

ОБЪЕДИНЕННЫЙ  
ИНСТИТУТ  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ  
ДУБНА

К 672

P5-87-590

А.А.Корнейчук, В.Шуберт, Х.Юнгклауссен

ОБОБЩЕННЫЕ СРЕДСТВА  
ОБЪЕКТНО-ОРИЕНТИРОВАННОГО  
ПРОГРАММИРОВАНИЯ

Направлено на семинар "Проблемы и применения  
искусственного интеллекта", Варна, сентябрь 1987 г.

1987

## I. Принципы объектно-ориентированного программирования

В настоящее время среди проблем искусственного интеллекта важную роль играет следующий вопрос: как можно повысить интеллектуальный уровень общения между человеком и ЭВМ и какие языковые средства нужны для того, чтобы ЭВМ стала достойным помощником человека в процессе решения задач, насыщенных машинной обработкой информации? В ходе исследований в этом направлении сложилось понятие об объектно-ориентированном моделировании и программировании. Важными этапами развития этого метода являются

- концепция класса (**class**) языка SIMULA 67 /1/,
- система программирования Dynabook /2/,
- концепция сообщения (**message**) и разделения труда при решении проблем /3/,
- система программирования Smalltalk /4/,
- объектно-ориентированные диалекты традиционных языков программирования /5/.

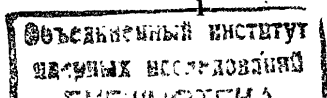
Именно с последней концепцией связана наша попытка интеграции более или менее стихийно сложившихся до сих пор методов и средств объектно-ориентированного программирования. Будем исходить из следующих принципов и понятий /6/.

1. Объектом (**object**) называется именованная единица объектно-ориентированной системы, обладающая некоторым "интеллектом" (способностью реагировать на воздействие извне) и являющаяся моделью некоторой части реального мира.

2. Каждый объект является представителем (примером, **instance**) некоторого класса (**class**). Объект ведет себя согласно предписаниям, содержащимся в определении данного класса.

3. Представители одного класса могут принимать определенные сообщения (**messages**) и применять определенные методы обработки (**methods**) сообщений. Каждый класс характеризуется своими сообщениями и методами обработки.

4. Классы образуют иерархию. Если вводится новый класс как подкласс некоторого более общего класса (суперкласса, **superclass**), то он унаследует от своего суперкласса все его свойства. В системе Smalltalk осуществление указанных общих принципов поддерживается некоторыми специфическими принципами программирования, получившими название (в том же порядке): 1) упрятывание информации



(information hiding) ; 2) абстракция данных (data abstraction) ; 3) динамическое связывание (dynamic binding) ; 4) наследование свойств (inheritance) .

Программирование в рамках объектно-ориентированных систем состоит в определении классов и иерархии классов, в генерировании представителей этих классов и в пересылке сообщений представлениям.

## 2. Статическое и динамическое моделирование

При моделировании надо различать статические и динамические аспекты оригинала, т.е. описание свойств, присущих оригиналу, и описание изменений этих свойств, включая описание ситуаций, в которых эти изменения происходят. Соответственно будем говорить о статическом и динамическом моделировании. Примерами этих двух типов моделирования являются создание базы фактов, с одной стороны, и создание базы правил вывода (правил продукций) экспертной системы, с другой стороны <sup>/7/</sup>. Сети Петри являются примером чисто динамических моделей. Динамическое моделирование характерно и для создания систем управления.

При естественном моделировании человек использует определенные отношения и операции (функции) для описания окружающей его действительности. Наиболее важные из них перечислены в табл. I. Не указано отношение имя - именуемое, без которого словесное моделирование вообще невозможно, а только о словесном моделировании здесь идет речь. Значительно более подробная таблица приведена в <sup>/8/</sup>.

Таблица I. Отношения и операция статического и динамического моделирования. В скобках - термины, используемые системой Smalltalk

Статическое моделирование:

- отношение часть-целое;
- отношение элемент-класс (instance of);
- отношение подкласс-суперкласс (subclass of).

Динамическое моделирование:

- отношение оператор-операнд, отношение воздействия;
- операция преобразования (method), оператор преобразует операнд в результат.

## 3. Формальное определение понятия объекта

Достаточно полная модель должна отражать статические и динамические аспекты оригинала. Этому требованию отвечают семиотические модели, введенные Поспеловым в <sup>/8/</sup>. Поэтому в целях определения по-

нятия объекта в смысле первого принципа объектно-ориентированного программирования (см. п.1), выберем (по примеру <sup>/9/</sup>) специальную форму семиотических моделей, основанную на исчислении предикатов I-го порядка и определим объект как интерпретированную элементарную теорию (см. <sup>/10/</sup> и <sup>/11/</sup> под термином "Алгебраическая система"). Элементарная теория - это пятерка  $T = (M, R, O, C, A)$ , где буквы в скобках - символы, обозначающие множества элементов разного типа, а именно: M - основное множество (или множество таких множеств, так называемых сортов), R - множество отношений, O - множество операций, C - множество постоянных, A - множество аксиом. Такой подход подсказывается тем обстоятельством, что при словесном (языковом) моделировании свойства оригинала описываются посредством отношений и операций (см. табл. I). Кроме того, подобный подход, хотя в более простой форме, успешно применяется для алгебраического определения абстрактных типов данных <sup>/12/</sup>.

Интерпретацией теории называется сопоставление символам конкретных элементов, так что каждому высказыванию теории T соответствует или конкретное, не подлежащее дальнейшей интерпретации высказывание, или высказывание более конкретной теории T'. Во втором случае говорят, что теория T' интерпретирует теорию T. Таким образом, можно сформулировать следующие положения объектно-ориентированного моделирования:

1. Объект определяется как интерпретированная элементарная теория.

2. Статическое моделирование осуществляется посредством статических объектов, т.е. объектов, определяемых как элементарная теория, не содержащая операций.

3. Динамическое моделирование осуществляется посредством динамических объектов, т.е. троек (начальный статический объект, операция преобразования, конечный статический объект).

## 4. Уровни моделирования

Для человеческого мышления, в частности для мысленного моделирования сложных оригиналов, например сложных технических систем, характерно пошаговое обобщение и абстракция от деталей. Согласно задаче, сформулированной в начале статьи, желательнее осуществить метод обобщения и в рамках объектно-ориентированного моделирования. Формально это нетрудно сделать, используя возможность пошаговой интерпретации, означающей пошаговую конкретизацию и детализацию. На рис. I показана, в качестве примера, трехуровневая структура. На каждом уровне используются для моделирования объекты определенного уровня обоб-

нения. Для обозначения уровней обобщения Месарович ввел слово "страта" (*stratum*) <sup>/13/</sup>. Многоуровневая структура рис. I нарисована по образцу так называемых сетей путей решения проблем или путей проектирования технических систем, введенной Херригом <sup>/14/</sup>. Таким образом, в наши первоначальные представления вкладывается новый смысл, а именно: уровни моделирования истолковываются как уровни решения проблем, в частности как уровни проектирования, а это как раз отвечает нашей исходной задаче. Стрелки на рис. I показывают возможные пути от задачи к решению. Чем больше число прошедших уровней (это число в принципе неограничено), тем выше затраты на первичное проектирование, но тем прозрачнее структура проекта-результата и тем ниже затраты на его изменения, а также на проектирование подобных систем. Опыт системы *Smalltalk* показал, что достаточно трех уровней. Такое число подсказывается и таблицей I, содержащей три отношения для статического моделирования. Они могут быть положены в основу определения объектов трех типов и соответствующих уровней моделирования (проектирования). Для этих объектов введем названия сеть (*net*), класс (*class*) и абстракт (*abstract*) и будем говорить об уровне сетей, уровне классов и уровне абстрактов.

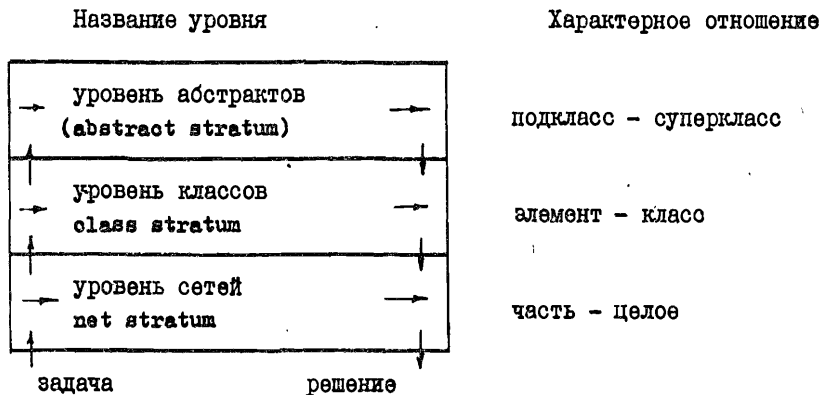


Рис. I. Уровни моделирования (проектирования).

Поясним смысл отдельных уровней и характер соответствующих объектов и моделей. На нижнем уровне статическое моделирование состоит в создании сетей и иерархий операторов и операндов <sup>/15/</sup>. Структура иерархии задается отношением часть-целое. Оператор любой степени сложности является объектом уровня сетей. Он может интерпретироваться фактами реальной действительности или средствами некоторого язы-

ка программирования. Это относится и к динамическому моделированию, которое описывает изменения операторов и операндов, изменения отношения часть-целое и изменения отношения воздействия (при интерпретации программой, например, применением семафоров для управления доступом к ресурсам).

На среднем уровне используется отношение элемент-класс. Объект этого уровня, т.е. класс, является обобщением ряда однотипных, но не обязательно идентичных, элементов. Если для создания модели на нижнем уровне требуется некоторый элемент (например, оператор) в нескольких экземплярах, то достаточно на среднем уровне определить соответствующий класс и реализовать некоторое число его представителей. Если, например, классу соответствует в программной интерпретации процедура, то реализация представителя класса - это вызов процедуры.

Если объект среднего уровня моделирует класс операторов, то в определении объекта можно включить определение класса операндов, с которыми "работают" операторы <sup>/16/</sup>. Если, кроме того, разрешить доступ к операндам только операторам данного объекта, то приходим к понятию абстрактного типа данных <sup>/12/</sup>. Если для подобных объектов вводится отношение воздействия, то приходим к понятию актора (*actor*) <sup>/3/</sup>.

На верхнем уровне строятся иерархии классов путем обобщающей абстракции. Структура иерархии задается отношением подкласс-суперкласс. Суперкласс является гомоморфным отображением своих подклассов. Рационализирующий эффект при моделировании достигается тем, что свойства классов автоматически передаются по иерархии сверху вниз, так сказать, в качестве наследия (*inheritance*) от старших к младшим представителям иерархии. При введении нового класса в иерархию достаточно указать лишь те свойства, по которым различаются подклассы соответствующего суперкласса. К динамическому моделированию на верхнем уровне относится описание любых изменений структуры иерархии.

### 5. Заключение

Сравнение выводов предыдущей главы с перечнем признаков объектно-ориентированного программирования, приведенным в I-й главе, показывает, что все эти принципы являются элементами или свойствами метода трехуровневого моделирования.

Укажем на то, что все средства моделирования, о которых шла речь, т.е. все объекты как интерпретированные элементарные теории, можно представить в виде предложенный языка DEKART <sup>/17/</sup>.

В настоящее время в ОИЯИ (Дубна) начались работы по реализации метода трехуровневого моделирования в рамках Ф-технологии - системы программирования, основанной на языке FORTRAN /18/.

#### Литература

- I. Андрианов А.Н., Бычков С.П., Хорошилов А.И. Программирование на языке СИМУЛА-67. Наука, М., 1985, 288 с.
2. Kay A. The Reactive Engine. Ph.D.Thesis, University of Utah September, 1969 .
3. Hewitt C. Viewing Control Structures as Patterns of Passing Messages. Artificial Intelligence, 8(1977), pp. 323-364 .
4. Goldberg A. Smalltalk-80: The Interactive Programming Environment. Reading, MA: Addison - Wesley, 1984 .
5. Schmucker K.J. Object-oriented Languages for the Macintosh. Byte, 11(1986)8, pp. 177-185.
6. Stoyan H., Görs G. Was ist objektorientierte Programmierung. id : Stoyan H., Wedekind H. (Hrsg.): Objektorientierte Software - und Hardwarestrukturen. Stuttgart: B.G. Teubner, 1983, s. 10-29 .
7. Эли Дж., Кумдо М. Экспертные системы: концепции и примеры. Финансы и статистика, М., 1987, с. 16-20.
8. Поспелов Д.А. Ситуационное управление: теория и практика. Наука, М., 1986, 284 с.
9. Nilsson N.J. Principles of Artificial Intelligence. Springer - Verlag, Berlin-Heidelberg-New York, 1982, pp. 131-159 .
10. Schreiber P. Grundlagen der Mathematik, 2. Auflage. Berlin: VEB Deutscher Verlag der Wissenschaften, 1984, s. 49-56 .
11. Математическая энциклопедия. (под ред. И.М. Виноградова), М., Сов. энциклопедия, 1977.
12. Замулин А.В., Скопин И.Н. Конструкции языка программирования как типы данных. - Прикладная информатика, 1986, вып. 2, с. 93-110.
13. Месарович М., Мако Д., Такахара И. Теория иерархических многоуровневых систем. Мир, М., 1973, с. 56-62.
14. Herrig D. Design theory for CAD systems and CAD objects. id : Encarnacao J., Krause F.-L. (Ed.): File structures and data bases for CAD. Amsterdam-New York-Oxford: North-Holland 1982.
15. Бюгкляуссен Х. Метод многоуровневого потока событий (МПС) для описания процессов в сетях и иерархиях операторов. ОИЯИ, РИИ-86-542, Дубна, 1986.
16. Cox B.J. Message / Object Programming: An Evolutionary Change in Programming Technology IEEE Software , 1(1984)1 , pp. 50-61 .
17. Бабаев И.О., Новиков Ф.А., Петрушина Т.И. Язык Декарт-входной язык системы СПОРА. - Прикладная информатика. 1981, вып. I, с. 35-73.
18. Железнова К.Н., Корнейчук А.А., Шарапова Э.В. Инструментальный программный комплекс, поддерживающий фортраноориентированную технологию программирования на ЭВМ центрального вычислительного комплекса ОИЯИ. ОИЯИ, РИИ-85-905, Дубна, 1985.

Рукопись поступила в издательский отдел  
28 июля 1987 года.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

Вы можете получить по почте перечисленные ниже книги, если они не были заказаны ранее.

Д3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	5 р.00 к.
Д7-83-644	Труды Международной школы-семинара по физике тяжелых ионов. Алушта, 1983.	6 р.55 к.
Д2,13-83-689	Труды рабочего совещания по проблемам излучения и детектирования гравитационных волн. Дубна, 1983.	2 р.00 к.
Д13-84-63	Труды XI Международного симпозиума по ядерной электронике. Братислава, Чехословакия, 1983.	4 р.50 к.
Д2-84-366	Труды 7 Международного совещания по проблемам квантовой теории поля. Алушта, 1984.	4 р.30 к.
Д1,2-84-599	Труды VII Международного семинара по проблемам физики высоких энергий. Дубна, 1984.	5 р.50 к.
Д10,11-84-818	Труды V Международного совещания по проблемам математического моделирования, программированию и математическим методам решения физических задач. Дубна, 1983.	3 р.50 к.
Д17-84-850	Труды III Международного симпозиума по избранным проблемам статистической механики. Дубна, 1984. /2 тома/	7 р.75 к.
Д11-85-791	Труды Международного совещания по аналитическим вычислениям на ЭВМ и их применению в теоретической физике. Дубна, 1985.	4 р.00 к.
Д13-85-793	Труды XII Международного симпозиума по ядерной электронике. Дубна, 1985.	4 р.80 к.
Д4-85-851	Труды Международной школы по структуре ядра. Алушта, 1985.	3 р.75 к.
Д3,4,17-86-747	Труды V Международной школы по нейтронной физике. Алушта, 1986.	4 р.50 к.
	Труды IX Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1984. /2 тома/	13 р.50 к.
Д1,2-86-668	Труды VIII Международного семинара по проблемам физики высоких энергий. Дубна, 1986. /2 тома/	7 р.35 к.

Заказы на упомянутые книги могут быть направлены по адресу:  
101000 Москва, Главпочтамт, п/я 79. Издательский отдел Объединенного института ядерных исследований.

Корнейчук А.А., Шуберт В., Юнгклауссен Х. P5-87-590  
Обобщенные средства объектно-ориентированного программирования

Излагается формальный подход для систематизации понятий, возникающих в связи с объектно-ориентированным программированием. Вводятся три уровня моделирования на основе трех отношений: часть-целое /нижний уровень/, элемент-класс /средний уровень/, подкласс-суперкласс /верхний уровень/. На каждом уровне средством моделирования являются специфические объекты: на нижнем уровне - сети и иерархии операторов, на среднем уровне - классы и акторы, на верхнем уровне - иерархии классов и акторов. Объекты определяются как элементарные теории. Объекты низшего уровня являются интерпретациями объектов высших уровней.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1987

Перевод О.С.Виноградовой

Kornejchuk A.A., Schubert W., Jungklaussen H. P5-87-590  
Generalized Means of Object-Oriented Programming

A formal approach to systematizing the object-oriented programming concepts is stated. Three model levels based on three relations: part-whole (the lowest level), element-class (the middle level), and subclass-superclass (the highest level) are introduced. The specific objects are model means on each level: operator nets and hierarchies on the lowest level, classes and actors on the middle level, and class and actors hierarchies on highest level. The objects are determined as elementary theories. The low level objects are interpretations of the high level objects.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Preprint of the Joint Institute for Nuclear Research. Dubna 1987