

УДК 681.518.3:681.3.06

<https://doi.org/10.36906/AP-2020/06>**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИСПОЛЬЗОВАНИЮ ОБЪЕКТНОЙ
МОДЕЛИ ПРИ ИЗУЧЕНИИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ****Дубенецкий В. А.***канд. техн. наук**Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ»**г. Санкт-Петербург, Россия***Кузнецов А. Г.***канд. техн. наук**Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ»**г. Санкт-Петербург, Россия***Цехановский В. В.***канд. техн. наук**Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ»**г. Санкт-Петербург, Россия*

Аннотация. Рассматриваются подходы к обучению технологии проектирования информационных систем на основе использования объектно-ориентированных моделей. Предлагается методика проведения правдоподобных рассуждений и элементы доказательства для проверки полноты и корректности построенных моделей. Данные рекомендации будут полезны при разработке практических занятий для подготовки бакалавров и магистров по направлению 09.03.02 «Информационные системы и технологии».

Ключевые слова: объектная модель, психологическая модель, метамодель, модель потоков.

Специалист в области информационных технологий постоянно сталкивается с необходимостью строить анализировать разнообразные модели, которые позволяют решать задачи проектирования, реализации и сопровождения информационных систем (ИС) различного назначения. Мышление на основе использования моделей является необходимым условием профессиональной деятельности таких специалистов. При организации обучения ИТ-специалистов важно понять и обеспечить связь между логическим представлением и его психологическим отражением в сознании обучающихся. Как отмечено в работе С. И. Шапиро «От алгоритмов — к суждениям» [1], «центральным при переходе от логической модели к психологической является вопрос о том, как, в соответствии с условием задачи, система сознается, то как нерасчлененный элемент, то как подсистема». С. И. Шапиро предложил математическую модель обучения, основанную на понятии импликации, он ввел соответствие между логической и психологической моделями. Посылкой А и заключением В логической модели соответственно связаны с А' — (мысль об А) и В' — (мысль о В). Операции (\rightarrow) — импликация ставится в соответствие (\vdash) — ассоциация. Показано, что

такое соответствие подтверждается рядом исследований. Установленное соответствие, как показано в [1], не изоморфно — оно однозначно определено только в направлении от психического к логико-математическому. Обучение связано с формированием у обучаемого целостной системы понятий. «Понятием» наименования является правило, решающее для произвольного объекта вопрос о его принадлежности множеству. Множества объектов образуют содержание «понятия» [1].

Классическая схема проектирования может быть представлена в виде преобразования моделей следующего вида:

$D_o \rightarrow D_a \rightarrow D_d \rightarrow D_r$, где

D_o — домен предметной области в границах проекта информационной системы (ИС);

D_a — домен абстрактного представления предметной в виде комплекса информационных моделей, полученных на основе анализа D_o .

D_d — домен проектных решений, представленный в виде информационных моделей этапа проектирования ИС;

D_r — домен реализации проектных решений, описанных в D_d .

Разрыв между доменом проблемы и доменом реализации остается достаточно большим. Разнообразие процессов, объектов и событий даже в рамках одного домена предметной области, для поддержки которой строится информационная система, велико. Кроме того, сам домен подвержен постоянным изменениям. В процессе развития выявляются новые процессы, ролевые структуры, изменяется логика поведения уже выявленных и новых объектов.

На рисунке 1 представлена диаграмма потоков, иллюстрирующая варианты получения решения по реализации ИС.

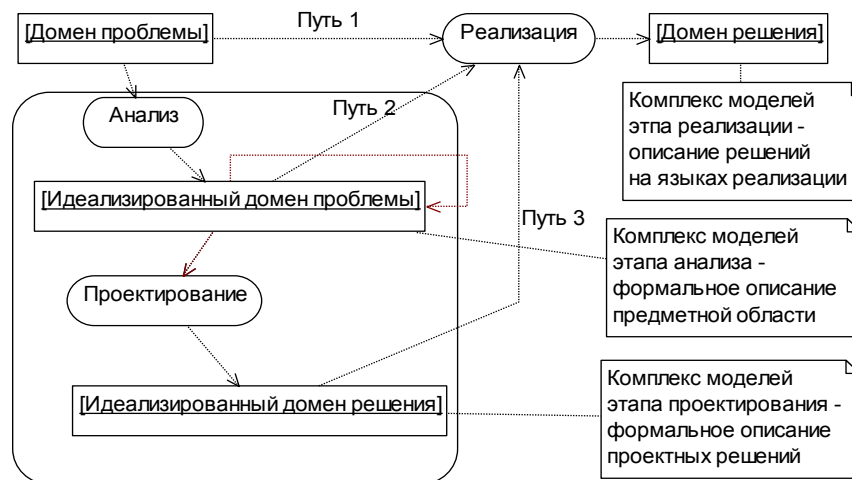


Рис. 1. Варианты получения решения по реализации ИС

Предпочтительным является путь 3, который предполагает выделение самостоятельных этапов *Проектирование* и формирование *Идеализированного домена решения*. В этом варианте процесс *Реализация* строится не на основе моделей этапа анализа, а на основе комплекса моделей этапа проектирования. Комплекс моделей этапа проектирования должен, по возможности, быть инвариантным к потоку изменений в модели этапа анализа. Модели *Идеализированного домена решения* представляют собой основу для реализации инструментов описания и реализации потоков изменения в модели анализа на периоде исполнения.

Существуют различные методологии моделирования предметной области, среди которых следует выделить функционально-ориентированные, информационно-ориентированные и объектно-ориентированные методологии (ООМ). Последняя достаточно универсальна и позволяет единообразно описывать необходимые и важные свойства любой другой информационной модели. Кроме того, ООМ обладает высоким уровнем формализации, что при определенных условиях позволяет использовать эти модели для исполнения. Перспективным направлением в развитии технологии создания ИС является управление разработкой приложений на основе моделей, допускающих исполнение. В работе «Фабрики разработки программ» Д. Гринфилд [2] формальная модель определяется как артефакт, фиксирующий метаданные в форме, которая может быть интерпретирована людьми и обработана инструментами.

При обучении технологии проектирования ИС важное место занимают объектно-ориентированные модели. ООМ обладают рядом несомненных преимуществ, таких как высокий уровень формализации описания объектов, процессов и событий, инкапсуляция, наследование, иерархическое представление. Эти свойства модели несомненно могут представлять расширенную основу для формирования понятий при обучении.

В качестве базового языка описания для доменов D_a , D_d , D_r предлагается использовать унифицированный язык моделирования UML, который удовлетворяет всем требованиям объектно-ориентированного моделирования и обладает выразительными средствами графического представления моделей. Для повышения моделирующих возможностей потребуются определенное расширение метамодели UML. Одним из направлений расширения является формулирование правил оперирования метаклассами и их взаимосвязями [3]. Ясно, что оперирование классами элементов как специальными элементами может обеспечить более гибкое и универсальное описание предметной области, проектных решений и разнообразных ограничений.

Домен D_o содержит, как правило, наборы конкретных объектов в виде документов, инструкций, методик, схем, алгоритмов, правил. Остальные домены в основном содержат информационные модели в виде метамodelей, не содержащих элементов. Метамodelи из D_a должны обеспечить, как минимум, создание, хранение, представление и необходимую обработку любых элементов из D_o . В свою очередь, компоненты метамodelи из D_a должны быть поддержаны в домене проектирования D_d . Построение модели классов для D_a включает в себя следующие шаги. Представим мета-мета модель для комплекса статических объектных моделей следующим образом:

$Q_{parentQ}$ — классификатор мета-мета модели;

$P_{supportB}$ — поддержки бизнес-процессов прецедентами;

R_{typeC} — типизация ролей;

$C_{provideO}$, C_{useesR} , R_{typeC} — зависимости между классами и представление их в виде ролей и операций;

$C_{supportU}$, $R_{supportU}$ — локальные модели классов, поддерживающие прецеденты;

где Q — множество компонентов мета-мета модели; P — множество процессов; B — множество бизнес-процессов в домене D_a ; U — множество прецедентов; C — множество классов; R — множество ролей; O — множество операций; C_t — множество инструментальных классов проектной модели; C_a — множество классов модели этапа анализа; $Q = P \cup C \cup R \cup O$; $P = B \cup U$; $C = C_t \cup C_a$.

По аналогии с соответствием логического и психического, представленного в [1], будем считать, что приведенной выше статической объектной модели соответствует

некоторая психологическая модель в виде системы понятий, которая должна сформироваться в процессе обучения.

С каждым процессом p_i может быть связана подмодель, в которой представлены множество классов C_i и множество ролей R_i , необходимых для исполнения прецедента p_i . Ставится вопрос: «Подмодель обладает необходимыми свойствами для поддержания заданного прецедента?».

В качестве исходного компонента — точки для формирования локальных моделей различного назначения, могут быть использованы компоненты следующих видов: один из бизнес-процессов B ; один из вариантов использования U ; один из классов C ; одна из операций класса O ; одно из состояний класса (один из процессов ЖЦ класса) S ; один из процессов выполнения операции W .

Проверка корректности моделей классов. Модель классов описывает базовые ограничения и правила работы с объектами и их взаимосвязями. Проверка корректности модели предполагает проведение ряда правдоподобных рассуждений, позволяющих показать, что известные факты в виде некоторых примеров не противоречат проверяемой модели. При этом проверяется возможность модели занести и запомнить известные факты и найти одни факты на основе других фактов. Кроме того, такая проверка проводится и для расширенного множества фактов, которые ранее не были учтены при разработке модели [3]. Модели этапа анализа строятся непосредственно на основе некоторой совокупности фактов из предметной области.

Основные виды проверок можно разбить на следующие группы: группа, проверяющая возможность фиксации фактов в рамках модели; группа, проверяющая возможность найти факты в рамках модели; группа, проверяющая возможность получения одних фактов на основе других.

Формируя некоторый класс и включая его в модель необходимо убедиться, что все его свойства могут быть описаны с помощью модели. Часть свойств описывается в виде атрибутов классов. В этой части необходимо убедиться, что новые привлеченные факты не требуют расширения атрибутивной модели. Противоречие разрешается путем включения новых атрибутов в спецификацию класса. При этом необходимо проверить наличие соответствующих классов для описания новых ассоциаций.

Типовая последовательность проверки модели этапа проектирования включает в себя следующие этапы.

1. Отбор представительных фактов (например, документов) для проверки.

Чем шире границы и выше уровень абстрагирования проверяемой модели, тем тщательнее должны подбираться исходные факты. Всегда остается вероятность того, что новые факты не смогут быть поддержаны проверяемой моделью. Каждой группе фактов должна соответствовать своя оригинальная модель классов этапа анализа. Все компоненты этих моделей должны быть определены глобально в рамках всего проекта. Дублирование и использование одного компонента в различных контекстах должны быть исключены.

2. Последовательно в соответствии с функциональными зависимостями компонентов.

Проверяется возможность объявить с помощью модели объекты, их атрибуты и ассоциации. Например, если необходимо объявить новый класс, то в модели необходим соответствующий классификатор. Если необходимо объявить новый элемент, соответствующий класс или есть в модели, или его можно объявить в классификаторе. Проверка возможности объявления объектов начинается с классификаторов, затем с классов-сущностей, далее проверяем возможности описания значений атрибутов и списков по ролям.

3. Выявление и описание ошибок и неполноты модели.

4. Если ошибок нет, то переход к пункту 6, иначе переход к пункту 5.
5. Коррекция исследуемой модели и переход к пункту 2.
6. Описание результатов проверки, эффектов использования модели и требований к правилам ее применения.

Модели потоков. В языке UML для описания поведения используются модели поведения. Одной из форм представления поведения являются диаграммы последовательности. Помимо потока управления на диаграммах деятельности можно показывать потоки объектов. Модели потоков будем описывать в виде модели деятельности, из которой исключены связи по управлению.

Поток объекта представляется на диаграмме в виде прямоугольника, в который помещено имя потока в виде «*имя класса*» [*имя состояния элементов класса*]. Одной из форм представления модели потоков является диаграмма потоков. Модель потоков строится на основе выбранного фрагмента модели классов. Множество действий в модели потоков соответствует множеству всех ролей R (в виде функций доступа) и вызовов операций O классов исходной модели классов. На модели потоков проще и нагляднее проверять полноту и корректность построенной модели классов. Кроме того, модель потоков может явиться хорошей основой для автоматизации обработки разнообразных запросов. Рассмотрим правила построения модели потоков на примерах.

Пример проверки модели классов этапа проектирования. На рисунке 2 представлена модель классов для ведения справочника изделий.

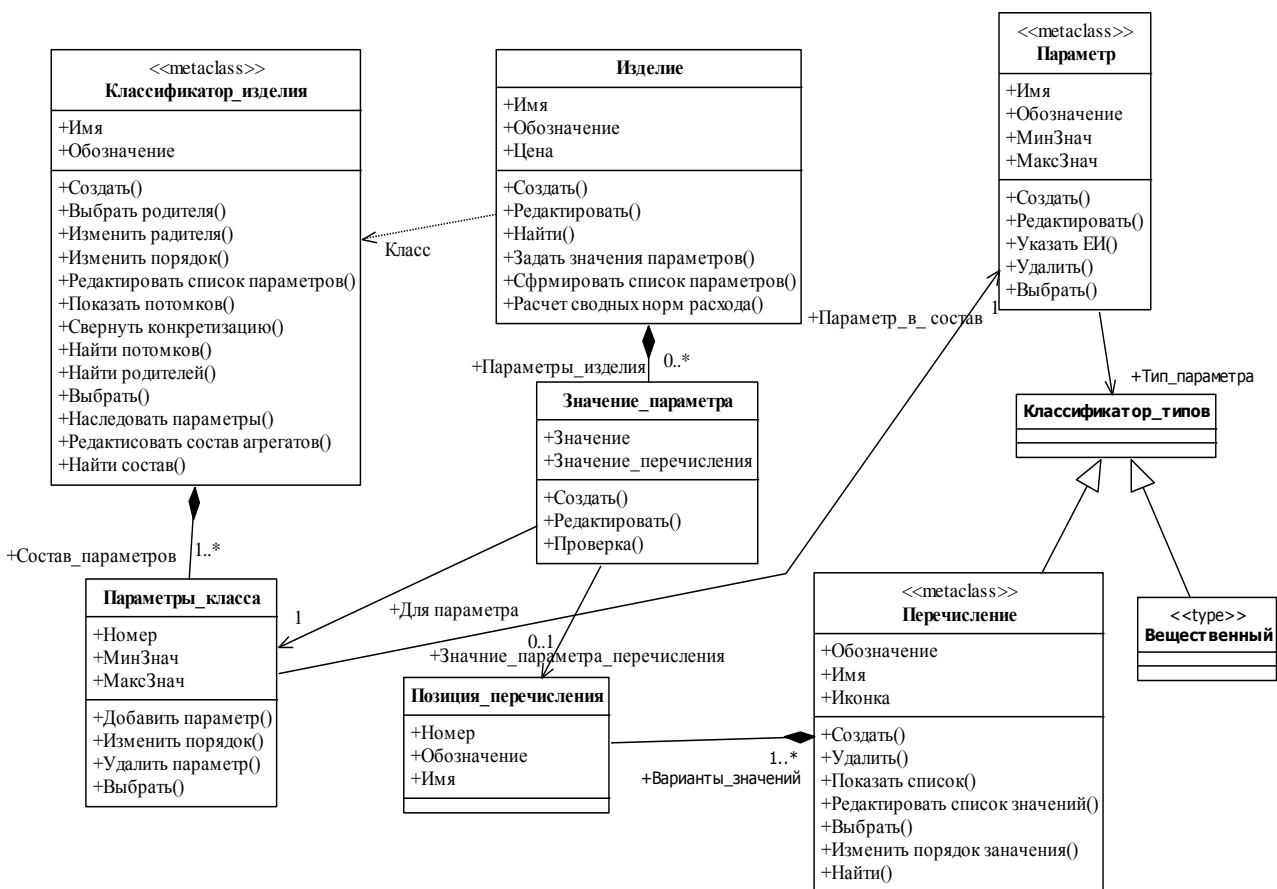


Рис. 2. Пример модели проектной классов для построения модели потоков

На рисунке 3 представлен фрагмент классификатора изделий, который будет использоваться для проверки проектной модели.

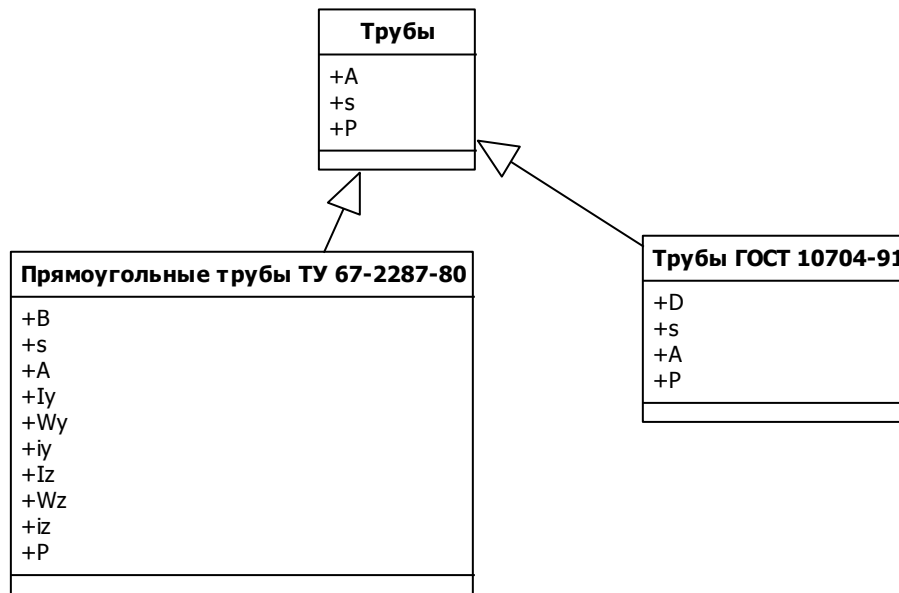


Рис. 3. Пример модели классов этапа анализа для построения модели потоков

На рисунке 4 представлена диаграмма потоков, соответствующая модели рис. 2. Выполним первый пункт проверки. Для описания фрагмента справочника необходимо средствами модели объявить классы *Трубы*, *Трубы ГОСТ 10704-91*, *Прямоугольные трубы ТУ 67-2287-80*. Метакласс *Классификатор_изделий* позволяет это сделать. Следующий этап — описание параметров с указанием их типов. Метаклассы *Параметр* и *Классификатор_параметров* позволяет описать необходимые параметры, представленные в модели на рисунке 3. В данном примере все параметры вещественного типа. Далее формируем списки параметров для классов примера. Класс *Параметр_класса* и ассоциации с ролями *Состав_параметров* и *Параметр_в_состав* позволяют сформировать эти списки и указать границы изменения значений параметров. Теперь можно включать объекты классов *Трубы ГОСТ 10704-91*, *Прямоугольные трубы ТУ 67-2287-80*. По известному классу объектов находится состав параметров. Значения параметров заносятся с помощью класса *Значение_параметра* и ассоциации с ролью *Параметры_изделия*. Выбранный пример для проверки не содержит всех возможных вариантов в части описания параметров различных типов. Поэтому целесообразно продолжить проверки с новыми исходными фактами.

Показано, что UML обладает необходимыми свойствами для решения задач анализа и проектирования ИС. Реализация объектной модели в среде реляционных СУБД может быть автоматизирована.

На основе предложенных расширений UML описаны модели метаклассов компонентов UML (метамодель прецедентов, метамодель классов, метамодель ролей и другие). Использование этих метамodelей в составе моделей этапа проектирования позволяет существенно повысить уровень абстрагирования и степень повторного использования проектных решений ИС.

Расширен состав диаграмм UML. Описаны правила построения и анализа модели потоков, являющей разновидностью модели последовательности.

Предложена методика проверки корректности и полноты моделей классов как этапа анализа, так и этапа проектирования. Показана их взаимосвязь.

Приведен пример использования метамodelей при разработке модели классов этапа проектирования.

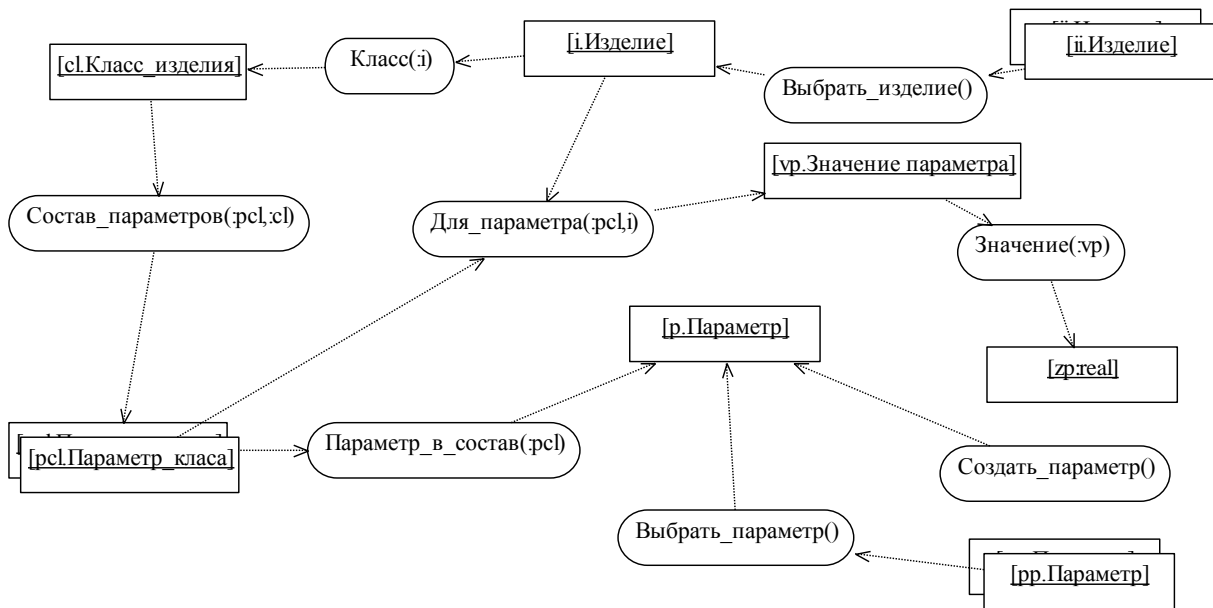


Рис. 4. Модель потоков данных для примера

Литература

1. Шапиро С. И. От алгоритмов – к суждениям. М.: Советское радио, 1973. 298 с.
2. Гринфилд Д., Шорт К. Фабрики разработки программ: потоковая сборка типовых приложений, моделирование, структуры и инструменты. М.: Диалектика, 2007. 591 с.
3. Дубенецкий В. А., Кузнецов А. Г., Цехановский В. В. Технология создания корпоративных информационно-управляющих систем на основе моделей, допускающих исполнение. СПб: Изд-во СПбГЭУ «ЛЭТИ», 2019, 158 с.

©Дубенецкий В. А., Кузнецов А. Г., Цехановский В. В., 2020