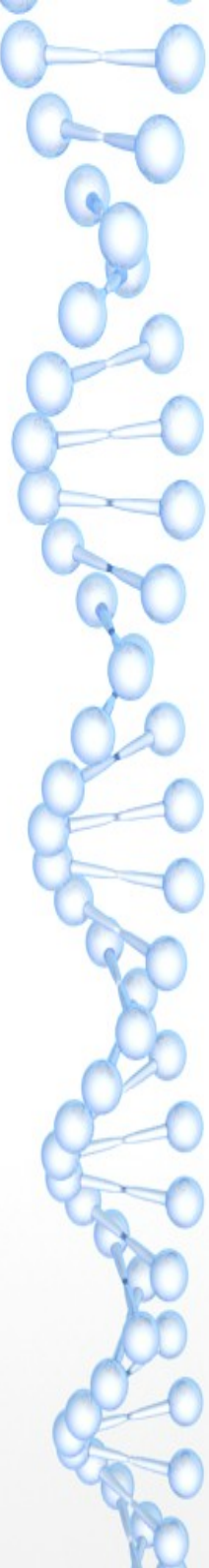


На каком уровне иерархии спецификаций начинается СПО?

XII Конференция
«Свободное программное обеспечение в высшей школе»
город Переславль-Залесский, 27-29 января 2017 года
29.01.2017
Захаров И.В.



Актуальность уточнения места СПО в пространстве спецификаций

Тезис 1

Стандарты, определяющие документы «Текст программы» и «Описание программы»:
ГОСТ 19.401-78 Единая система программной документации. Текст программы. Требования к содержанию и оформлению.
ГОСТ 19.402-78 Единая система программной документации. Описание программы.
ГОСТ 19.101-77 Единая система программной документации. Виды программ и программных документов.
ГОСТы 24 и 34 серий в этой части ссылаются на ГОСТы 19 серии.

Идеология СПО выражает себя через отношение к спецификациям уровня «Текст программы» и «Описание программы» (праве доступа, модификации и распространении кода), но подразумевает нечто большее.

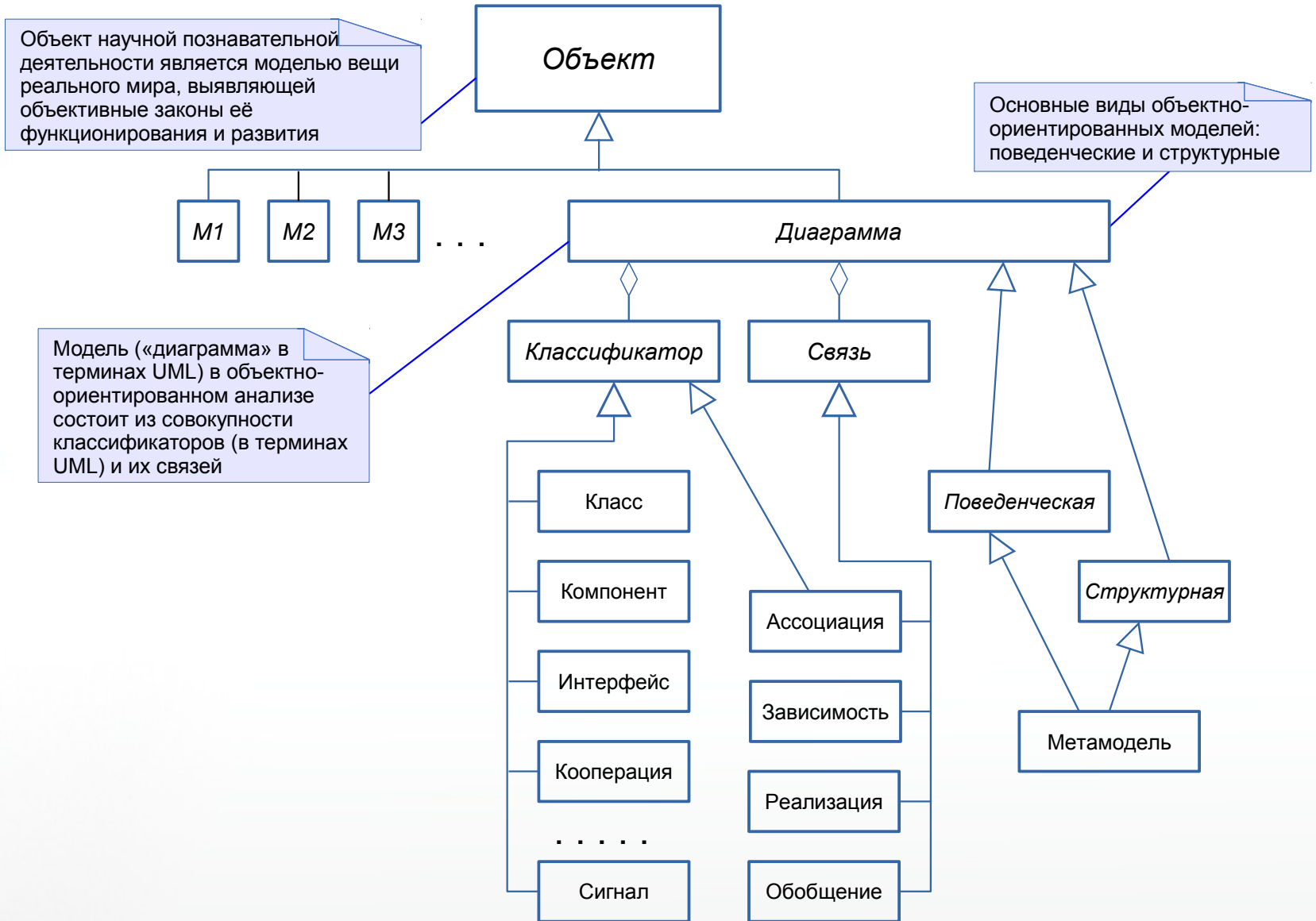
Тезис 2

Информационные технологии быстро меняются (в том числе развиваются) в рамках мощных трендов.
Например, предметом новых стандартов является не «Текст программы», а архитектура, процессы, «точка зрения» и т. п.:
ISO/IEC/IEEE 42010 System and software engineering — Architecture description;
ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств.

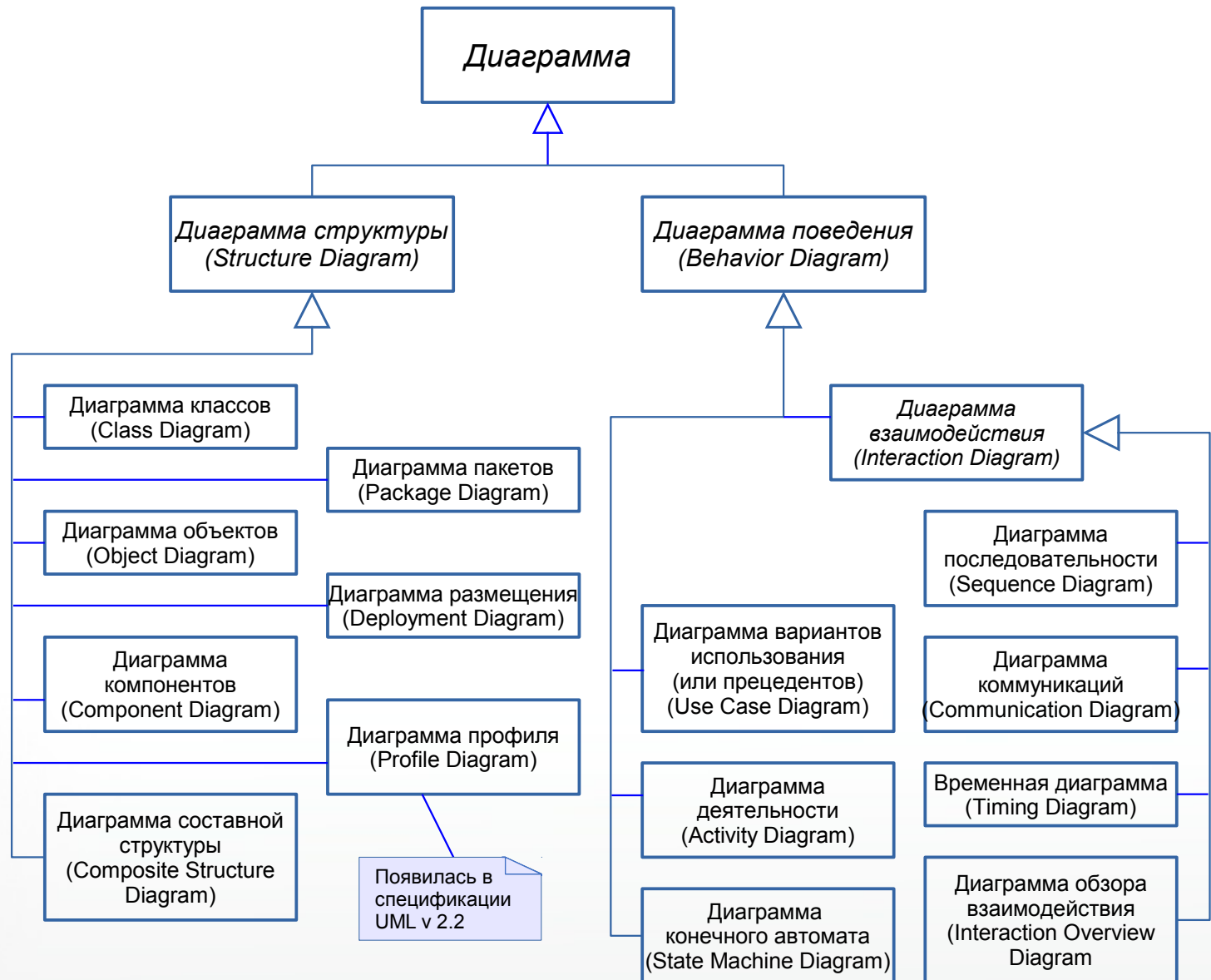
Тезис 3

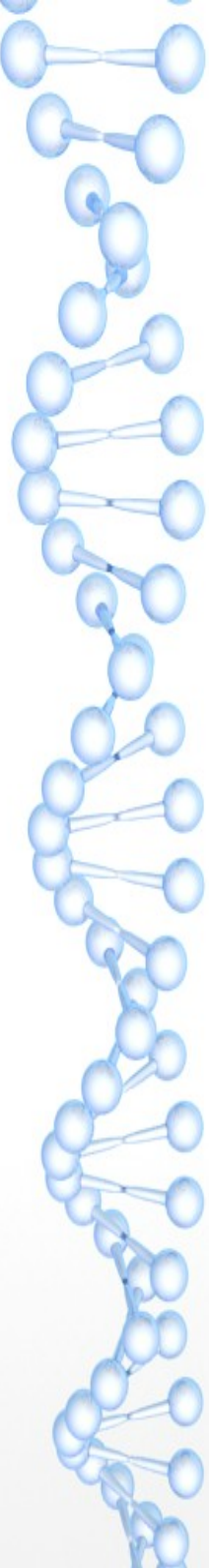
Требуется уточнение идеологии СПО по отношению к спецификациям других уровней. Для определения отношения различных документов (спецификаций, стандартов) с идеологией СПО необходима разработка или выбор определенных иерархий спецификаций.

Объект научной познавательной деятельности как родительский класс для спецификаций объектно-ориентированного анализа



Детализация класса *Диаграмма UML* (на основе стр. 683 спецификации v 2.5)





Пример: краткая детализация (теоретическая схема) диаграмм взаимодействия UML

Диаграмма взаимодействия – модель поведения кооперации, реализующая интегративное свойство системы. Другие варианты контекста: взаимодействие между объектами в реализации операций и ВСКД семантики класса.

Способы моделирования взаимодействий:

в контексте времени (диаграмма последовательности);

в контексте структурных связей (диаграмма коммуникации).

Взаимодействие (interaction) – поведение, в которое вовлечено множество сообщений, передаваемых между объектами, которые играют определенные роли в некотором контексте для достижения заданной цели.

Сообщение – это спецификация взаимодействия объектов, которая передает информацию и ожидает последующих действий.

Сообщение чаще всего вызывает операции или передачу сигнала, а также может сопровождаться созданием или уничтожением объектов.

Типы сообщений:

call (вызвать) – вызов операции объекта (наиболее частое событие);

return (возвращает) – вызывает сообщение вызывающему объекту;

send (послать) – посылает сигнал объекту;

create (создать) – создать объект;

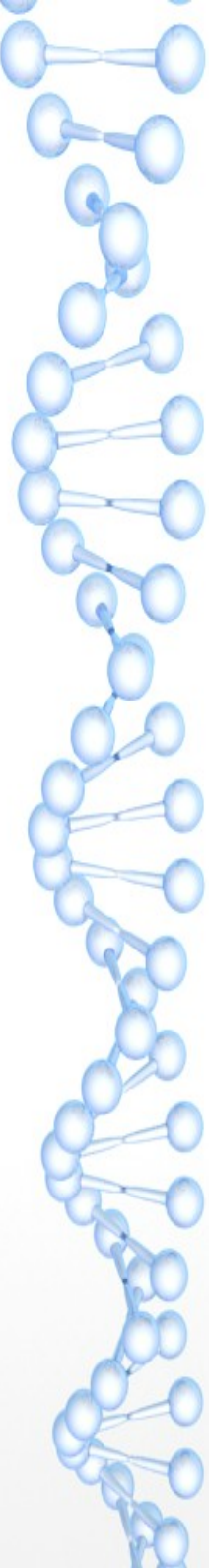
destroy (уничтожить) – уничтожает объект, в т.ч. сам себя;

не специфицируемые в UML сообщения (в части синтаксиса и семантики).

Синхронное сообщение – сообщение, ожидающее ответа (стрелка с закрашенным треугольником), ответ на синхронное сообщение (возврат из вызова) — пунктирная стрелка уголком (может не отображаться).

Асинхронное сообщение – сообщение, не ожидающее ответа (стрелка с уголком).

Роль – прототип объекта.



Пример: краткая детализация (теоретическая схема) диаграммы конечного автомата UML

Диаграмма состояний (state diagram) – модель конечного автомата, показывающая поток управления от одного состояния к другому (граф с вершинами и ребрами). Может использоваться для моделирования практически любого элемента системы, однако чаще всего применяется для моделирования всей системы, подсистемы, класса, сценариев варианта использования, поведения интерфейсов.

Конечный автомат (state machine) – описание последовательности состояний, через которые проходит объект на протяжении жизненного цикла, реагируя на события, а также описание реакции на эти события. Подходит для спецификации поведения объекта, который должен отвечать на асинхронные сообщения либо его поведение зависит от прошлого.

Явно специфицируются: события, на которые объект может реагировать, реакция на эти события, влияние более раннего поведения на текущее.

Состояние (state) – ситуация в жизненном цикле объекта, на протяжении которой он удовлетворяет некоторому условию, выполняет некоторую деятельность или ожидает некоторого события.

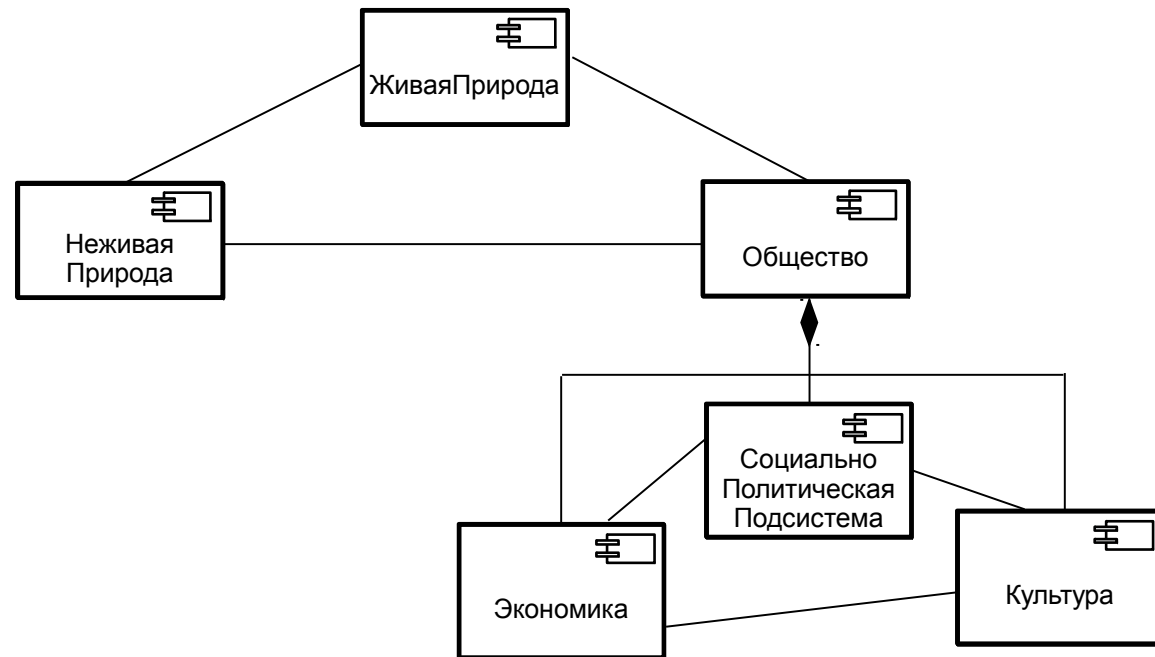
Событие (event) – спецификация существенного факта, которое происходит во времени и пространстве. В контексте автомата событие – это воздействие, которое вызывает переход между состояниями.

Переход (transition) – связь между двумя состояниями, показывающая, что объект, находящийся в первом состоянии, должен выполнить некоторые действия и перейти во второе, как только произойдет определенное событие и будут выполнены определенные условия.

Деятельность (activity) – процесс, работа, происходящая внутри автомата.

Действие (action) – примитивное выполняемое вычисление, приводящее к смене состояния модели или возврату значения.

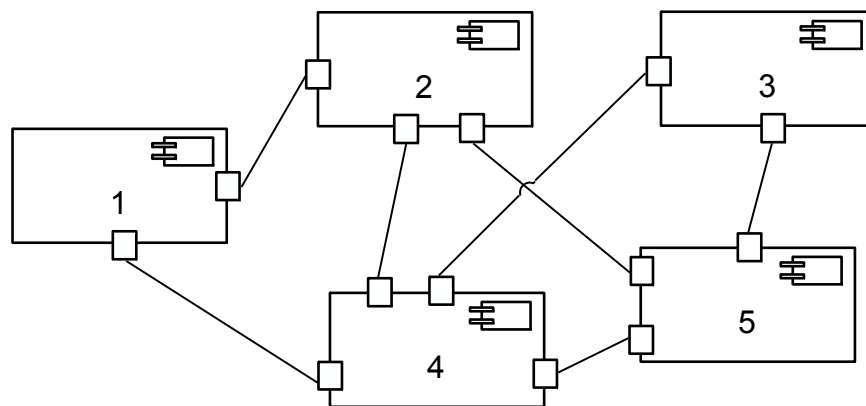
Метамоделли, начиная с наиболее общего уровня: структурная модель «Современная научная картина мира»



Особое значение этой модели в том, что перечень её компонентов фактически является общепринятым.

Если для каких-то задач целесообразно изобразить эту схему в виде диаграммы компонентов (концептуально более мягкая форма - пакеты), то в смысле объектно-ориентированного проектирования надо согласиться, что эта схема изображает структуру родительского класса для любых других систем, которые должны наследовать свойства и операции этого класса.

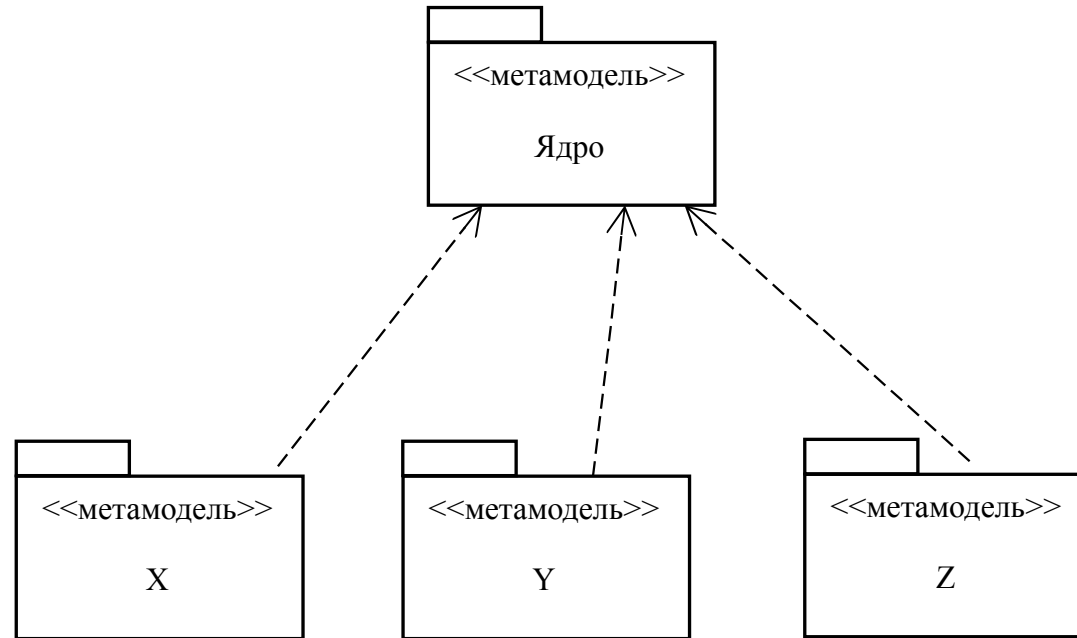
Метамоделли, начиная с наиболее общего уровня: структурная модель «Внутренняя сложность»



Перспективная метамодель, позволяющая составить формализованное представление о сложной системе.

Эта метамодель отражает подход к моделированию сложной системы, когда в качестве первой спецификации удалось выявить отдельные компоненты системы и определить их взаимосвязи. Стохастические процессы и другие неформализуемые характеристики остаются внутри компонентов, которые как «черные ящики» могут в дальнейшем быть более подробно специфицированы через состояние портов.

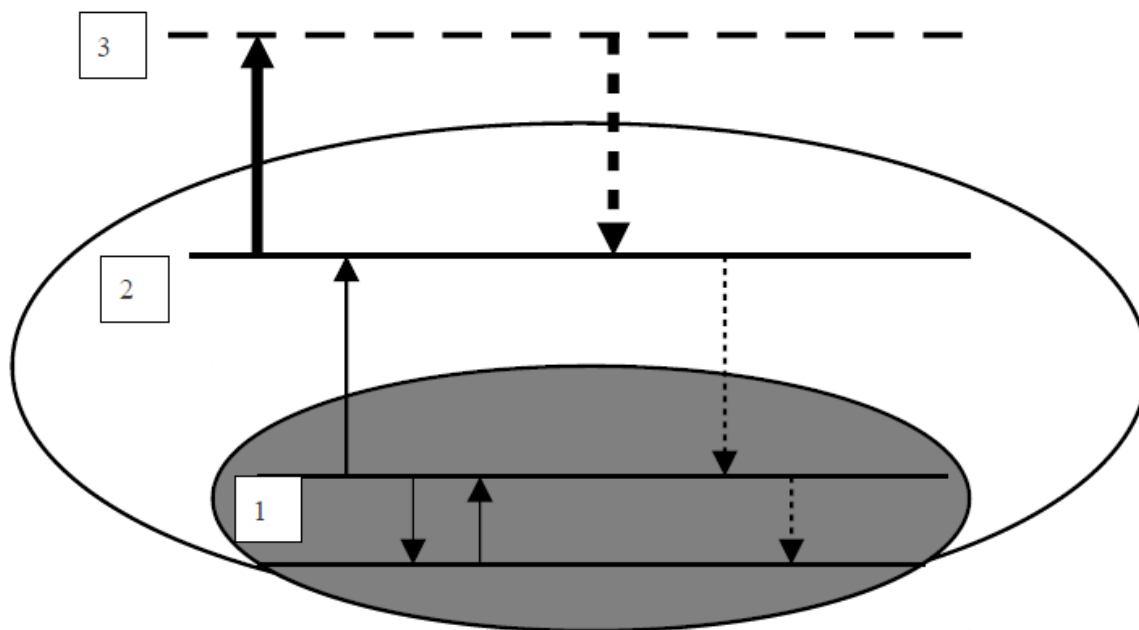
Метамоделли, начиная с наиболее общего уровня: структурная модель «Ядро - периферия»



Часто используемая мета модель, отражающая первичное представление о сложной системе, как о логически разделяемом представлении: ядро и его окружение (периферия). Мета модель изображена в виде совокупности «пакетов» между которыми существуют связи типа «зависимость». «Пакеты» являются логическим представлением проектировщика о системе. На физическом уровне в системе такое разделение может отсутствовать.

Метамоделю, начиная с наиболее общего уровня: поведенческая модель «Новый уровень саморегуляции»

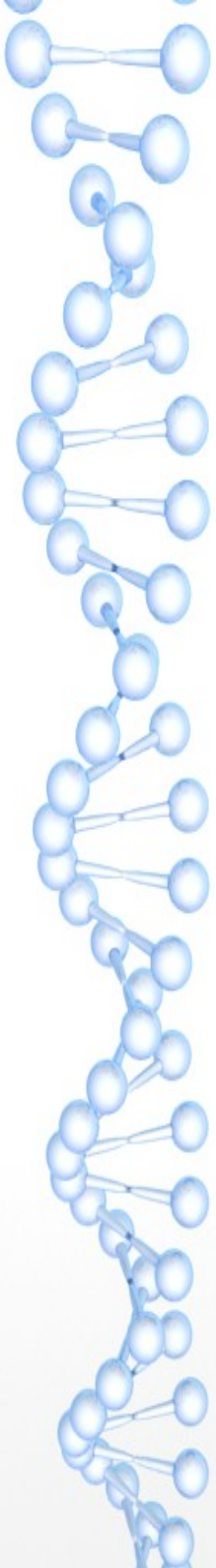
Поведенческая метамоделю (в авторской нотации), которая отражает сценарий поведения систем уровня системной сложности с самоорганизацией (саморазвитием), предложенная В.С. Стёпиным.



1. Исходная саморегуляция.
2. Новый тип саморегуляции, основанный на трансформации предшествующих уровней иерархии системы.
3. Потенциально возможный уровень организации при продолжении развития системы как возможность нового типа саморегуляции.

Выводы

1. В терминах рассмотренной иерархии спецификаций исторически сложившаяся, «естественная», конкуренция свободного и проприетарного ПО находится в области спецификаций конкретных Классов, Объектов (в смысле экземпляров классов) и Артефактов. Расширение этой области конкуренции может привести к негативным для развития ИТ последствиям.
2. Целесообразно вовлечение в сферу свободных лицензий максимального объема спецификаций, начиная с максимально обобщенных уровней. Предположительно, наиболее важным на начальном этапе является публикация метамodelей различного рода и уровня иерархии, включая профили, шаблоны и т. д.
3. Требуется дальнейшая проработка вопросов построения иерархии(-ий) спецификаций, в рамках которых возможно определение отношения различных документов (спецификаций, стандартов) с идеологией СПО.
4. Если строить иерархию спецификаций от уровня Объекта научной познавательной деятельности, то это может создать юридические основания для защиты открытости нижележащих спецификаций, так как они наследуют его свойства и операции.





Спасибо за внимание!

Захаров И.В.
z@7xt.ru