

Глава 2. Инструменты и документы. UML

Университет марксизма-ленинизма (УМЛ) — форма высшего звена системы просвещения Коммунистической партии Советского Союза. (Из Википедии)

2.1. UML – как же он моделирует?

Достаточно бесспорное, а поэтому и непродуктивное определение UML можно дать так:

UML – унифицированный язык моделирования для описания программной системы на всех этапах жизненного цикла с использованием парадигмы объектного моделирования

То, что UML используется для описания программных систем и может быть применен на всех этапах жизненного цикла, следует из его назначения. Аналогично, объектно-ориентированный подход может быть как формальным общим местом, так и краеугольным камнем, но это в большей степени зависит не от UML как инструмента, а от практики его применения. Остается само название. Разберем «по буквам».

Унифицированный. UML возник не на пустом месте: все его компоненты имеют исторические прототипы. Достоинство UML в том, что он объединил в единое целое. Другая интерпретация термина: UML не привязан к программным системам, он может использоваться для моделирования любых сложных систем. Применительно к программной инженерии это могут быть предметная область, бизнес-процессы. Парадоксальным фактом является и то, что UML используется разработчиками и как мета-язык для описания расширений UML, а также унифицированного процесса разработки ПО.

Язык. UML – формализованный графический язык. Все его элементы строго специфицированы, что дает основания назвать его **формальным графическим языком**. Но формальный язык можно рассматривать в двух аспектах:

- язык описаний – спецификаций модели;
- язык исполнения (интерпретации) модели, т.е. собственно моделирования.

Со вторым дело обстоит значительно хуже. Хотя отдельные компоненты языка позволяют генерировать непосредственно программный код (например, диаграммы классов), собственно принципы интерпретации компонент модели не оговорены, да и не могут быть оговорены, поскольку модель не рассматривается изначально как модель программного кода системы, а может описывать любые произвольные сущности. Кроме того, никто не гарантирует отсутствие «дыр» в комплексном описании модели в виде нескольких компонент (диаграмм), т.е. ее семантической целостности.

Моделирование. Здесь более всего вопросов, поэтому рассмотрим претензии UML на моделирование отдельно.

Что понимать под моделированием?

«Он лег на спину и заложил обе руки под голову. Илья Ильич занялся разработкою плана имения. Он быстро пробежал в уме несколько серьезных, коренных статей об оброке, о запашке, придумал новую меру, построже, против лени и бродяжничества крестьян и перешел к устройству собственного житья-бытья в деревне».

И.А.Гончаров. «Обломов»

Прежде всего, надо разобраться с самим термином «моделирование». Согласно определению, модель – это **система, воспроизводящая поведение оригинала**. Соответственно, моделирование – анализ или «исполнение» модели с целью изучения свойств или поведения оригинала. Модели и способы моделирования бывают разными (Рис.2.1):

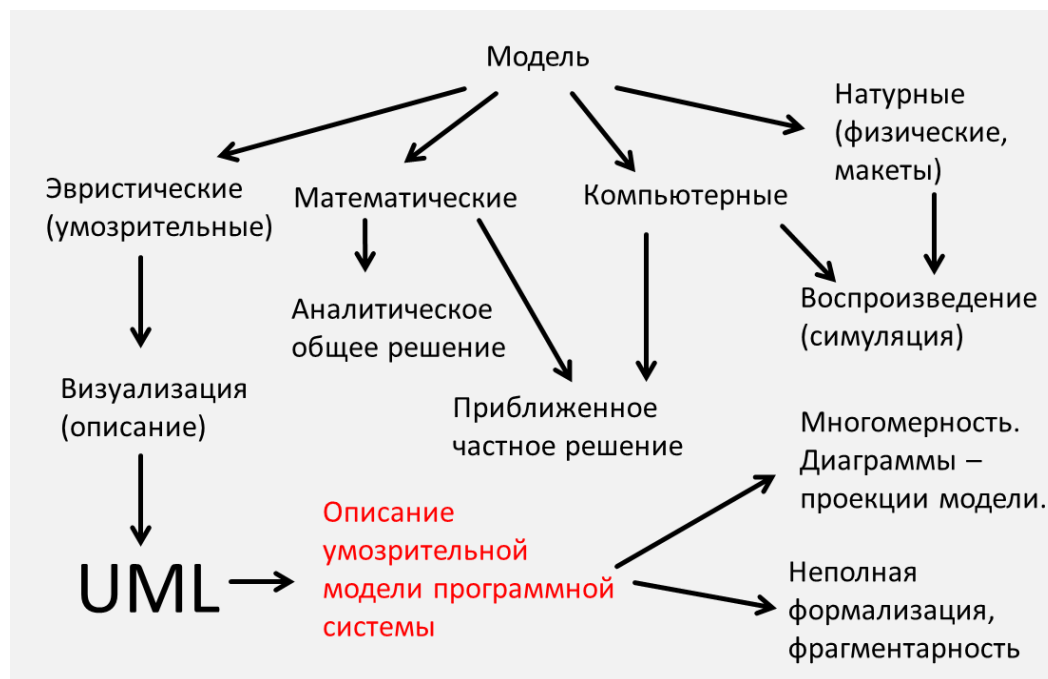


Рис.2-1. Виды моделей

- **натурная модель** представляет собой физическую копию (в натуральном размере, или уменьшенную) для изучения ее поведения в той же среде, что и оригинал, способ моделирования – прямое **воспроизведение**;
- **математическая модель** – описание поведения оригинала средствами того или иного математического аппарата (например, система обычных или дифференциальных уравнений, система массового обслуживания). Способы моделирования могут быть разными:
 - **аналитическое решение** в виде функциональной зависимости параметров модели от времени, координат и т.п.. Обычно аналитическое решение может быть получено для достаточно простых моделей, построенных в результате упрощения исходной модели или игнорирования в ней несущественных деталей;
 - для сложные математических моделей можно получить **приближенное частное решение** для некоторого набора значений параметров при помощи **методов вычислительной математики**, которые обычно реализуются программными средствами. Решение в таком виде получается «по точкам»,

т.е. в виде набора частных решений, представленных в виде таблицы или графика;

- **компьютерная модель** – программа, которая вычисляет приближенное частное решение математической модели, либо осуществляет **симуляцию** – представление компонент оригинала в виде программных объектов и воспроизведение в них последовательности событий или состояний оригинала;
- **эмпирическая** – умозрительная модель, представленная в виде описания или схемы, моделирование заключается в логическом анализе (верификации), умозрительном или компьютерном исполнении сценариев.

Как это ни странно, но модель в UML относится к самому «слабому» в смысле воспроизведения оригинала типу моделей – эмпирическому. В модели в UML нельзя воспроизвести оригинал в традиционном для моделирования понимании.

Модель в UML – умозрительное представление программной системы в «воспаленном воображении разработчиков». Документы UML – проекции (срезы) этого представления, отражающие отдельные аспекты организации или поведения системы.

Фанатизм от UML и чувство меры

Как и в случае с документированием программного проекта, в отношении UML имеются две крайности: от отрицания необходимости использования вообще, до полной фиксации в документе каждого «чиха» в процессе проектирования. Несколько замечаний на этот счет:

глубина документирования и использования UML в проекте может меняться в самых широких пределах в зависимости от принятой методологии;

- имеются полезные неканонические альтернативы: схемы (например, диаграммы экранов GUI), наброски, неформальные документы, нестандартные артефакты;
- в дополнение к основному назначению модели можно использовать как средство коммуникации, визуализации, восстановления контекста разработки или справочник;

Есть еще пара замечаний по использованию UML как модели программной системы. В 1.1 уже был сформулирован тезис о том, что только программный код является **чертежом проекта**, ибо способен гарантированно воспроизвести проектируемую программную систему. Все остальные документы и модели не обладают свойством чертежа, а являются **эскизами**.

Вторая претензия - общего плана. Любой программный код имеет **комбинаторное множество вариантов исполнения** (в терминах UML является классификатором, порождающим экземпляры). Некоторые UML-диаграммы, описывающие **поведение** системы, изображают один из возможных вариантов (**экземпляров**) поведения, следовательно, не обладают необходимой всеобщностью.