторию смены ситуаций, а в работе [3] — объектно-ориентированный подход к построению ИСМ, в основе которого лежит идея децентрализованной самоинтерпретации отдельных элементов модели. В настоящем сообщении рассматривается путь практической реализации этих идей на основе сочетания объектного и реляционного подходов.

### 1. ПРЕДЛАГАЕМЫЙ ПОДХОД К РЕАЛИЗАЦИИ ИСМ

В рамках реализации ИСМ необходимо проработать вопросы реализации:

- исходной модели (ИМ), позволяющей на этапе проектирования автоматизированной системы задавать иерархию возможных ситуаций и условия их смены;

- модели текущего состояния (МТС), позволяющей на этапе функционирования автоматизированной системы осуществлять контроль этих управленческих ситуаций на основе претации ИСМ;

- интерфейса взаимодействия ИСМ с внешней и внутренней средой, позволяющего формировать управление и принимать решения в различных

Таким образом, реализация ИСМ в некоторой предметной области при помощи ИСМ. Рассматриваемый класс ИСМ ориентирован на поддержку принятия решений в организационных системах, поэтому необходимо обеспечить взаимодействие с базами данных и с пользователями, принимающими решения.

Реализация объектных ИСМ, взаимодействующих с прикладными базами данных, естественно решается при использовании объектно-ориентированных баз данных в качестве текущего состояния. Однако в настоящее время объектно-ориентированные базы имеют существенно меньшее количество точек зрения реализации ИСМ по сравнению с реляционными. ИСМ на в среде баз данных реляционного типа

Предлагаемый подход к реализации ИСМ основан на следующих идеях:

- ИМ следует реализовать в объектно-ориентированной среде в соответствии с современными тенденциями развития ИСМ;

- МТС следует реализовать в реляционной форме в виде некоторой реляционной базы данных, состояние которой отражает текущее состояние ИСМ и ретроспекцию ее ситуаций;

- интерфейс взаимодействия с внешней средой следует реализовать в смешанной объектно-реляционной форме (на основе объектных экранных форм и меню, а также на основе реляционных запросов к базам данных).

В результате внешние предикаты ИСМ смогут через запросы обращаться к внешним базам данных для проверки условий функционирования системы или через экранный интерфейс учитывать решения пользователей, а внутренние предикаты аналогичным образом смогут обращаться к базе данных МТС для проверки текущих ситуаций или их ретроспекции. Процедуры-акции ИСМ смогут подобным же образом изменять внешние базы данных или выводить информацию пользователю.

Отметим, что реляционная реализация МТС не вполне соответствует принципам объектной реализации ИСМ, изложенным в [3] и предполагающим, что самоинтерпретация ИСМ выполняется объектами МТС. Объектная модель предполагает инкапсуляцию в объектах определенных свойств (данных) и методов (процедур обработки данных). Реляционная модель, очевидно, может воспроизвести лишь свойства объектов МТС. Следовательно, функции, возлагавшиеся на методы объектов МТС, необходимо реализовать методами объектов ИМ, обеспечив при этом эквивалентность интерпретации.

## 2. РЕАЛИЗАЦИЯ МОДЕЛИ ТЕКУЩЕГО СОСТОЯНИЯ

Первый вопрос, возникающий при разработке принципов реализации МТС, это вопрос об однозначной идентификации объектов МТС. Каждому объекту МТС соответствует единственный объект ИМ. Обратное утверждение, вообще говоря, неверно: в ряде случаев (массивы субмоделей, рекурсивные и ретроспективные субмодели и др.) одному объекту ИМ (например, ситуации) может соответствовать несколько объектов в МТС. Поэтому для идентификации этих объектов недостаточно уникальных имен (или других идентификаторов) соответствующих объектов ИМ, а требуется некоторая дополнительная информация.

Для обеспечения однозначной идентификации введем понятие версии объекта МТС. Пусть  $A_n$  — идентификатор некоторого объекта ИМ. Под версией  $V_A$  будем понимать натуральное число, означающее номер экземпляра объекта МТС, соответствующего объекту  $A_n$ .

Следовательно, идентификатор текущего объекта  $A_T$  представляет собой пару  $A_T = (A_n, V_A)$ , где  $A_n$  — идентификатор соответствующего исходного объекта;  $V_A$  — версия данного текущего объекта. Значение версии формируется автоматически при порождении очередного экземпляра объекта в МТС. Наряду с идентификатором исходного объекта версия сохраняется в качестве

свойства текущего объекта. При порождении первого экземпляра объекта  $V_A = 1$ , при порождении следующего экземпляра его версия устанавливается на единицу больше, и так далее.

Конкретный объект МТС представляется в базе данных МТС (БД МТС) в виде строки (кортежа) в таблице, соответствующей классу этого объекта. Объекты-ситуации хранятся в таблице Sit, объекты-переходы — в таблице Jump, объекты-погружения — в таблице Dive и т. д. Структура таблиц БД МТС (состав атрибутов) соответствует функциональному назначению объектов и включает как общие, так и специфические атрибуты (рис. 1).

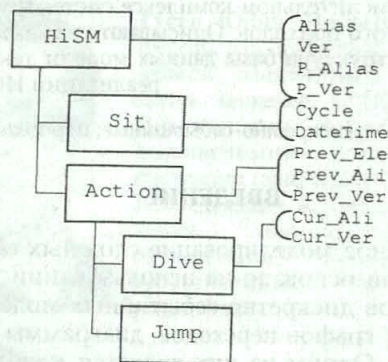


Рис. 1. Иерархия наследования атрибутов объектов модели текущего состояния

Все таблицы БД МТС содержат идентификатор объекта  $A_T = (A_n, V_A)$ , а также идентификатор родительского объекта  $A_T^P = (A_n^P, V_A^P)$ . Для объекта-ситуации родительским объектом является объект-погружение в данную субмодель, а для объектов-действий (переходов, погружений и т. д.) — объект-ситуация, с которым ассоциированы эти действия.

Объект-ситуация дополнительно содержит временную метку  $T$  типа «Дата-время», фиксирующую момент создания этого объекта (т. е. момент, когда данная ситуация стала текущей). Кроме того, здесь содержится идентификатор цикла интерпретации и другая служебная информация.

Объект-действие дополнительно содержит тип и идентификатор предшествующего объекта  $A_T^N = (A_n^N, V_A^N)$ , связывающие его с предшествующим действием в упорядоченном списке действий, ассоциированных с ситуацией.

Объект-погружение дополнительно содержит идентификатор текущей ситуации  $V_T^C = (A_n^C, V_A^C)$ , указывающий на текущую ситуацию субмодели, которую задает это погружение.

В целом, указатели-идентификаторы в БД МТС имеют обратную направленность по отношению к указателям исходной модели, что в совокупности дает возможность двунаправленной навигации в ходе интерпретации ИСМ, в частности, при вычислении значений внутренних предикатов.

### 3. РЕАЛИЗАЦИЯ ИСХОДНОЙ МОДЕЛИ

ИМ строится на этапе проектирования из объектов-экземпляров соответствующих классов. Разработанная в рамках реализации ИМ иерархия классов, отражающая наследование их свойств и методов, приведена на рис. 2.

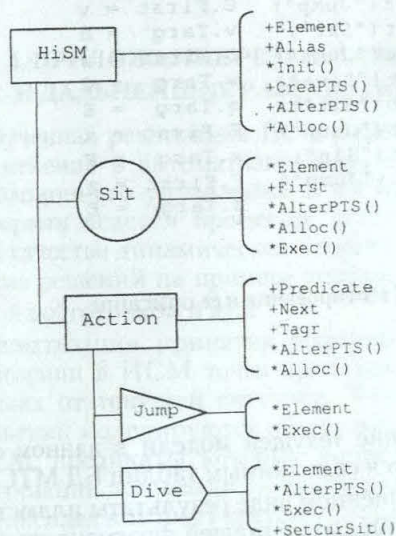


Рис. 2. Иерархия наследования свойств и методов в объектах исходной модели

Класс HiSM расположен на верхнем уровне иерархии и является базовым для остальных классов объектов ИМ ИСМ. В нем инкапсулированы свойства и методы, необходимые всем его потомкам. В частности, свойство Element хранит тип объекта, а свойство Alias — идентификатор объекта  $A_i$ . Метод CreaPTS() создает структуру таблицы (в случае ее отсутствия в БД МТС) для хранения текущих объектов, соответствующих исходному (создаются столбцы, присутствующие в таблицах всех типов объектов, см. рис. 1). Метод Alloc() размещает в БД МТС экземпляр текущего объекта, соответствующий исходному объекту. Для этого вычисляется версия объекта и в таблицу БД МТС заносится соответствующий кортеж. Метод Exec() выполняет интерпретацию объекта (на верхнем уровне иерархии это шаблон, который уточняется на нижестоящих уровнях).

Класс Sit соответствует объектам-ситуациям и порождается на базе класса HiSM. Дополнительное свойство First предназначено для хранения указателя на первый объект списка действий, ассоциированных с данной ситуацией. Дополнительный метод AlterPTS() добавляет в структуру таблицы Sit в БД МТС новые столбцы, специфичные для объекта-ситуации (в других объектах этот метод выполняет аналогичные функции для соответствующих таблиц БД МТС, см. рис. 1). Модифицированный метод Alloc() при размещении текущего объекта-ситуации дополнительно размещает и связывает в список текущие объекты-действия, ассоциированные с этой ситуацией. Модифицированный метод Exec() в ходе интерпретации ситуации осуществляет последовательный за-

пуск интерпретации ассоциированных объектов-действий.

Класс Action (промежуточный) соответствует объектам-действиям и базируется на классе HiSM. Дополнительное свойство Predicate предназначено для хранения ссылки на предикат, управляющий активностью действия, а дополнительное свойство Targ — ссылки на целевой объект, который необходимо интерпретировать в случае истинности предиката. Модифицированный метод Alloc() при размещении текущего объекта-действия включает его в список действий, ассоциированных с родительской ситуацией.

Класс Dive соответствует объектам-погружениям и базируется на классе Action. Дополнительный метод SetCurSit() устанавливает указанную ситуацию в качестве текущей ситуации данной субмодели (атрибуты Cur\_Ali и Cur\_Ver таблицы Dive). Метод Exec() выполняет интерпретацию погружения, в ходе которой вычисляет значение предиката и в случае его истинности вызывает:

- метод Alloc() (при необходимости) для размещения текущего объекта-погружения в таблице Dive БД МТС;

- метод Alloc() (при необходимости) для размещения текущего объекта-ситуации, соответствующего начальной ситуации субмодели (свойство Targ), в таблице Sit БД МТС и устанавливает эту ситуацию в качестве текущей путем вызова метода SetCurSit();

- метод Exec() текущей ситуации субмодели, запуская тем самым ее интерпретацию.

Класс Jump соответствует объектам-переходам и базируется на классе Action. Метод Exec() выполняет интерпретацию перехода, в ходе которой вычисляет значение предиката и в случае его истинности вызывает:

- метод Alloc() целевой ситуации перехода (задаваемой свойством Targ) для размещения текущего объекта-ситуации в таблице Sit БД МТС;
- метод SetCurSit() целевой ситуации для новки этой ситуации в качестве текущей;
- метод Exec() целевой ситуации, запуская тем самым ее интерпретацию.

### 4. РЕАЛИЗАЦИЯ В СРЕДЕ Visual

Необходимый для реализации элемент технических возможностей (запросы к реляционным интерфейсам с пользователями) и современные инструменты тирования приложений (VBA, C+), работа в качестве платформы инструментальной среды Microsoft (VFP), в полной мере удовлетворяют требованиям.

В этой среде МТС (см. рис. 1) в виде базы данных VFP, а иерархия объектов (см. рис. 2) — на основе невидимого топ из набора базовых классов VFP. На приведен пример простой двухуровневой И.

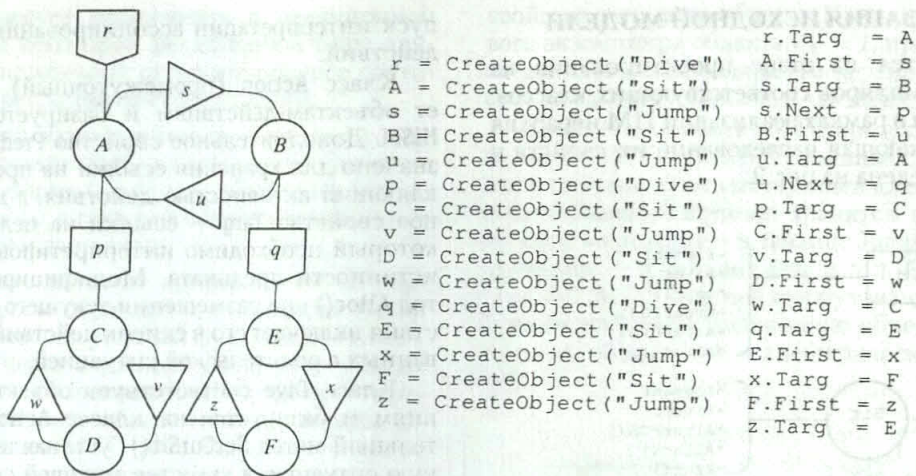


Рис. 3. Двухуровневая исходная модель для проведения тестирования и ее описание в среде Visual FoxPro

с фрагментом программного кода, задающего эту модель в среде VFP.

Запуск очередного цикла интерпретации модели производится путем вызова метода интерпретации корневого объекта-погружения: `r.Exec(1)`. В качестве параметра передается значение первой версии текущего экземпляра погружения.

Тестирование модели проводится по методике многоуровневой верификации ИСМ [4], предусматривающей следующие этапы:

- составление сценария переходов ситуаций и соответствующей активности предикатов переходов и погружений;
- построение эталонной циклограммы состояний текущей модели в соответствии с составленным сценарием и правилами интерпретации;
- сравнение эталонной циклограммы с действительной, полученной при интерпретации реализации модели.

Состояние текущей модели в данном случае определяется содержимым таблиц БД МТС.

В укрупненном виде результаты иллюстрируются в таблице, содержащей фрагмент эталонной циклограммы. На первом цикле интерпретации модель (корневое погружение *r*) пассивна; цикл 2 соответствует текущей ситуации *A* при пассивной субмодели *p*; на цикле 3 субмодель *p* активизируется с текущей ситуацией *C*; цикл 4 соответствует переходу субмодели *p* в ситуацию *D*, а цикл 5 — возвращению в ситуацию *C*; на цикле 6 происходит переход корневой модели в ситуацию *B* при активной субмодели *q* с текущей ситуацией *E*; цикл 7 — переход субмодели *q* в ситуацию *F*; цикл 8 — возврат корневой модели в ситуацию *A* при активной субмодели *p* с текущей ситуацией *C*. Для краткости в таблице опущены промежуточные изменения состояния на отдельных шагах цикла интерпретации и приведены лишь конечные состояния для каждого цикла.

Фрагмент эталонной циклограммы тестирования модели

Таблица

Цикл	Значения предикатов									Содержимое таблиц модели текущего состояния									
	r	p	q	s	u	v	w	x	z	Dive			Sit			Jump			
										Запись	A <sub>r</sub>	A <sub>r</sub> <sup>p</sup>	A <sub>r</sub> <sup>s</sup>	Запись	A <sub>r</sub>	A <sub>r</sub> <sup>p</sup>	Запись	A <sub>r</sub>	A <sub>r</sub> <sup>p</sup>
1	0									1	r,1	—	—						
2	1	0								1	r,1	—	A,1	1	A,1	r,1	1	s,1	A,1
3	1	1	0		0					2	p,1	A,1	C,1	2	C,1	p,1	2	v,1	C,1
4	1	1	0		1	0				2	p,1	A,1	D,1	3	D,1	c,1	3	w,1	D,1
5	1	1	0		0	1				2	p,1	A,1	C,2	4	C,2	p,1	4	v,2	C,2
6	1		1	1	0			0		1	r,1	—	B,1	5	B,1	r,1	5	u,1	B,1
7	1		1		0			1	1	3	q,1	B,1	E,1	6	E,1	q,1	6	x,1	E,1
8	1	1	1	0	1	0		0		1	r,1	—	A,1	7	F,1	q,1	7	z,1	F,1
										4	p,2	A,2	C,3	8	A,2	r,1	8	s,2	A,2
														9	C,3	p,2	9	v,3	C,3

Результаты тестирования показали соответствие реального поведения модели и эталонной циклограммы для всех сценариев, охватывающих существенные варианты развития ситуаций. Это позволяет сделать вывод о корректности как общего подхода объектно-реляционной реализации ИСМ, так и конкретного его воплощения в среде VFP.

### 5. ПРИЛОЖЕНИЕ РЕЗУЛЬТАТОВ И ДАЛЬНЕЙШИЕ РАЗРАБОТКИ

Полученная реализация ИСМ ориентирована на применение в автоматизированных системах организационного управления. В настоящее время авторами ведется проект по использованию ИСМ в качестве динамических моделей в системе принятия решений на примере управления магистерской подготовкой в вузе.<sup>1</sup>

Автоматизация принятия решений основана на выделении в ИСМ точек принятия решений, зависящих от текущей ситуации. Точки принятия решений моделируются объектами-переходами ИСМ, предикаты которых вычисляются на основе реакции пользователя на отображаемый на экране дисплея контекст текущей ситуации, определяемый текущим состоянием ИСМ. При этом пользователю отображаются рекомендуемые варианты решений, которые также формируются в зависимости от состояния ИСМ.

В рамках этих исследований предполагается также разработка лингвистического обеспечения, упрощающего процессы визуального построения исходных моделей и трансляции их в объектную форму инструментальной среды (см. рис. 3).

### ЗАКЛЮЧЕНИЕ

1. В основе предложенной реализации иерархических ситуационных моделей, ориентированных на применение в системах автоматизированного принятия решений организационного управления, лежат идеи объектной реализации исходной модели и реляционной реализации модели текущего состояния, что должно обеспечить достаточно простой интерфейс с пользователем и внешними базами данных.

2. Реляционная реализация модели текущего состояния основана на ведении базы данных состояния модели, в таблицах которой каждый текущий объект идентифицируется парой: идентификатор соответствующего исходного объекта — версия текущего объекта.

3. Объектная реализация исходной модели включает иерархию классов объектов, свойства и методы которой обеспечивают ведение базы данных модели текущего состояния в процессе функционирования и позволяют ее корректно интерпретировать.

4. Результаты тестирования модели свидетельствуют о корректности как общего подхода объектно-реляционной реализации ИСМ, так и конкретного его воплощения в среде Visual FoxPro.

5. Исследования предполагается продолжить в рамках проекта разработки динамических моделей процессов организационного управления и принятия решений.

### СПИСОК ЛИТЕРАТУРЫ

1. **Yusupova N. I., Mironov V. V.** Hierarchical situational models and linguistic means their realization // Proc. of World Multiconference on Systemics, Cybernetics and Informatics and 5th Int. Conf. on Information Systems Analysis and Synthesis. Vol. 7. July 31 – Aug 4, 1999. Orlando, Florida. P. 233–236.
2. **Миронов В. В., Ситчихин А. Н.** Иерархические ситуационные модели с предьсторией // Управление в сложных системах: Межвуз. науч. сб. Уфа: УГАТУ, 1999. С. 55–68.
3. **Mironov V. V., Akhmetshin R. F., Yussupova N. I.** Hierarchical situational models with decentralized interpreting // Computer Science and Information Technologies (CSIT'2000): Proc. of the 2nd Int. Workshop. Ufa, Russia, 2000. Vol. 3. P. 29–31.
4. **Юсупова Н. И., Сметанина О. Н., Миронов В. В.** Об обеспечении помехоустойчивости иерархических ситуационных моделей управления // Управление в сложных системах: Межвуз. науч. сб. Уфа: УГАТУ, 1998. С. 115–138.

### ОБ АВТОРАХ



**Миронов Валерий Викторович** профессор кафедры автоматизированных систем УГАТУ. Дипл. радиотехнический гос. ун-т, техн. наук по управлению техническими системами (УГАТУ, 1995). Исследования моделей критических ситуаций и ситуационного управления в сложных системах.



**Ситчихин Андрей Николаевич**, аспирант той же кафедры. Дипл. инж. по информационным системам (УГАТУ, 1998). Готовит диссертацию о ретроспективных иерархических ситуационных моделях.



**Ахметшин Радик Фагимович** аспирант той же кафедры инж. п.с. системам диссертацию о автоматизированных и иерархических ситуационных моделях.

<sup>1</sup> В исследованиях принимает участие аспирант кафедры АСУ И. Р. Валиахметов.