



2. Палферова С.Ш., Иванов О.И., Бабенко Н.Г., Кузнецова О.А. Математическая модель ценового согласования при распределении вычислительных ресурсов./Вестник Казанского технологического университета. 2008. № 4. С. 182-187.

3. Ярыгин О.Н., Беляев М.А. Уточнение вида функции предпочтения альтернатив в методе анализа иерархий./ Карельский научный журнал. 2013. № 4. С. 49-52.

И.И. Бикмуллина

## ИНТЕЛЛЕКТУАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА СИНТЕЗА ДИАГРАММ КЛАССОВ UML НА ОСНОВЕ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ

(Казанский национальный исследовательский технический университет  
имени А.Н. Туполева)

В исследовании предлагается искусственно интеллектуальный [1] подход к разработке UML[2] модели на примере построения диаграмм классов. Основное внимание уделено структурному синтезу информационных систем.

Основная цель исследования - повышение качества проектной проработки информационных систем за счет автоматизации синтеза диаграммы классов UML. Конечная цель автоматизации синтеза диаграммы классов заключается в передаче функций разработки автоматизированных систем специалисту-предметнику, потому что только он может вложить в автоматизированную систему все богатство используемых методических приемов.

Предмет исследования: автоматизация структурного синтеза объектов с неиерархическими связями на основе семантических моделей предметной области.

Задачи исследования:

1. Анализ существующих методов синтеза структурной модели информационных систем.
2. Разработка методики семантического моделирования структурных отношений предметной области.
3. Разработка методики синтеза структурной модели информационной системы с неиерархическими связями.
4. Разработка архитектуры автоматизированной системы структурного синтеза и алгоритмов решения задач.
5. Экспериментальное исследование разработанных методик и алгоритмов.

Разработка самой интеллектуальной информационной системы синтеза диаграммы классов UML декомпозируется на два уровня: абстрактный и конкретный. Это связано с тем, что рассматривая такие категории как понятие предметной области (концептуальная модель) и программная модель постоянно сталкиваемся с двумя уровнями рассмотрения: абстрактным и конкретным.

Абстрактный уровень характерен для описания понятий предметной об-



ласти и представляется структурными категориями:

- системами понятий,
- именами основных понятий предметной области,
- рекомендациями(ограничениями) по выбору значений атрибутов,
- именами и типами операций, с помощью которых можно производить вычисления,
- семантическими категориями: смысл того или иного понятия, операции,
- правила определения присутствия того или иного свойства,
- правила (система аксиом) образования классов,
- правила (система аксиом) образования отношений между классами.

В абстрактном типе предметной области собраны вместе формальные структурные и общие семантические признаки понятия предметной области:

а) сигнатура понятия предметной области представляет собой пару  $\Sigma=(S, \Omega)$ , где  $S$  - множество обозначений для всех базовых множеств данных, а  $\Omega$  - множество идентификаторов для всех базовых операций над понятиями;

б) совокупность  $E$  утверждений (требований, условий, ограничений и т.п.), которые имеют место для элементов понятия предметной области.

Конкретный уровень в исследовании представляет собой структурную модель программы, заданную виде диаграммы классов UML. Где в описании структурной модели программы определены состав, атрибуты со своими значениями, свойства, понятия предметной области, выполнены все необходимые семантические проверки, определены и расставлены виды отношений между классами, определены ассоциативные отношения.

Так на рисунке 1 абстрактным уровнем будут данные подаваемые на вход синтеза моделей UML, а в конкретный уровень попадут сами синтезированные модели UML.

Таким образом, в исследовании представляют интерес две канонические задачи построения процедур взаимодействия абстрактного и конкретного уровней.

Первая задача состоит в том, что строится структурная модель программы (конкретный уровень), которая предъявляется вычислительной процедуре с целью проверки, соответствует ли она свойствам сформулированной в системе понятий модели предметной области (абстрактный уровень). Нетрудно убедиться, что это задача анализа соответствия объекта описанию предметной области.

Содержанием второй задачи является: с помощью вычислительной процедуры и сформулированных в системе понятий структурной модели предметной области (абстрактный уровень) правил построить структурную модель программы (конкретный уровень). Это задача синтеза структурной модели программы. При этом если мы хотим формально определить структурную модель программы с помощью добротной системы понятий и обозначений, то определение ее структуры нужно подчинить определению семантики: структура модели программы должна быть верна смыслу (в нашем случае смысл выражается с



помощью описания свойств).

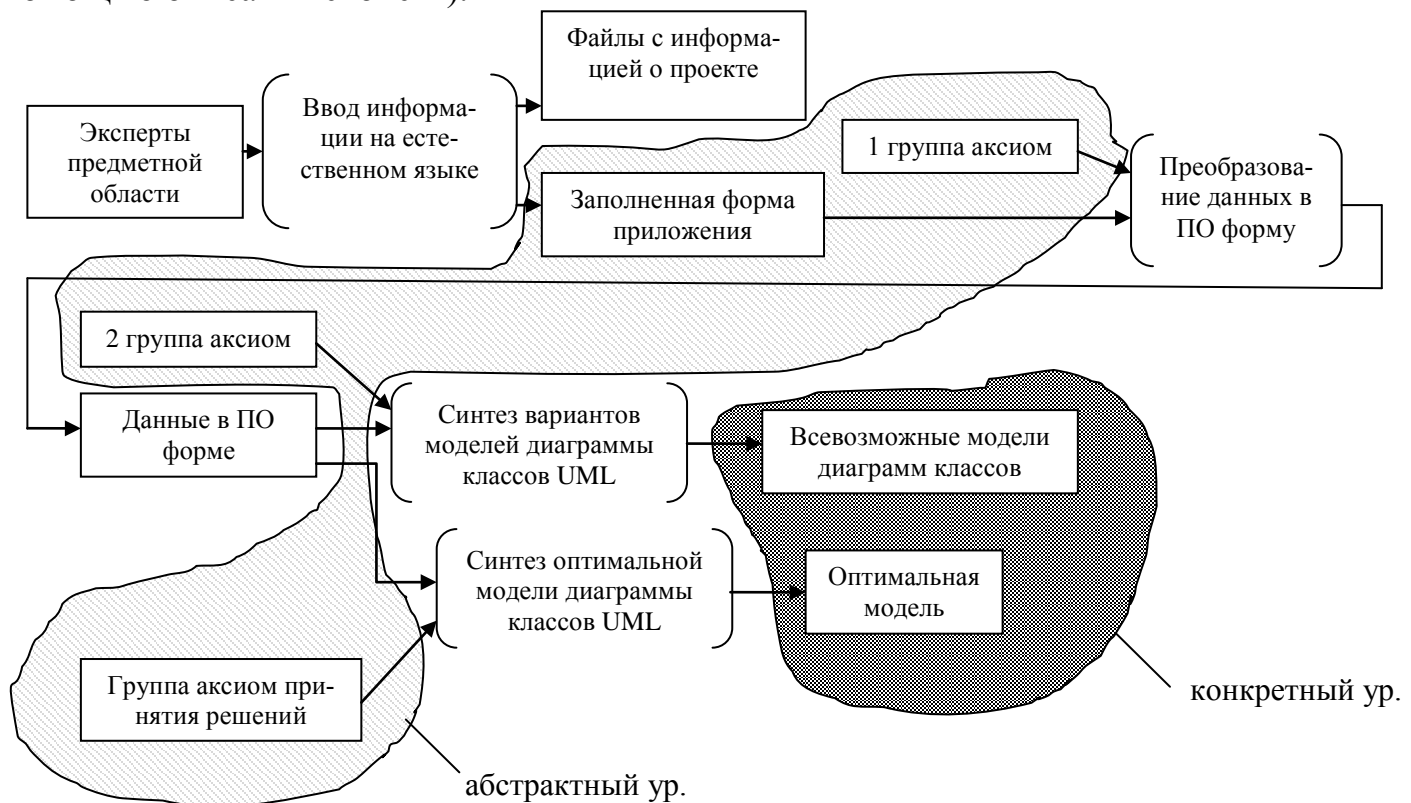


Рис. 1. Процесс синтеза моделей UML

Для построения интеллектуальных систем мы должны зафиксировать содержание понятия в виде описания его свойств, а выделение объема понятия будет осуществляться процедурно путем вычисления свойств рассматриваемых объектов. Для примера спецификации предметной области и задачи проектирования рассмотрим класс прикладных задач «Статистические характеристики», в котором используются понятия: «массив экспериментальных данных», «среднее значение», «среднее отклонение». Необходимо получить модель диаграммы классов и решить прикладную задачу:

```

Понятие прикладная_задача
{
  Атрибуты
    С : среднее
    Д : дисперсия
  Функции Вывести_результат(С, Д)
  Значение -
  Свойство Вывести_результат >0
}
    
```

При этом входные данные вводимые экспертом формируются в семантическое описание задачи, а для этого на данном этапе закладываются в семантический язык статистические характеристики:

(1) Спецификация свойств и особенностей класса, в нашем случае, прогноз банков состоит из разделов. Так спецификация «массив экспериментальных данных» содержит раздел: «исходные данные». А раздел «основные стати-



стические характеристики» содержит такие спецификации, как «дисперсия» и «значение среднего».

(2) Спецификация профессионального раздела представляет собой систему описаний понятий исследования. Определение отдельного понятия имеет структуру: Понятие <название> <тело понятия> Конец <название>.

(3) Описание тела понятия является основой для автоматизированной обработки и может содержать следующие разделы: «Параметры», «Состав», «Атрибуты», «Внешние», «Свойства».

(4) Раздел описания «Атрибуты» определяет атрибутивные свойства понятия и представляет собой бинарное отношение «денотат характеризуется ...» между определяемым понятием и элементами раздела. Описание и использование раздела «Атрибуты» строится по тем же правилам, что и описание раздела «Состав».

В результате семантическое описание задачи выглядит следующим образом:

...*Понятие* среднее\_значение

*Атрибуты*

Данные : {массив\_экспериментальных\_дан-ных}.

Значение : {вещественное}.

*Функции* Получить\_значение(Данные : {вектор из вещественное}): вещественное.

*Значение* Сумма\_вектора(Данные) / Размер из Данные.

*Конец* среднее\_значение...

Таким образом, если мы хотим формально определить структурную модель программы с помощью добротной системы понятий и обозначений, то определение ее структуры нужно подчинить определению семантики: структура модели программы должна быть верна смыслу (в нашем случае смысл выражается с помощью описания свойств). Предлагается рассматривать структурную семантическую модель предметной области как логическую модель описания наследственно конечной иерархической структурной надстройки над моделью последовательности простых данных, характеризующей последовательность базовых данных вычислителя.

### Заключение

Итак, определены абстрактный и конкретный уровни процесса синтеза диаграмм классов UML.

Формируется семантическое описание, как задачи, так и постановки того, что необходимо найти.

На основе семантических моделей разрабатывается задача синтеза структурной модели информационной технологии: синтез диаграммы классов языка UML. Разрабатывается формальная система порождения структурных решений, которая позволяет описать пошаговый процесс построения структурных решений по предварительно заданной системе понятий.



## Литература

1. Романов, В.П. Интеллектуальные информационные системы в экономике: учеб. пособие / В.П. Романов; под ред. д.э.н., проф. Н.П. Тихомирова. — М.: Экзамен, 2003. — 496 с.
2. Рамбо, Дж. UML/ Дж. Рамбо, А. Якобсон, Г. Буч: специальный справочник— СПб.: Питер, 2002. - 656 с.

М.В. Бураков

## КОНСТРУИРОВАНИЕ НЕЧЕТКОГО РЕГУЛЯТОРА ПИД-ТИПА

(Санкт-Петербургский государственный  
университет аэрокосмического приборостроения)

Понятие «интеллектуальные регуляторы» обычно ассоциируется с нечеткими или нейронными регуляторами. Нечеткие логические регуляторы (НЛР) используют для реализации закона управления правила, связывающие нечеткое описание ситуации и сигнал управления ([1, 2] и другие). НЛР могут использоваться для организации контура адаптации в системе, управляя параметрами регулятора нижнего уровня [3, 4], но обычно они включаются последовательно с объектом управления, являясь нелинейным корректирующим звеном. Основная проблема использования НЛР заключается в отсутствии простых методик их настройки. В данной работе предлагается универсальный алгоритм для решения этой проблемы.

В промышленности нечеткие регуляторы пока не могут конкурировать с традиционными ПИД-регуляторами, причиной популярности которых является простота построения и использования, пригодность для решения большинства практических задач и низкая стоимость [5, 6].

Однако существующие методы расчета параметров ПИД-регуляторов ориентированы на линейные системы, поскольку сам регулятор является здесь линейным динамическим звеном. Стремление к улучшению качества работы ПИД-регуляторов привело к появлению НЛР «ПИД-типа» ([7, 8] и другие). Это название подчеркивает, что регулятор получает такие же входные сигналы, что и обычный ПИД-регулятор, но реализует более сложный нелинейный алгоритм управления.

Для обоснования алгоритма конструирования НЛР ПИД-типа рассмотрим закон управления, реализуемый линейным ПИД-регулятором:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}, \quad (1)$$

где  $e(t)$  – ошибка управления,  $k_p$ ,  $k_i$ ,  $k_d$  – коэффициенты, выбираемые в процессе проектирования.