

Министерство образования и науки Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

В.Л. Баденко, В.К. Ядыкин

Методология модельно-ориентированного
системного инжиниринга

Учебное пособие

Санкт-Петербург
2025

УДК 004.9

Баденко В.Л. **Методология модельно-ориентированного системного инжиниринга.** – СПб., 2025. – 88 с.

Модельно-ориентированный системный инжиниринг (Model Based System Engineering – MBSE) – современный этап развития системной инженерии. Цель реализации методологии MBSE состоит в том, чтобы создать единую, унифицированную модель целевой системы («единый источник истины»), из которой могли бы быть сформированы все документы и представления об объекте. В современном мире MBSE становится эффективным инструментом по реализации сложных строительных объектов. В настоящем пособии изложены основные методологические подходы, которые используются при реализации сложных инженерных проектов.

Пособие предназначено для студентов, обучающихся в Инженерно-строительном институте и институте Энергетики СПбПУ. Пособие также будет полезно всем обучающимся, интересующимся современными инженерными методами цифровой трансформации предприятий.

Ил. 24. Библиогр.: 84 назв.

Оглавление

Введение.....	3
Предпосылки внедрения методологии модельно-ориентированного системного инжиниринга	7
Семантика домена, метамодели и онтологии	14
Процесс разработки онтологии для методологии MBSE	27
Применение методологии MBSE	34
Построение системной архитектуры в MBSE.....	47
Архитектура систем на основе моделей и принятие решений.....	57
Заключение	77
Литература	80

Введение

Актуальность настоящего пособия определяется тем, что наблюдается экспоненциальный рост разработок в области инструментов инжиниринга [1, 2], а также нового поколения соответствующего программного обеспечения [3]. При этом существующие продукты еще далеки от удовлетворения всех потребностей инженеров-проектировщиков [4]. Растущее понимание в необходимости расширения внедрения новых цифровых технологий, в частности цифровых двойников, активно формирует новые методы и инструменты цифрового системного инжиниринга, которые позволяют обеспечить должный уровень адекватности, чтобы говорить о формировании цифрового двойника [5, 6]. Практическое применение методов системной инженерии при разработке адекватных цифровых представлений, существующих сложных технических объектов, показал, что при этом используются разные модели – вербальные, экспертные, математические, цифровые, интеллектуальные, которые результативно дополняют друг друга, и поэтому надо формировать инструменты, чтобы системно и быстро совмещать это множество моделей [7]. Как ответ на эти вызовы возникла и стала развиваться идея модельно-ориентированного системного инжиниринга (Model Based System Engineering, MBSE) [8], позволяющего совмещать и использовать в инжиниринге «все» типы моделей. Согласно MBSE подходу опорную структуру моделей системного инжиниринга можно упорядочить по принципу последовательности разработки [8, 9]: онтологические модели → архитектурные модели → параметризированные архитектурные модели → математические модели → компьютерные модели. В этой цепочке каждый последующий тип моделей гармонизирован с предыдущими, наследует и дополняет их новыми дополнительными свойствами и характеристиками. Таким образом, проекты разработки цифровых представлений сложных технических систем на базе MBSE стартуют с формирования онтологии предметной области, которые представляют собой справочник основных

определений понятий, которые используются для создания архитектурных системных моделей целевых объектов. Такие модели представляют собой текстовые описания объектов в унифицированной форме справочников, иерархически структурированных сущностей (требований, функций, компонентов и процессов) и матриц связанностей этих сущностей [1, 2, 8, 9]. В формировании нового индустриального уклада Индустрия 4.0 [1] ключевую роль сыграла системная инженерия, представляющая собой междисциплинарный комплексный подход к созданию и эксплуатации сложных технических объектов, который охватывает все этапы жизненного цикла проекта и позволяет формализовать все сложные взаимодействия и взаимовлияния объектов и субъектов для удовлетворения потребностей заказчика [2]. В настоящее время наибольшие успехи системный инжиниринг демонстрирует в аэрокосмической отрасли, чему посвящены соответствующие учебные программы, развиваемые в Московском Физико-техническом институте – национальном исследовательском университете (МФТИ) [3-5]. Будущим инженерам-политехникам полезно ориентироваться в системной инженерии при реализации современных проектов сложных технических систем.

Системный инжиниринг – это междисциплинарная область инженерии, посвященная проектированию и управлению сложными инженерными проектами в соответствии с их жизненным циклом. На современном этапе развития данного направления мы говорим о развитии системной инженерии на новом уровне и говорим о модельно-ориентированной системной инженерии. Этот термин является переводом с английского Model Based System Engineering. Существует соответствующая международная организация – Международный совет по системной инженерии (International Council on Systems Engineering – INCOSE), которая является некоммерческой организацией, ставящая своей целью развитие системной инженерии, разработку стандартов для неё и профессиональный рост системных инженеров. Согласно INCOSE, MBSE представляет собой формализованное

применение моделирования для поддержки системных требований, проектирования, анализа, верификации и валидации, начиная с этапа концептуального проектирования и продолжая на протяжении всей разработки и более поздних фаз жизненного цикла [6, 7]. MBSE – это технический подход к системной инженерии, который фокусируется на создании и использовании моделей предметной области в качестве основного средства обмена информацией. Иными словами, используется модельно-ориентированный подход в противоположность более традиционному документно-ориентированному подходу.

Анализ практики создания цифрового представления существующих промышленных предприятий позволяет сделать следующие выводы [1-5]:

- Сегодня нет бизнеса, не затронутого цифровой трансформацией, и нет вида работы, который не был бы изменен, пусть и незначительно, цифровыми инструментами
- Технических навыков и глубокого интереса к новым технологиям недостаточно для процветания в цифровой экономике
- Цифровые технологии постоянно меняются, меняются и способы их применения. Это означает, что организации и люди в них находятся в состоянии постоянных изменений
- В век данных, алгоритмов и искусственного интеллекта, невозможным добиться успеха без цифрового мышления, позволяющего по-новому взглянуть на мир и изменить свое поведение
- Цифровое мышление — это набор подходов, которые мы используем для осмысления и использования данных и технологий

В этой связи актуальной является задача по созданию такого цифрового представления сложного технического объекта, которое базируется на рассмотрении такого объекта как сложной технической системы. В настоящее время развитие современных инструментов модельно-ориентированного системного инжиниринга позволяет решить перечисленные проблемы [6-12].

В настоящем пособии собранные материалы, опубликованные в работах INCOSE, сотрудников МФТИ и ГК Росатом, которые не всегда могут быть доступны студентам.

Данное пособие предназначено для студентов, обучающихся по направлению «Строительство» и «Энергетика». Оно также будет интересно для студентов других специальностей.

Предпосылки внедрения методологии модельно-ориентированного системного инжиниринга

Мы живем в эпоху гиперсвязности и экспоненциальных технологических тенденций. Первая продолжает увеличивать сложность систем, в то время как вторая продолжает быстро производить новые технологические возможности. Модельно-ориентированный системный инжиниринг (MBSE) и цифровая инженерия (DE) являются двумя значимыми примерами таких достижений [13-15].

Эти достижения иногда приводили к путанице и неправильному пониманию MBSE и DE даже в инженерном сообществе. Например, поскольку практически вся инженерия использует модели, то резонно возникают вопросы: что отличает системную инженерию на основе моделей от традиционной инженерии с использованием моделей? Чем цифровая инженерия отличается от MBSE с точки зрения акцента, области применения и методологий? Как модели, созданные с помощью MBSE, интегрируются со сторонними моделями? Как концепции цифровой инженерии, такие как цифровой двойник (digital twin – DT) и цифровой поток (digital thread), могут дополнять и улучшать MBSE? Как можно использовать технологию цифровых двойников для облегчения проверки, валидации и тестирования систем? Такие вопросы, среди прочих, послужили к развитию методологии MBSE [16-18].

Международный совет по системной инженерии (INCOSSE) определяет MBSE как «формализованное применение моделирования для поддержки требований к системе, проектирования, анализа, верификации и валидации, начинающееся на этапе концептуального проектирования и продолжающееся на протяжении всей разработки и последующих этапов жизненного цикла» [13-14]. Рассмотрим некоторые вопросы, которые связаны с развитием методологии MBSE.

Какие новые разработки в области системной инженерии важны для решения задач XXI века?

Сегодня системную инженерию можно рассматривать как объединение двух направлений. Первое из них охватывает традиционные методы системной инженерии, которые часто характеризуются как «старая школа» и которые хорошо работают для относительно зрелых и не слишком сложных систем. Второе включает «новые методы», которые можно характеризовать как системную инженерию для новой эпохи и которые мотивированы сложностью и широтой современных социотехнических систем, а также продолжающимися достижениями в области вычислений, инженерии и материалов. Для последнего требуется новый тип руководства, которое будет реализовывать такие разработки на основе передовых цифровых технологий. Например, системные требования, как и в прошлом, должны быть строго определены, но теперь требования должны быть сформулированы таким образом, чтобы решение включало бы использование современных методов и алгоритмов для доказательства правильности. Аналогичным образом, искусственный интеллект и глубокое обучение могут быть использованы для улучшения и ускорения детального проектирования.

В чем уникальность MBSE?

В то время как системные инженеры регулярно работали с моделями в течение нескольких десятилетий, MBSE предлагает целостное предложение, основанное на способе создания, представления и использования моделей. В частности, MBSE основан на нескольких ключевых руководящих принципах:

- MBSE включает унифицированную модель, которая является междисциплинарной, охватывает жизненный цикл системы и используется в качестве авторитетного источника истины для всех видов деятельности по системной инженерии;
- Документы, возникают в моделях или генерируются на основе моделей, так что управление и контроль конфигурации не является отдельной деятельностью, которая страдает от хорошо известных проблем обслуживания и выхода из строя;

- MBSE является нейтральным по отношению к методологии моделирования, можно использовать различные методы и конструкции моделирования на основе доступности информации и сложности отраслей/областей знаний (доменов). MBSE занимается созданием и эксплуатацией моделей домена, а не документов, как основного средства обмена информацией.

Использование формального представления в MBSE может поддерживать рассуждения и способствовать взаимодействию между разнородными моделями. Растущий акцент на использовании онтологий для захвата знаний предметной области, отраженных в сценариях и вариантах использования, помогает ограничить режим применимости (т.е. область применения), в рамках которого могут быть поставлены вопросы и на них можно ответить, а также выполнять рассуждения. Модели, созданные с помощью MBSE, связывают дисциплины, облегчают сотрудничество между заинтересованными сторонами и поощряют переосмысление предположений и ограничений. Важно, что модели в MBSE можно проверять в имитационных средах (с помощью виртуального прототипирования и разработки цифровых двойников) в рамках проверочного тестирования, тем самым снижая стоимость и усилия физического тестирования.

Какие типы моделей можно создать с помощью MBSE?

Модели, созданные с использованием MBSE, могут варьироваться от абстрактных вербальных моделей до подробных представлений. Они могут быть детерминированными или стохастическими, с низкой или высокой точностью, приближительными или точными. Они могут отражать одну или несколько перспектив разработки системного проекта. Особенно сложной проблемой при построении моделей является определение того, что следует включить в описание и представление (в первую очередь цифровое) системы, а что можно разумно опустить без ущерба для обоснованности модели и полезности для предполагаемых целей. При этом следует помнить, что исключение ключевых факторов может подрвать точность и целостность

моделей, а включение нерелевантных факторов может внести излишнюю сложность.

Как можно применить MBSE к системам на базе искусственного интеллекта?

Сегодня все большее число систем используют искусственный интеллект (ИИ) для когнитивных задач, т. е. задач, требующих некоторой комбинации рассуждений, оценки, прогнозирования и обучения. Примерами таких задач являются автономное управление и навигация в беспилотных автомобилях, планирование и принятие решений в современных истребителях и беспилотных летательных аппаратах (БПЛА) и их командах, а также адаптивные энергетические сети. Моделирование, проверка и тестирование систем, управляемых ИИ, остаются постоянными проблемами, поскольку такие системы неизменно демонстрируют сложное, недетерминированное поведение. Типичные проблемы MBSE для таких систем включают моделирование системы с частичным знанием начальных условий, пространства состояний системы и операционной среды. Для таких систем вероятностное моделирование и связанные с ним методы проверки и тестирования продолжают демонстрировать значительные перспективы.

Как MBSE справляется со сложностью?

Сложность системы возникает из-за множества характеристик. Первая – это большое количество компонентов и интерфейсов в системе. Такие системы часто описываются как сложные системы. Вторая – это случайное поведение, например, связанное с финансовыми рынками и поведением клиентов на розничных рынках. Третья – это динамически изменяющиеся организационные структуры, процессы и математические алгоритмы. Четвертая – это реконфигурируемые, обучающиеся системы и организации, а также инновационные методы. Эта последняя форма сложности обычно называется сложностью самоорганизации.

Сложность рассматриваемых систем также характеризуется как структурная сложность, эмерджентная сложность и социально-политическая

сложность. Структурная сложность возникает из-за наличия множества заинтересованных сторон, множества компонентов, множества интерфейсов и множества рабочих процессов. Эмерджентная сложность является результатом наличия неизбежных неопределенностей и большой, и часто неожиданной чувствительности к небольшим изменениям в начальных условиях или ограничениях. Социально-политическая сложность охватывает меняющиеся политические тенденции, противоречивые приоритеты, институциональные ограничения и правила, налагаемые руководящими органами. Решение таких сложностей в MBSE должно начинаться очень рано, при определении системных архитектур и концепций операций.

Для снижения сложности в описании систем использовались различные методы, включая абстракцию, агрегацию, гомогенизацию (т.е. процесс создания компонентов и интерфейсов однородными или схожими), декомпозицию, преобразование и отображение, сокращение конфликтов заинтересованных сторон с помощью методов достижения консенсуса, схемы распределения функций между оборудованием, программным обеспечением и людьми (кибер-физические системы), выбор технологий для облегчения взаимодействия и соответствия стандартам, а также выборочная точность модели и приближенные алгоритмы для снижения вычислительной сложности. Кроме того, были использованы несколько принципов управления сложностью, включая «разделение интересов», стабильные промежуточные формы, структурное выравнивание во время декомпозиции, обнаружение скрытых взаимодействий посредством моделирования на основе историй и использование шаблонов во время интеграции систем, включая интеграцию человек-система (например, шаблон супервизорного управления, шаблон дополненного интеллекта). Структурное выравнивание в декомпозиции относится к минимизации количества интерфейсов между компонентами на разных уровнях иерархии декомпозиции, обычно путем установления взаимно-однозначного соответствия между компонентами на уровнях иерархии. Минимизация количества отображений между архитектурными

перспективами также является способом снижения структурной сложности. Обеспечение стабильных промежуточных форм во время разработки системы снижает сложность «отладки» при диагностике несоответствий и неисправностей. Явное представление контекста в рамках общей онтологической модели обеспечивает удобный способ отделения релевантных факторов от нерелевантных. Последующее использование контекста в качестве фильтра во время выполнения сценария может снизить сложность как во время моделирования, так и в ходе реальной работы. Обеспечение стабильных промежуточных форм во время разработки системы снижает сложность отладки при диагностике несоответствий и неисправностей. Наконец, размер пространства состояний системы и плотность взаимодействия компонентов системы являются ключевыми характеристиками, которые вносят вклад в сложность системы.

Как можно использовать цифровой двойник в MBSE?

Цифровой двойник (DT) подразумевает динамическое цифровое представление физической системы или устройства, которое имеет двунаправленную связь с физической системой и может работать параллельно с физической системой, работающей в реальном мире [15]. В реальности DT может представлять всю физическую систему, подсистему или только определенные аспекты системы, необходимые для определенных приложений. DT динамически обновляется в реальном времени или в почти реальном времени по мере изменения статуса разработки физической системы во время проектирования или по мере изменения состояния физической системы во время эксплуатации, например, из-за износа или воздействия окружающей среды. По определению, полнофункциональный цифровой двойник не может существовать изолированно; для него требуется физический аналог (его физический двойник) для существования в некотором состоянии разработки или использования. Коммуникационная инфраструктура (digital thread) предлагает средства для достижения цифровой связности на протяжении всего жизненного цикла системы или продукта, что обеспечивает

комплексное представление актива на протяжении всего жизненного цикла продукта или системы. Вместе с концепцией цифрового двойника это гарантирует, что данные со всех этапов жизненного цикла системы возвращаются в процесс разработки, концептуализации и создания системы. Учитывая эти характеристики, эти концепции могут эффективно работать вместе с MBSE на всех этапах разработки системы.

Есть ли возврат инвестиций в MBSE?

MBSE может эффективно использоваться для исследования альтернативных системных конструкций и их компромиссов производительность/стоимость. Он также может обеспечить надежную основу для оптимизации общих затрат на весь срок службы, связанных с различными конфигурациями системы. В сочетании с технологией цифровых двойников MBSE может улучшить оптимизацию затрат на жизненный цикл, заменив обслуживание, основанное на времени, на предиктивное и основанное на состоянии обслуживание. Возврат инвестиций (ROI) обусловлен тем фактом, что MBSE обеспечивает необходимую структуру для управления сложностью, быстрого поиска и повторного использования артефактов и возможности исследовать модели и методы систем-кандидатов в моделировании, прежде чем прибегать к тестированию оборудования в дорогостоящем и иногда опасном реальном мире. Поскольку MBSE может поддерживать широкий спектр подходов к моделированию и алгоритмов для обучения и оптимизации, часть ROI будет зависеть от выбора наиболее подходящих методов и алгоритмов. Важно отметить, что организация должна иметь надлежащие обязательства, подготовку и необходимые ресурсы для реализации ROI от принятия MBSE. Важно, что некоторые отрасли (домены), которые испытывают нечастые изменения и работают в стабильной рыночной динамике, могут не получить ту же выгоду, что и отрасли, которые испытывают частые изменения в политике, правилах и процессах, поддерживаемых технологиями [16].

Семантика домена, метамодели и онтологии

В этом разделе представлен обзор метамodelей, семантики и онтологий, которые являются основой методологии MBSE. По своей сути MBSE в значительной степени зависит от согласованной терминологии и отношений между основными системными сущностями: требованиями, функциями и компонентами. Без этой согласованности представления систем с помощью моделей могут быть неясными, неправильно истолкованными и, возможно, конфликтовать с другими представлениями. Далее будут предоставлены рекомендации по разработке и использованию метамodelей и онтологий. Хотя тщательная разработка имеет важное значение, не менее важно использовать метамodelи и онтологии во всей предметной области (домене).

Семантика домена, метамodelи и онтологии являются основополагающими формализмами, которые делают возможными основные преимущества MBSE: совместная работа всех заинтересованных сторон в области системного проектирования, совместное использование моделей, повторное использование моделей, согласованность терминологии и сокращение неоднозначности и ненужной избыточности. Семантика домена (предметной области) – это значение, присвоенное системным концепциям, т.е. «физика», лежащая в основе компонентов системы и их взаимодействия. Метамodelи состоят из обобщенных типов объектов, объектных отношений, атрибутов объектов и композиционных правил. Метамodelи определяют абстрактную семантику и синтаксис для языков моделирования для выражения/представления системной моделей в интересующей области. Онтология – это экземпляр метамodelи для конкретной проблемы или системы. Все метамodelи являются онтологиями, но онтологии не обязательно должны быть метамodelями. Язык моделирования содержит объекты и отношения, используемые при создании модели, которая соответствует метамodelи и представляет аспекты системы [17-19].

Выделим несколько важных моментов:

- Модели помогают получить ответы на ряд конкретных вопросов о предметной области.
- Интерпретация модели требует понимания ее терминологии (онтология).
- Четко определенный и согласованный словарь снижает путаницу в интерпретации модели.
- Модельные словари, основанные на специальной терминологии, могут привести к неверным предположениям о значении модели, что приведет к ошибкам в интерпретации.

Семантика и семантическая область

Семантика касается смысла чего-либо [20]. Семантика рассуждений, которые мы выполняем с использованием модели, и ожидаемые результаты определяют словарь модели. Кроме того, семантика помогает определить область действия модели не только с точки зрения того, что она содержит, но и позволяет определить, когда модель будет достаточно полна для своих целей. Эти моменты важны, поскольку модели создаются для определенных целей и являются полными, когда они удовлетворяют этим целям. Более того, совместное использование моделей, сотрудничество и рассуждения требуют явно определенных словарей [18, 21]. Семантический домен определяет общую терминологию, связанную с определенной областью интересов или проблемой. Создание семантического домена для футбола, понятного нескольким заинтересованным сторонам, относительно просто, поскольку существует признанный контролирующий орган, который определяет терминологию. И наоборот, как и во многих инженерных начинаниях, универсально согласованный семантический домен может оказаться невозможным, и часто попытки его создания могут вызывать неразрешимые разногласия.

Метамодели

Метамоделли определяют абстрактный синтаксис языка моделирования, используемого для описания системных моделей в интересующей области. Метамоделли включают в себя общие типы объектов, отношения между типами объектов, атрибуты типов объектов и правила объединения объектов и отношений. Модели представляют собой семантическое отображение онтологии домена в систему, в то время как метамоделли являются основой онтологий метамоделлей, используемых для определения онтологии домена [18]. Метамоделли могут быть определены иерархически, то есть мета-метамоделль может определять терминологию метамоделли, то есть метамоделль является примером мета-метамоделли. Связь существует между двумя сущностями, когда одна из сущностей является примером другого, то есть является членом или создана из другого субъекта. Собака — это экземпляр домашнего животного, например, и наследует свойства домашний питомец.

Метамодель представляет собой базовый набор концепций, которые используются для создания последовательных и полных моделей. Она определяет семантику и синтаксис нотации моделирования, которая будет использоваться. Несмотря на то, что язык моделирования обязательно будет иметь свою собственную семантику и синтаксис, метамодель адаптирует эти основные свойства языка к конкретной области. Таким образом, метамодель домена предназначена для описания ключевых концепций для области, которые могут служить для обеспечения некоторого стандартизированного подхода для этой области и для обеспечения лучшей интеграции моделей в этой области.

На рисунке 1 показано, что модель является представлением системы. Эта модель является специфическим экземпляром метамодели. Модель выражается на языке моделирования, который использует синтаксис, семантику, ограничения и шаблоны, определенные метамodelью. Существует семантическое отображение метамодели в онтологию метамодели, которая

является специализацией концепций метамодели, необходимых для создания экземпляра онтологии домена (предметной области проекта).

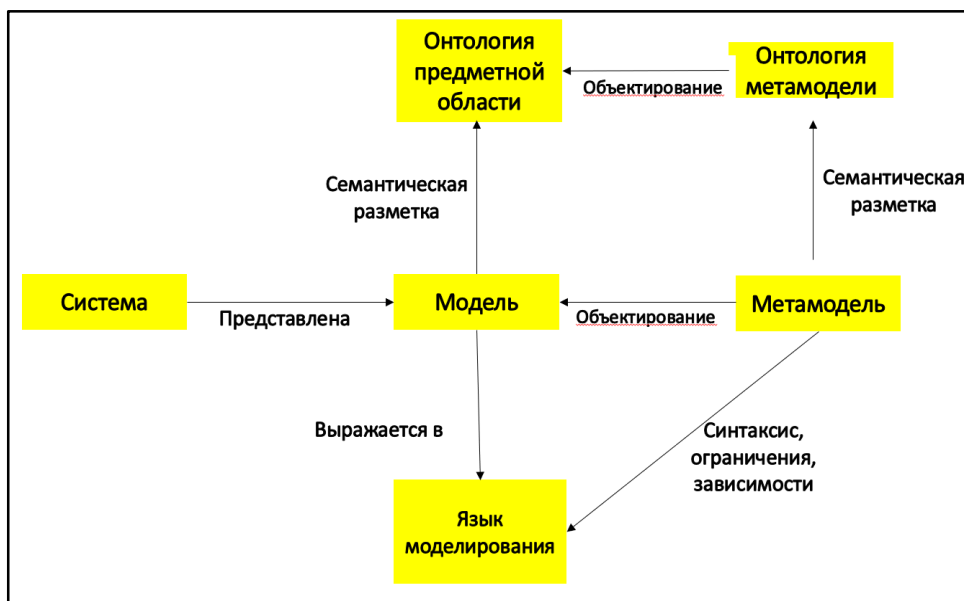


Рис. 1 – Отношения модели, метамодели и онтологий

В строгой метамодели (Metamodel), если элемент модели (Model Element) m_i является экземпляром (instance-of) модели m_j , то каждый элемент в m_i должен быть экземпляром некоторого элемента в m_j . В модели, включающей L уровней, $m_0, m_1, m_2 \dots$ и m_{L-1} , каждый элемент модели уровня m_i должен быть уникальным экземпляром модели уровня m_{i+1} для всех $i < L-1$, и любая другая связь, отличная от связи экземпляра между двумя элементами m_j и m_k , подразумевает, что $\text{level}(j) = \text{level}(k)$, как показано на рисунке 2 [18].

Свободное назначение элементов допускает наследование, ассоциации и другие отношения, выходящие за границы метауровня, что ухудшает целостность модели, если элементы модели сгруппированы в подгруппы схожего назначения. Что еще более важно, нестрогое метамоделирование подрывает устоявшиеся объектно-ориентированные декомпозиции, в которых создание экземпляра класса (инстанциация) может находиться на том же уровне, что и метамодель, экземпляром (instance-of) которой она является [22, 23].

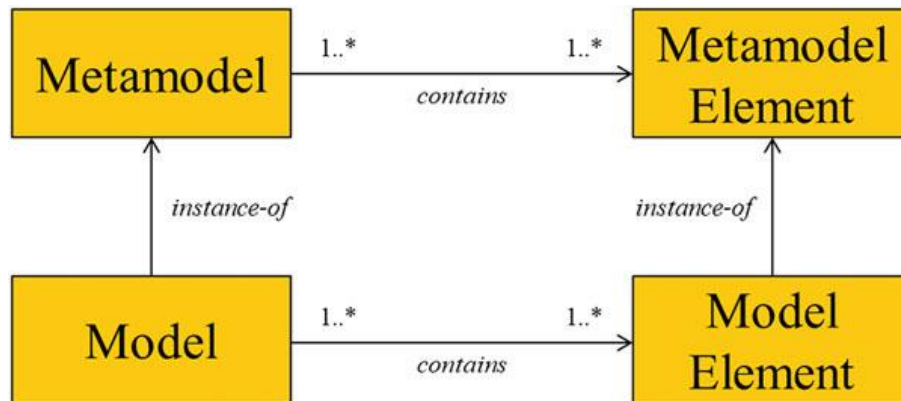


Рис. 2 – Строгая метамодель

Подводя итог, метамодели:

- Содержат правила и грамматику языка системного моделирования
- Имеют мало или вообще не имеют семантического содержания
- Представляют собой общий класс всех моделей, которые могут быть выражены языком моделирования, т.е. модель является реализацией метамодели для заданной семантической области.
- Система представлена моделью, которая соответствует метамодели.

Онтологии

Онтологии – это контролируемые словари, которые включают согласованные наборы явно перечисленных, однозначных и избыточных терминов, отношений и ограничений, необходимых для построения и интерпретации моделей [24, 25]. Эти словари создаются с учетом вклада экспертов в предметной области и конфигурируются контролирующим органом. Онтологии являются основополагающими для MBSE и необходимы для определения области действия модели, ее создания и использования.

Онтологии имеют множество применений, включая следующие:

- Распространение общего понимания того, как структурируется и именуется информация, среди таких субъектов, как заинтересованные стороны и базы данных программного обеспечения.

- Поддержка повторного использования знаний в предметной области путем содействия специализации для нескольких предметных областей.
- Обеспечение необходимости явных предположений о предметной области, что снижает неоднозначность и упрощает внесение изменений, если они потребуются.
- Разделение предметной области и операционных знаний, что позволяет создать систему из ее компонентов, независимую от компонентов и продуктов, которые она производит.
- Обеспечение возможности рассуждения на основе знаний предметной области, когда онтология представлена формальными, декларативными утверждениями.

Онтологии фиксируют соглашение об использовании терминологии и отношений, но не являются, в классическом понимании, ни словарем, ни таксономией. Онтологии обеспечивают согласованную коммуникацию в семантической области, но не обязательно являются полными. Хотя термины «таксономия» и «онтология» иногда используются взаимозаменяемо, эти термины принципиально различны. Таксономии иерархически упорядочены и используются для наименования, описания и классификации терминов в предметной области. Упорядочение таксономии определяется набором последовательных и однозначных правил. И наоборот, онтологии связывают концепции предметной области с отношениями способами, которые поддерживают более глубокое понимание.

Например, на рисунке 3 показана часть таксономии транспортных средств [18, 23]. Основное понятие *Транспортное средство* декомпозируется на конкретные типы транспортных средств: *Воздушное транспортное средство*, *Наземное транспортное средство* и *Водное транспортное средство*. Эта таксономия может ответить на такой вопрос, как «Каковы типы *Воздушных транспортных средств*?» или «Каким *Транспортным средством* является *Велосипед*?». Однако он не может ответить на вопрос: «Сколько

пассажиров помещается в *Весельную лодку?*» без декомпозиции типа *Весельная лодка*, основанной на вместимости, например. Другая декомпозиция, основанная на скорости и типе, необходима, если мы хотим узнать, насколько быстро *Весельная лодка* может перемещаться. Таксономии легко обходят с помощью бинарного и древовидного поиска, кластеризации и т.д.

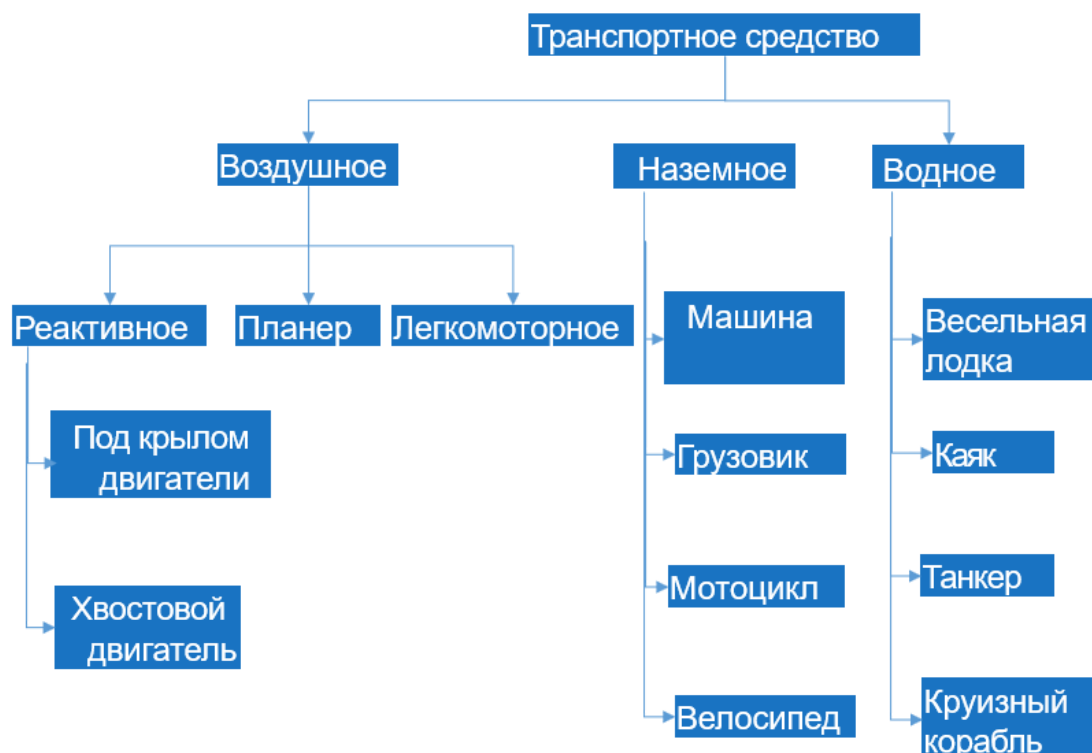


Рис. 3 – Пример таксономии транспортных средств

Онтологии включает понятия и отношения для определенной предметной области, они более формальны и содержат больше информации, чем таксономии. Рисунок 4 показывает онтологию с использованием отношения *это (is-a)*, в котором понятие более низкого иерархического уровня *Водное транспортное средство*, *Наземное транспортное средство* и *Воздушное транспортное средство* приобретают/наследуют свойства родительского понятия «Транспортные средства» [18]. То есть, модель, в которой создается экземпляр *Воздушного транспортного средства*, будет

иметь свойства Грузоподъемность, Ограничения по погоде и Максимальное расстояние в дополнение к унаследованным свойствам Тип двигателя, Тип рулевого управления и Использование. Свойства понятий могут быть конкретизированы по мере необходимости. Рисунок 5 показывает конкретизацию на тип *Транспортное средство* называемого *Гребная лодка Тип 1* и *Тип 2* [18, 23]. Как показано на рисунках 4 и 5 общими компонентами онтологий являются понятия (concepts), отношения, экземпляры (instances), аксиомы, методы конкретизации и правила.

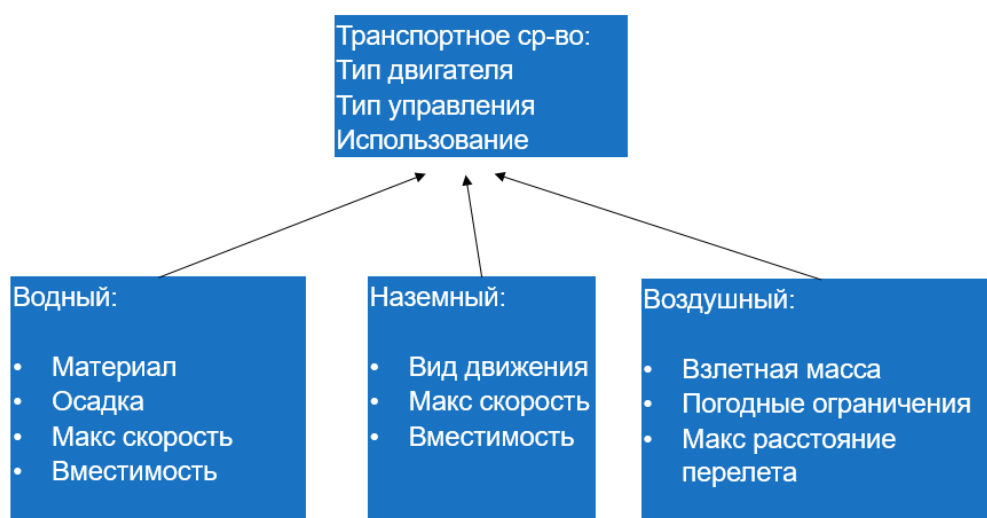


Рис. 4 – Пример онтологии транспортных средств.

Весельная лодка Тип 1	Весельная лодка Тип 2
Материал: Сосна Осадка: 0,3 метра Мах Скорость: 3 узла Мест: 2 Двигатель: весло каноэ Рулевое управление: ножной руль Использование: перевозка	Материал: Дуб Осадка: 0,5 метра Мах Скорость: 3.5 узла Мест: 4 Двигатель: Пара весел Рулевое управление: внебортовой Использование: перевозка

Рис. 5 – Пример конкретизации онтологии транспортных средств.

Понятия: представляют собой вещи или наборы вещей в пределах семантической области, например, *боксер* это понятие в области (домене)

собака. Базовыми (Primitive) являются такие понятия, которые имеют только необходимые условия для принадлежности к классу, например, все члены класса, млекопитающие, должны иметь волосы, три косточки среднего уха, молочные железы и неокортекс (новые области коры головного мозга, которые у низших млекопитающих только намечены, а у человека составляют основную часть коры). Животное, лишенное любого из этих свойств, не может быть млекопитающим. *Определяющие понятия (Defined concepts)* – это свойства, которые являются как необходимыми, так и достаточными для принадлежности к классу. Условие, что рост взрослого человека составляет $\geq 1,5$ м, является необходимым и достаточным условием класса «*Рост не менее или более 1,5 метров*».

Отношения: это связи (associations), созданные между системными сущностями и экземплярами сущностей. Отношения могут определять родительские/дочерние связи, потоки от источника к месту назначения, композицию, распределения, декомпозиции и многое другое.

Экземпляры: конкретные или абстрактные вещи, представленные понятием. В строгом смысле, поскольку онтологии являются концептуализациями предметной области, они не должны включать экземпляры. Концептуализация – это наделение или определение теоретического смысла слов и превращение их тем самым в понятия. Концептуализация есть абстрактная модель некоторого явления в мире, заданная посредством понятий, релевантных этому явлению. Онтологию и связанные с ней экземпляры называются базой знаний.

Аксиомы: утверждения, ограничивающие величины/значения (values) класса и экземпляра. Например, правило может потребовать, чтобы космический корабль имел размеры меньше x , y и z , чтобы поместиться в обтекатель конкретной ракеты-носителя.

Ограничения: позволяют определить, что должно быть истинным для принятия утверждения (assertion) в качестве входных данных. Время запуска

космического корабля должно быть в пределах окна запуска, которое позволяет избежать космического мусора.

Правила: выражение в виде утверждения-гарантии, описывающие логические выводы, сделанные из утверждений. Если мы утверждаем, что космический корабль помещается в определенный обтекатель, то его размеры меньше x , y и z .

Строгость и приверженность метамоделям и онтологиям необходимы для разработки однозначных и хорошо сформированных моделей, и следует предпринять все разумные попытки для создания, поддержания и обеспечения согласованной семантики в масштабах всего проекта. Но не менее важно помнить, что модели создаются для представления определенных аспектов области (домена), определяемой интересами и потребностями заинтересованных сторон. Более того, инженерные проекты имеют ограничения по стоимости и графику, которые задают темп работы и управляют артефактами, создаваемыми в жизненном цикле разработки и развертывания. Как и в большинстве мероприятий по системной инженерии, работа, которая должна быть выполнена, и работа, которая может быть выполнена в рамках стоимости и графика, почти всегда заканчивается сокращением или устранением работы. MBSE не предлагает здесь волшебной палочки, но может способствовать более глубокому пониманию рисков, возникающих при сокращении запланированной работы. Рассматривая конечную цель проекта, которая заключается в предоставлении удовлетворительной системы, разумно подумать о времени и усилиях, необходимых для достижения онтологии проекта золотого стандарта. Также важно признать, что маловероятно, что какая-либо метамодель или онтология будет идеальной. Все системные инженеры хотят иметь модели, которые «достаточно хороши» для выполнения нашей работы, не беспокоясь о каждом нюансе.

Более того, терминология, фокус на иерархии разработки и процессы системной инженерии почти наверняка будут отличаться в крупных

мультиагентских взаимодействиях и кооперациях. Не следует отказываться от концепции единой онтологии при определенных условиях. Однако с точки зрения реального мира надо рассматривать практические аспекты построения онтологии, управления конфигурацией и использования системной модели.

Рассмотрим пример, включающий три полуавтономные организации, разрабатывающие различные компоненты системы анализа и прогнозирования погоды, которая предоставляет набор аналитических продуктов, разработанных на основе данных, полученных с помощью разнообразного набора космических, воздушных и наземных датчиков. Одна организация может отвечать за определение логической архитектуры с точки зрения высокоуровневых функций и интерфейсов. Другая организация может отвечать за создание инфраструктуры для управления запросами ученых-метеорологов, сбора данных о погоде, отправки данных о погоде в соответствующие аналитические приложения и затем доставки обработанных продуктов запрашивающей стороне. На другом уровне находятся ученые-метеорологи, которые указывают, какие данные они хотят получить, когда их следует собирать, какая обработка им нужна и куда направляются результаты. Хотя все эти участники заинтересованы в одной и той же системе, у них разные интересы и они используют разную терминологию. Возвращаясь к предыдущему пункту об ограничениях в усилиях и желании найти «достаточно хорошие» решения, эта трехуровневая система кажется полной неразрешимых проблем.

Программное обеспечение по разработке онтологических моделей - OWL (<https://www.w3.org/OWL>) и Protégé (<https://protege.stanford.edu>) уже предвидят онтологические конфликты посредством использования отношений Equivalent-To и Disjoint-With [26]. Более того, как отмечено в предыдущем разделе, повторное использование названий концепций для разных целей помечается как ошибка. Использование специализированного программного обеспечения является мощным средством для того, чтобы позволить заинтересованным сторонам создавать собственную семантику и

при этом иметь общую согласованную онтологию. Конечно, недостатком является то, что-либо всем сторонам нужен доступ к файлу Protégé, либо кто-то (или назначенная группа) должен владеть файлом Protégé и получать входные данные от всех заинтересованных сторон [27].

Формальные, строгие методы поощряются, когда это практично. Однако реальность такова, что эти методы, вероятно, не являются разумными в большинстве ситуаций. По большей части системные инженеры не знакомы с формальными методами. Но что еще важнее, работа системного инженера заключается в разработке системы, а не в разработке математически доказуемых системных представлений, которые требуют значительных усилий обычно в сжатые сроки. Оценка степени формальности и обеспечение координации в разработке общей онтологии может выиграть, если подумать о факторах, влияющих на согласование точек зрения.

Обычно речь идет о системной модели, разработанной в рамках MBSE, как о размещенной в централизованном репозитории, доступном всем заинтересованным сторонам. Концепция центрального единого источника истины хорошо работает, когда все заинтересованные стороны работают в одном и том же проблемном пространстве, имеют легкий доступ к центральному репозиторию (который может быть распределенным и виртуализированным) и используют идентичные или совместимые инструменты. Однако часто возникают ситуации, в которых одно или несколько из этих условий не выполняются.

Существуют общие типы системного моделирования, которые влияют на форму и содержание онтологии:

1. В *каноническом подходе* модели MBSE, который учитывает многочисленные точки зрения, заинтересованные стороны сотрудничают через *центральный источник истины*. Для этих моделей семантика согласуется с простой онтологией, которая находится в единой иерархической структуре. Такая простая модель является идеальным вариантом, и существуют хорошие инструменты для совместной работы и анализа. Хотя

большие простые онтологии концептуально возможны, для того, чтобы избежать непредусмотренных недостатков согласованности требуется большое внимание к разработке и управлению конфигурацией со стороны владельцев онтологии и разработчиков моделей.

2. *Иерархическое мультимоделирование*, которое включает иерархически объединенные различные модели, использующих уникальные знания и семантику. Хотя каждая модель может использовать свою собственную онтологию, семантика связей между онтологиями должна быть согласована, поскольку несогласованная внутримодельная семантика может вызвать глобальные проблемы для рассуждений о системных процессах. Интегрированные, мультимодельные структуры могут избежать некоторых проблем, которые могут возникнуть при попытке создания единой простой онтологии для различных заинтересованных сторон.

3. *Аморфная неоднородность* включает в себя несколько отдельных моделей, которые произвольно объединяются. Примером является модель, которая допускает несколько одновременных механизмов взаимодействия между компонентами модели. Как правило, аморфные неоднородные модели связаны с онтологической семантической неоднородностью, которая возникает, когда соглашения в масштабах домена либо невозможны, либо непрактичны, либо не нужны. Аморфная неоднородность подразумевает свободную межмодельную семантику, которая может повлиять на рассуждения и сотрудничество в рамках общей картины. Может возникнуть необходимость в отображении, которое переводит семантику из одной онтологии в другую.

Процесс разработки онтологии для методологии MBSE

Как правило, онтология домена должна принадлежать всем заинтересованным сторонам в том смысле, что после ее создания все заинтересованные стороны должны согласиться с ней и будут ее соблюдать. Дополнения или изменения в онтологии контролируются органом управления конфигурацией, так что отдельные заинтересованные стороны не могут вносить изменения в одностороннем порядке. Но, как отмечено выше, может возникнуть неразрешимое расхождение во мнениях.

В разработке Стэнфордского университета описывают здравый смысл и очень практичный процесс из семи шагов разработки онтологии, соответствующий концепции простой модели, описанной выше (<http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html>). Однако этот процесс легко адаптируется к большинству ситуаций.

Шаг 1. Определите домен и область действия

На этом этапе рассматриваются четыре основных вопроса:

1. Какова область применения онтологии?
2. Как будет использоваться онтология?
3. На какие вопросы и проблемы должны быть даны ответы при создании экземпляров онтологии?
4. Кто будет использовать и поддерживать онтологию?

Ответы на эти вопросы не будут и не должны быть статичными, а скорее отражают понимание на определенный момент времени в разработке онтологии и модели(ей), которые создаются на ее основе.

На вопрос о предметной области (домене) следует ответить как можно более подробно, но при этом учесть факторы, представляющие интерес для всех заинтересованных сторон. Скорее всего, не желательно, чтобы домен был слишком большим, как, например, все космические аппараты, а например,

ограничен на космических аппаратах, которые предоставляют информацию о погоде. Со временем эта область может расшириться до сбора данных о погоде с самолетов и наземных датчиков, необходимых для учета мнений заинтересованных сторон.

Второй вопрос связан с пониманием того, будет ли онтология использоваться исключительно для рассуждений, как это может быть выполнено в Protégé, в одной или нескольких моделях MBSE или в комбинации того и другого. При использовании онтологии для логических рассуждений обязательно потребуется значительная строгость, чтобы результаты были последовательными и надежными. При использовании в качестве основы для моделей строгость может быть снижена, как отмечалось выше, но снижение строгости может привести к путанице и несоответствиям.

На вопрос 3 можно ответить, создав набор вопросов о проекте. Это вопросы, ответы на которые заинтересованные стороны хотели бы получить с помощью онтологии или ее реализации. Примерами вопросов для метеорологического космического аппарата могут быть следующие:

1. Какие датчики необходимы для измерения содержания воды в снегу?
2. Какие датчики необходимы для измерения скорости ветра?
3. Какие датчики необходимы для измерения температуры грунта?
4. Какие датчики необходимы для измерения атмосферного давления?
5. Какие датчики необходимы для измерения содержания углекислого газа в атмосфере?
6. Как система может оценивать изменения климата и концентрацию углекислого газа?
7. Какой угол наклона обеспечивает стабильные измерения независимо от положения солнца?
8. Какая погода сейчас в Лондоне?
9. Каков прогноз погоды в Лос-Анджелесе на 5 дней?
10. Какая орбита покрывает наибольший процент поверхности Земли?

Шаг 2: Найдите и проанализируйте существующие онтологии

Онтологии были созданы для очень многих областей, которые могут быть идентичны или очень похожи на интересующую область. Во многих случаях эти онтологии были хорошо протестированы и использовались большим сообществом и, следовательно, заслуживают рассмотрения, даже если требуется адаптация для конкретного приложения. Конкретные онтологии легко найти с помощью веб-поиска, в ссылках [28] или при просмотре приведенного ниже сокращенного списка онлайн-онтологий:

- Indiana University:

<https://info.sice.indiana.edu/~dingying/Teaching/S604/OntologyList.html>

- EMBL-EBI: <https://www.ebi.ac.uk/ols/ontologies>
- University of Michigan: <https://guides.lib.umich.edu/ontology/ontologies>
- W3C: https://www.w3.org/wiki/Good_Ontologies
- Aber-OWL: <http://aber-owl.net/ontology>
- Protégé Ontology Library:

https://protegewiki.stanford.edu/wiki/Protege_Ontology_Library

Шаг 3: Перечислите ключевые термины

На этом этапе заинтересованные стороны составляют неофициальный список ключевых терминов, которые необходимы для решения вопросов о компетентности или желаемых тем для обсуждения. Для представления терминов часто используют таксономии (см. предыдущий раздел). Список включает в себя свойства термина, если они известны, и некоторое представление о том, что важно в этом термине. Для метеорологического спутника важными параметрами могут быть масса, орбита, типы измерительных приборов, точность измерительных приборов, покрытие Земли и так далее.

Шаги 4 и 5: Определение понятий, иерархии понятий и свойств понятий

Ключевые термины и вопросы о компетентности определяют концепции (понятия), структуру и свойства онтологии. На шаге 4 разрабатываются классы и иерархия классов, что эквивалентно определению концепций (понятий) онтологии и иерархии понятий. На шаге 5 создаются свойства класса для классов, указанных на шаге 4. Однако также следует отметить, что, поскольку шаг 4 и шаг 5 тесно вплетены, рекомендуется выполнять их параллельно. Концепции и иерархия концепций могут быть определены многими способами, которые можно условно разделить на три категории: нисходящие, восходяще-ориентированные и комбинированные:

Нисходящий подход начинается с перечисления наиболее общих понятий предметной области, а затем разбивается на специализации. Например, мы могли бы начать с понятия “космический корабль” и создать дополнительные понятия “определение ориентации и управление, телекоммуникации, распределение электроэнергии, управление командами и обработка данных, терморегулирование, двигательная установка и контрольно-измерительные приборы”.

Подход “*снизу вверх*” (восходяще-ориентированный) начинается с наиболее специфических понятий, которые затем группируются в более общие. Например, мы могли бы начать с реактивного колеса, звездного указателя и инерциальной единицы измерения и обобщить их как определение ориентации и управления.

Комбинированный подход использует как нисходящий, так и восходящий подход в соответствии с каждым понятием. В некоторых случаях конкретные вспомогательные понятия более низкого уровня иерархии могут быть еще неизвестны, и в этом случае используется нисходящий подход. В других ситуациях может быть еще неясно, как сгруппировать понятия в наиболее общие, поэтому восходящий подход может позволить сгруппировать промежуточные понятия.

Ни один из подходов не лучше другого, и его лучше выбирать, исходя из предпочтений и точек зрения заинтересованных сторон, а также уровня детализации, доступного в предметной области. Однако одного шага 4 недостаточно для полного ответа на вопросы об однозначности понятий. Необходима дополнительная информация, поэтому к понятиям на шаге 5 добавляются свойства, которые описывают внутренние детали понятий, разработанных на шаге 4.

Многие из нераспределенных по иерархии терминов из шага 3, вероятно, являются свойствами понятий из шага 4. Каждый нераспределенный термин исследуется и определяется понятие, которое наилучшим образом его описывает. Некоторые термины могут относиться к нескольким понятиям, и для некоторых понятий могут потребоваться свойства, которых нет в списке ключевых терминов. Однако рекомендуется, присваивать свойства наиболее общему понятию, соответствующему данному свойству. Например, если наиболее общим понятием в предметной области является системная электроника, то добавьте свойство *масса* сюда, а не в понятия более низкого иерархического уровня, потому что все следующие по иерархии понятия унаследуют свойство *масса* от системной электроники, что делает онтологию менее подверженной путанице и более доступной для изменения при необходимости.

Термины, выделенные для формирования понятий, становятся, слотами, включенными в понятие. Во фреймовом представлении знаний, слот – это элемент фрейма и обычно один фрейм содержит множество слотов, хотя может состоять и из одного, а слот представляется в виде (Имя_слота. Значение_слота) [29]. Существует четыре основных типа свойств:

Внутренние свойства – это аспекты понятия, которые должны существовать, если понятие существует. Например, физический компонент будет иметь массу.

Внешние свойства – это внешние свойства, например, такие как название производителя, поставившего физический компонент.

Частные свойства определяют физические или логические объекты, которые определяют иерархию понятий, например, реактивное колесо состоит из контроллера, интерфейсной платы, двигателя и вращающейся массы.

Взаимосвязи объясняют, как понятие взаимодействует с другими понятиями, зависит от них, используется ими и т.д.

Шаг 6. Определение слотов фасетов

Фасет – это группа однородных терминов, связанных общностью какого-либо признака (характеристики основания деления). Служит средством построения информационно-поисковых языков фасетной структуры. Фасетная классификация значительно облегчает многоаспектное отражение объектов, поскольку можно строить классы из различных сочетаний признаков [30-32]. Фасеты – это ограничения на такие области, как мощность, диапазон значений и единицы измерения. Не для всех областей требуются фасеты, и чрезмерное использование фасетов может затруднить использование онтологии, сделав ее чрезмерно ограниченной. Учтите, что если онтология ограничивает массу электронного компонента, то любой компонент, выходящий за рамки этого ограничения, не может существовать в экземпляре онтологии. К типичным типам фасетов относятся следующие:

Строки (Strings): содержит текстовую информацию.

Число (Number): обычно указывается в виде числа с плавающей запятой или целого числа.

Логический (Boolean): значение должно быть либо истинным, либо ложным.

Перечисление (Enumerated): указывает допустимый диапазон значений.

Экземпляр (Instance): указывает, что слот является экземпляром понятия, разрешенные понятия являются его возможными значениями, а понятие, которому принадлежит данный экземпляр, это его домен.

Рекомендуется учитывать дополнительные факторы для слота
Экземпляр (Instance):

Используйте самые общие понятия при определении диапазона и домена, но слишком общий выбор может вызвать проблемы. В слоте Экземпляр (Instance) должны быть описаны все соответствующие п домена, и все экземпляры классов диапазона должны относиться к этому слоту. Однако, если вещь является наиболее общим понятием, то использование ее для диапазона экземпляра является слишком широким.

Удалите понятия более низкой иерархии, если диапазон экземпляра включает как понятия, так и ее подпонятия.

Добавьте понятие в диапазон или домен, если слот содержит все свои подпонятия, но не само понятие.

Если список понятий в пределах диапазона или предметной области слота содержит только подмножество подпонятий, то это понятие следует пересмотреть для использования только в качестве определения диапазона.

Шаг 7: Создание экземпляров

На этом заключительном этапе определяются экземпляры понятий путем выбора понятия, создания конкретного экземпляра этого понятия и последующего присвоения значений интервалов, соответствующих аспектам интервалов. Как правило, экземпляры становятся функциями моделей, но, что более важно, помогают ответить на вопросы о компетентности. Существует несколько размытая грань между тем, что должно быть понятием, и тем, что должно быть экземпляром понятия. В определенной степени решение может основываться на назначении онтологии и необходимой степени детализации для оценки компетенций. Другое соображение заключается в том, что чрезмерно специфичная онтология может ограничить повторное использование.

Применение методологии MBSE

Методология определяется как совокупность взаимосвязанных процессов, методов и инструментов [33-35]. Методология MBSE таким образом, можно охарактеризовать как совокупность взаимосвязанных процессов, методов и инструментов, используемых для поддержки дисциплины системной инженерии в «основанном на моделях» или «управляемом моделями» контексте [8, 36]. При этом следует отметить, что во многом методология MBSE является развитием идей известного российского мыслителя философа Щедровицкого Г.П., который рекомендовал при решении проблем и формировании последовательности реализации проектов придерживаться следующей последовательности ответов на вопросы: что?, почему?, зачем? и как? [37].

Слово «методология» часто ошибочно считается синонимом слова «процесс». Рекомендуется использовать следующие определения для того, чтобы различать понятия методологии от процесса, методов и инструментов:

Процесс (Р) – это логическая последовательность задач, выполняемых для достижения определенной цели. Процесс определяет «ЧТО» (“WHAT”) должно быть сделано, не уточняя «КАК» (“HOW”) выполняется каждая задача. Структура процесса обеспечивает несколько уровней агрегации, что позволяет проводить анализ и определение на разных уровнях детализации для поддержки различных потребностей в принятии решений.

Метод (М) состоит из приемов выполнения задачи, другими словами, он определяет «КАК» для каждой задачи. (В этом контексте слова «метод», «техника», «практика» и «процедура» часто используются взаимозаменяемо.) На любом уровне задачи процесса выполняются с использованием методов. Однако каждый метод также сам по себе является процессом с последовательностью задач, которые должны быть выполнены для этого конкретного метода. Другими словами, «КАК» на одном уровне абстракции становится «ЧТО» на следующем более низком уровне.

Инструмент (Т) – это инструмент, который при применении к определенному методу может повысить эффективность задачи, при условии, что он применяется правильно и кем-то с надлежащими навыками и подготовкой. Целью инструмента должно быть содействие выполнению «КАК». В более широком смысле инструмент улучшает «ЧТО» и «КАК». Большинство инструментов, используемых для поддержки системной инженерии, основаны на компьютерах или программном обеспечении, что также иногда называются как инструменты автоматизированного инжиниринга (Computer-Aided Engineering – CAE tools).

На основе этих определений методологию можно определить как совокупность связанных процессов, методов и инструментов. Методология по сути является «рецептом» и может рассматриваться как применение связанных процессов, методов и инструментов к классу проблем, которые имеют что-то общее. Расширяя это определение немного дальше, методологию MBSE можно охарактеризовать как совокупность связанных процессов, методов и инструментов, используемых для поддержки дисциплины системной инженерии в «модельно-ориентированном» или «управляемом моделью» контексте.

С вышеприведенными определениями процесса, методов (и методологии) и инструментов связана среда. Среда (Е) состоит из окружения, внешних объектов, условий или факторов, которые влияют на действия объекта, отдельного человека или группы. Эти условия могут быть социальными, культурными, личными, физическими, организационными или функциональными. Целью среды проекта должна быть интеграция и поддержка использования инструментов и методов, используемых в этом проекте. Таким образом, среда делает возможным (включает) или невозможным (отключает) «ЧТО» и «КАК».

Наглядное изображение, иллюстрирующая взаимосвязь между так называемыми элементами «PMTE» (процесс, методы, инструменты и среда),

представлена на рисунке 6 вместе с влиянием технологий и людей на элементы PMTE.

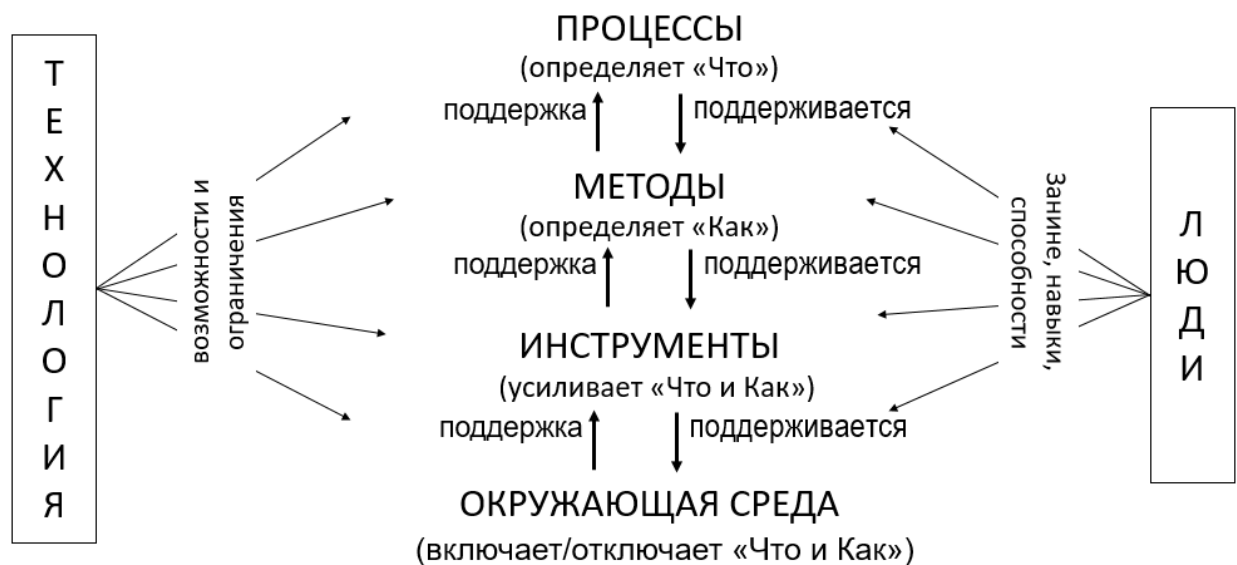


Рис. 6 – Взаимосвязь между элементами “PMTE” (процесс, методы, инструменты и окружающая среда)

В работе [34] указывается, что возможности и ограничения технологии должны учитываться при разработке среды разработки системной инженерии. Этот аргумент распространяется, конечно, и на среду MBSE. Технология не должна использоваться «только ради технологии». Технология может либо помогать, либо мешать усилиям по системной инженерии. Аналогично, при выборе правильного сочетания элементов PMTE необходимо учитывать знания, навыки и способности (the knowledge, skills, and abilities – KSA) вовлеченных людей [33-36]. Когда используются новые элементы PMTE, часто KSA людей должны быть улучшены посредством специального обучения и специальных заданий.

Методология MBSE использует подход на основе моделей для представления различных артефактов, созданных в ходе разработки, часто используя язык SysML [38, 39] в качестве преобладающего языка моделирования. Таким образом, это позволяет системному инженеру точно фиксировать, анализировать и определять систему и ее компоненты, а также

обеспечивать согласованность между различными представлениями системы. Артефакты моделирования также могут быть уточнены и повторно использованы в других приложениях для поддержки линейки продуктов и эволюционных подходов к разработке. На рисунке 7 показано, что по мнению авторов [34, 35] методология включает в себя следующие действия по разработке:

- Анализ потребностей заинтересованных сторон
- Определение требований к системе
- Определение логической архитектуры
- Синтез кандидатов на выделенные архитектуры
- Оптимизация и оценка альтернатив
- Проверка и верификация системы

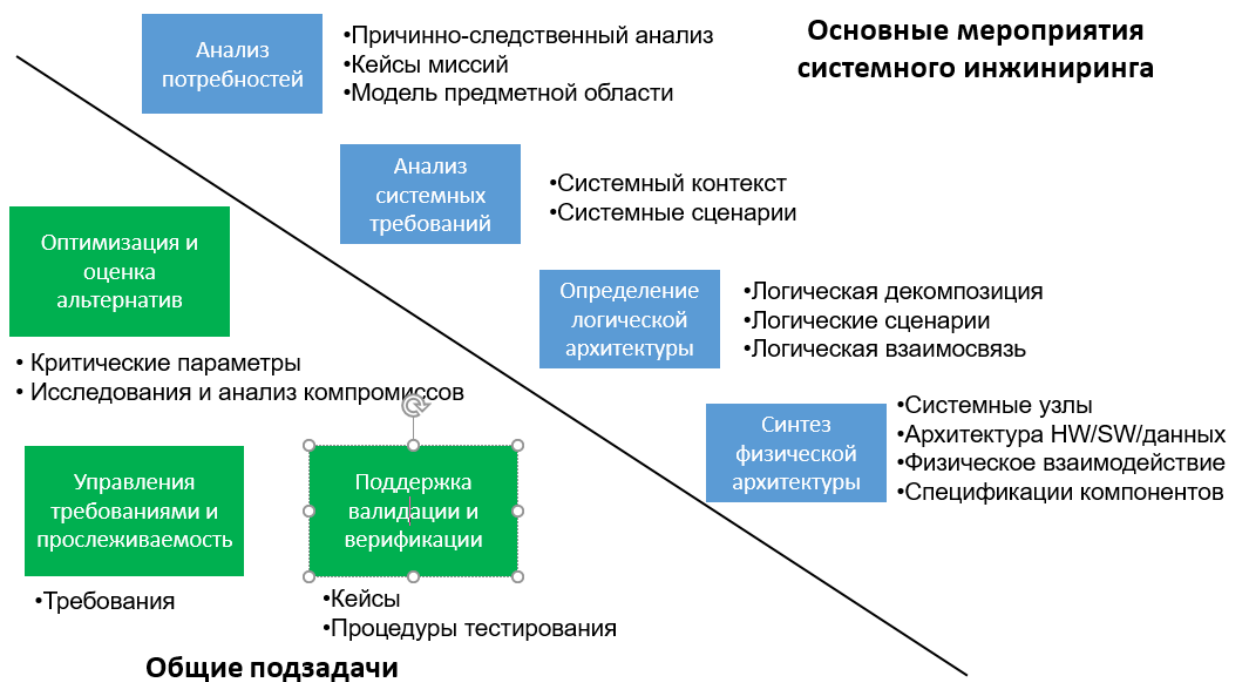


Рис. 7 – Основные артефакты метод объектно-ориентированной системной инженерии

Анализ потребностей заинтересованных сторон включает анализ существующих систем и предприятий, связанные с ними ограничения и

потенциальные области улучшения. Общая структура определения потребности заинтересованных сторон показана на рисунке 8. Результаты анализа «как сейчас есть» используются для разработки будущей системы для будущего состояния, например, промышленного предприятия, и связанных с ним требований миссии. Модель предприятия отображает предприятие, его составляющие системы, включая системы, которые должны быть разработаны или изменены, и субъектов предприятия (сущности, внешние по отношению к предприятию). «Как есть» предприятие анализируется с использованием методов причинного анализа для определения его ограничений и используется в качестве основы для вывода требований миссии и модели будущего предприятия. Требования миссии указываются в терминах целей миссии/предприятия, мер эффективности и вариантов использования верхнего уровня. При этом варианты использования и сценарии охватывают функциональность предприятия.

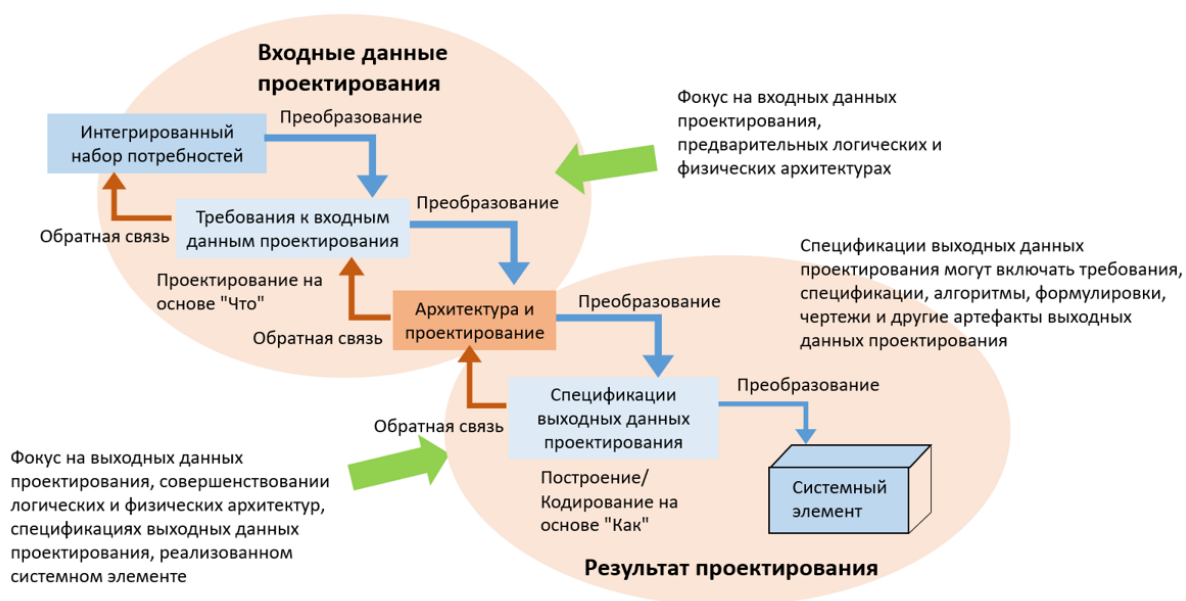


Рис.8 – Определение потребностей заинтересованных сторон

Создание интегрированного набора потребностей гарантирует, что в процессе определения потребностей заинтересованных сторон будут проанализированы все перспективы, включая риски, движущие силы,

ограничения, а также анализ и проработку концепций жизненного цикла. Определение потребностей заинтересованных сторон в общем случае направлено на формирование концепции будущей системы (рис. 9). При этом следует помнить, что важнейшая задача внедрения MBSE на действующем предприятии – это поиск путей снизить стоимость эксплуатации объектов за счет следующих эффектов:

- установление общего понимания структуры и значения информации,
- повышение качества проекта за счет точных семантически/синтаксически однозначных системных моделей,
- улучшенные коммуникации,
- повышенная способность управлять сложностью системы,
- улучшенное качество продукции,
- расширение знаний и повторного использования информации,
- визуализация компонентов и процессов,
- расширенные возможности учить и учиться для персонала.

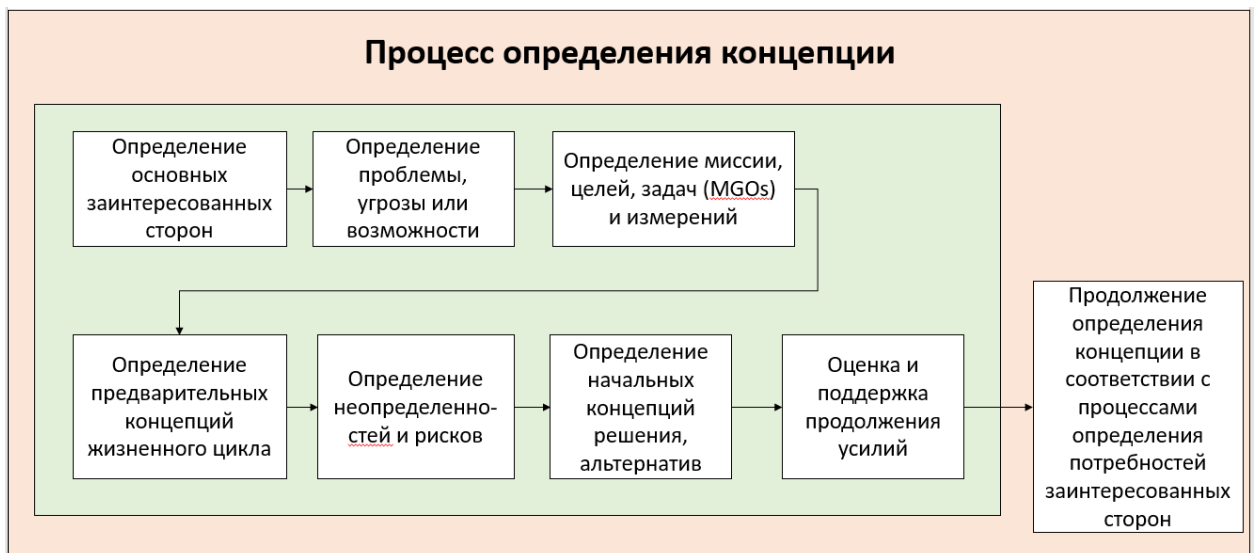


Рис. 9 – Определение концепции системы. Анализ бизнеса, формулировка миссии

Методология MBSE тесно связана с понятиями жизненного цикла. Модель жизненного цикла состоит из этапов жизненного цикла, которые

обычно включают стадии концепции, реализации, производства, поддержки, использования и вывода из эксплуатации.

Ряд моделей разработки жизненного цикла был создан и применен к крупномасштабным проектам разработки систем и программного обеспечения, используемым в правительстве, промышленности и научных кругах, но большинство из них основано на одной из трех основополагающих моделей. Модель Ройса водопада [40], Бёма – спиральная модель [41], а также Форсберга и Муга «Модель Ви» [42]. Графическое изображение каждой из этих моделей разработки жизненного цикла схематично показано на рисунке 10 [33].

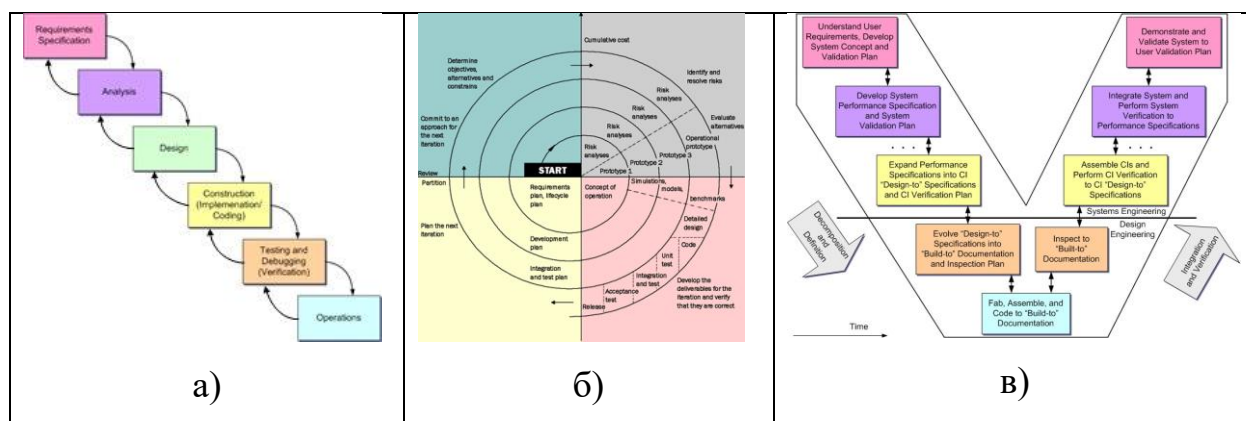


Рис. 10 – Основополагающие модели развития жизненного цикла: (a) Водопад, (b) Спираль, (c) «V-образная».

Существует большое количество литературы, описывающей каждую из этих моделей. Поэтому подробное описание каждого из них здесь не приводится. Достаточно сказать, что вариации водопадной и спиральной моделей для поддержки структурированной, а также итеративной и поэтапной разработки широко использовались в проектах разработки программного обеспечения, в то время как модель «Vee» и модифицированные версии «Vee» широко применялись в области системной инженерии и разработки систем.

Помимо признания существования таких основных моделей разработки жизненного цикла, они также могут служить метамодели для разработки

жизненного цикла. Другими словами, они предоставляют шаблоны разработки жизненного цикла, на основе которых строятся планы для конкретного проекта или предметной области в рамках общей методологии MBSE. Процесс системной инженерии (SE) – это модель процесса, которая определяет основные действия («ЧТО»), которые необходимо выполнить для внедрения системной инженерии. Процессы SE связаны с фазами в модели жизненного цикла, поскольку процесс обычно начинается на ранней стадии жизненного цикла системы, как правило, в самом начале проекта; однако в некоторых случаях процесс SE может также начинаться в середине жизненного цикла.

Итак, жизненный цикл представляет собой ценность для заинтересованных сторон и решение проблем, описываемые как набор этапов жизненного цикла, на которых можно изучать и решать проблемы, а также управлять ресурсами. Процессы жизненного цикла: системы мероприятий, направленных на создание и обмен знаниями, связанными с системным подходом, которые могут быть использованы для продвижения целостного подхода на протяжении всего жизненного цикла (рис. 11).

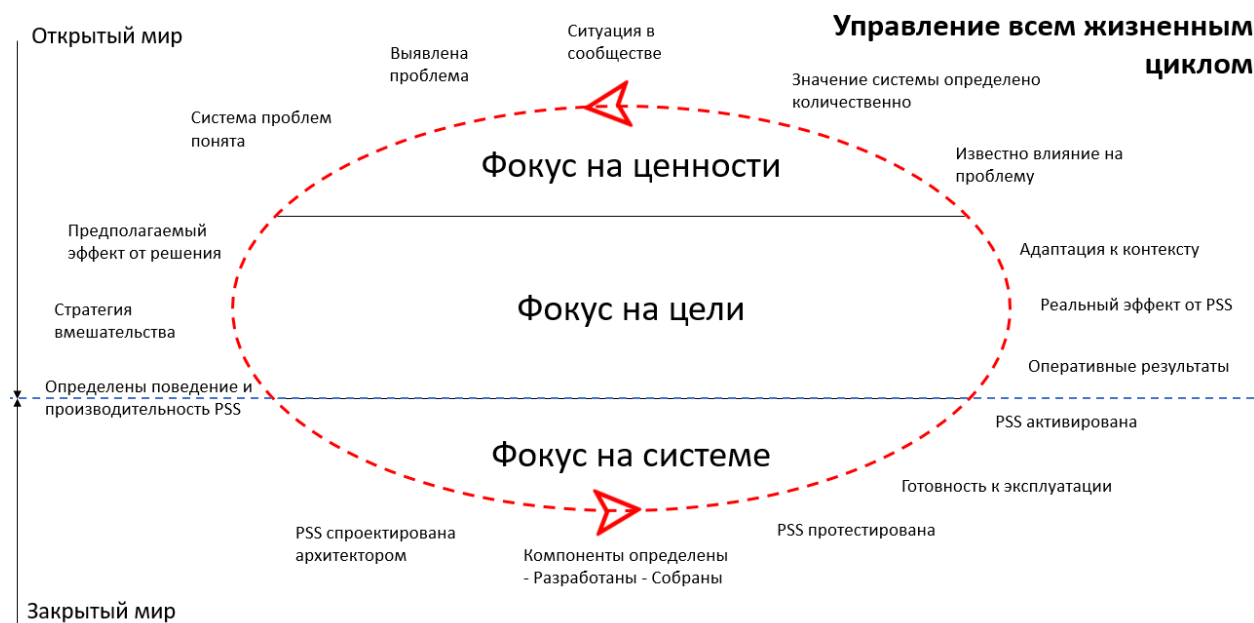


Рис. 11 – Управление жизненным циклом

Проектирование на основе моделей определяет повышение роли моделей в процессе реализации проекта системы до центральной и ведущей роли в спецификации, проектировании, интеграции, проверке и эксплуатации системы. Для многих организаций – это сдвиг парадигмы от традиционных подходов, основанных на документах и моделях жизненного цикла, многие из которых следуют «чистой» водопадной модели определения системы, проектирования системы и квалификации проекта. Один из самых больших коммуникационных барьеров, существующих между традиционными дисциплинами инженерного проектирования (включая дисциплину системного проектирования) и MBSE, заключается в том, что в процессе, основанном на моделях, действия, поддерживающие процесс проектирования, должны выполняться посредством разработки все более подробных моделей. Эта коммуникационная пропасть существует уже много лет, и многие менеджеры и практики до сих пор не осознают тот факт, что различные модели процессов MBSE и поддерживающие их методы призваны продемонстрировать акценты, а не быть не просто водопадом, и что вся модель системы со временем расширяется и растет.

Таким образом жизненный цикл системы начинается со стадии «определения концепции». На первых шагах этой стадии заказчик (или предприятие) идентифицирует новые желательные возможности системы в виде потребностей (needs) заинтересованных сторон. Затем проводится бизнес-анализ или анализ миссии, результатом которого являются высокоуровневый набор стратегий и потребностей, выраженные в виде бизнес-требований, отражающих миссию проекта. Далее проводится анализ возможных решений, направленных на удовлетворение потребности заинтересованных сторон, с целью уточнений потребностей и их трансформации в требования заинтересованных сторон (представление системы с точки зрения пользователя) Следующей стадией жизненного цикла систем является стадия «определения системы». На этой стадии требования

заинтересованных сторон преобразуются в системные требования. Описанные выше процессы и составляют основу инженерии требований.

Определение системных требований предназначено для определения требований, которые соответствуют требованиям миссии. Система моделируется как черный ящик, который взаимодействует с внешними системами и пользователями, представленными в модели предприятия. Варианты использования и сценарии на уровне системы отражают операционную концепцию использования системы для поддержки предприятия. Сценарии моделируются с использованием диаграмм деятельности, которые представляют систему «черного ящика», пользователей и внешние системы. Сценарии для каждого варианта использования используются для определения функциональных требований, интерфейса, данных и производительности системы «черный ящик». Во время этого действия база данных управления требованиями обновляется, чтобы отслеживать каждое системное требование в соответствии со сценарием использования уровня предприятия/задачи и требованиями миссии.

Инжиниринг требований (requirements engineering) один из важнейших разделов SE, от качественного исполнения которого целиком и полностью зависит успешность любого проекта по созданию и использованию инженерных систем. Под инжинирингом требований понимается ряд последовательных действий по разработке, декомпозиции, использованию спецификаций пользовательских потребностей и требований, включая требований заинтересованных сторон и системных требований, с переходом к требованиям проектирования низкого уровня.

Требование – это утверждение, которое идентифицирует эксплуатационные, функциональные параметры, характеристики или ограничения проектирования продукта или процесса, которое однозначно, проверяемо и измеримо. Требования необходимы для приемки продукта или процесса (потребителем или внутренним руководящим принципом обеспечения качества) ISO 29148 (<https://analytics.infozone.pro/requirements-in->

iso-iec-29148). Требование выражает условие или возможность, которая должна выполняться системой или системным элементом, чтобы соответствовать контракту, стандарту, спецификации или другому официально установленному документу. Требование определяет ожидаемую количественную или качественную характеристику, или свойство объекта, а также связанные ограничения и условия [43]. В жизненном цикле разработки и поддержки системы обычно выделяют несколько классов требований, каждый из которых имеет своё назначение. Ниже перечислены основные классы требований и их назначение [22, 44]:

Требование заинтересованных сторон (Stakeholder Requirement) или требования пользователя (User requirements - StRS) включают: требуемые входные данные/выборы/информационные наблюдения, которые пользователи/операторы/специалисты по обслуживанию должны выполнять посредством использования системы; выходные данные, которые им требуются от системы для выполнения этих задач; любые применимые условия или ограничения, регулирующие их взаимодействие с системой.

Бизнес-требования (Business Requirement) определяют включают: цели и задачи бизнеса, которые должна решать система, функциональность системы

Системные требования (System Requirement), включают следующие классы требований проекта:

- Функциональные требования (Functional requirements)
- Требования к удобству использования (Usability requirements).
- Требования к производительности (Performance requirements).
- Системные интерфейсы (System interfaces)
- Требования к интеграции человека и системы (Human system integration requirements)
- Требования к ремонтпригодности (Maintainability)
- Требования к надежности (Reliability)
- Физические требования (Physical requirements)

- Требования к адаптивности (Adaptability requirements)
- Требования к безопасности системы (System security)
- Нефункциональные требования (Non-functional requirements)
- Требования к данным (Data requirements)
- Требования к процессам (Process requirements).

Требования заинтересованных сторон, требования к системе и требования к элементам системы должны быть четко сформулированными [2, 22, 32]. Для правильно сформулированного требования имеется возможность проверить удовлетворяет ли это утверждение о проектируемой системе потребностям конечного пользователя (заказчика или заинтересованного лица). Требования могут быть записаны/написаны на формальном или естественном языках. Все термины, используемые для разработки требований, должны быть формально определены.

В требовании должен быть указан субъект (subject) требования (например, система, программное обеспечение и т. д.) и то, что должно быть сделано (shall be done) (например, выполнять действия в соответствии с установленным планом). В случае выражения в форме естественного языка, требование должно строиться по следующим синтаксическим правилам:

[Условие] [Субъект] [Действие] [Объект] [Ограничение]

ПРИМЕР: Когда на вход системы приходит сигнал «цель обнаружена» [Условие], система [Субъект] должна создать и инициировать [Действие] процесс «сопровождение цели» [Объект] в течение 100 мсек [Ограничение]

Или

[Условие] [Действие или ограничение] [Значение]

ПРИМЕР: При вхождении цели в зону обнаружения [Условие] радиолокационная система должна обнаруживать цель на расстоянии до [Действие или ограничение] 100 км [Значение]

Или

[Субъект] [Действие] [Значение]

ПРИМЕР: Система обслуживания счетов [Субъект] должна отображать ожидающие счета клиентов [Действие] в порядке возрастания [Значение], в котором счета должны быть оплачены

В общем случае в жизненном цикле создаваемой инженерной системы должны реализовываться следующие процессы требований:

Процесс определения требований заинтересованных сторон

Процесс анализа требований

При этом эти процессы носят итерационный характер и приводят к построению иерархической структуры требований (архитектуры требований). Подчеркнем, что формулировка должна быть построена на основе ранее созданной онтологии.

Построение системной архитектуры в MBSE

Итак, MBSE является дальнейшим развитием методологии системной инженерии, границы управления которой показаны на рис. 12 [45]. Как видно из рисунка, применение методов SE охватывает различные этапы жизненного цикла систем, включая:

- инженерия требований
- определение архитектуры
- тестирование
- испытания
- развитие решений при рассмотрении проблемы в целом от исследования концепции системы до ее утилизации



Рис. 12 – Взаимосвязь трех дисциплин - Системной инженерии, реализации систем (system implementation), и Управления проектами/системами (project/systems management)

Хотя традиционно MBSE используется в проектах создания новых технических объектов, тенденции цифровой трансформации промышленности требуют разработки к применению MBSE для цифровизации и существующих объектов/предприятий [46]. Для того чтобы в ходе цифровой трансформации использовать все преимущества цифровых технологий, необходимо создать цифровое представление в виртуальном (цифровом) мире и связать его с физическим объектом в реальном мире двусторонними информационными связями. Другими словами, речь идет о построении в виртуальном мире цифрового представления объекта, существующего в реальном мире (рис. 13).

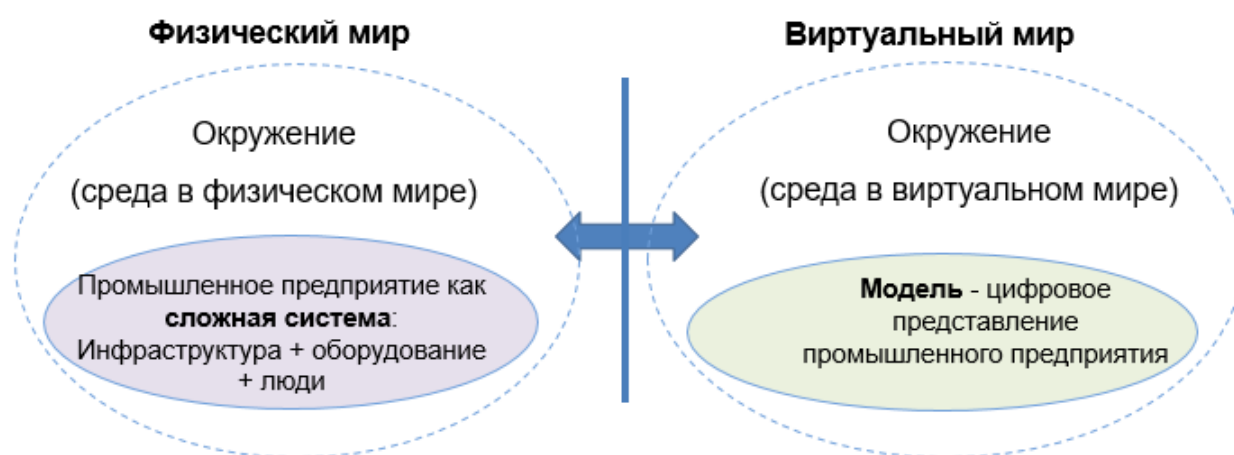


Рис. 13 – Сложные технические системы (существующие промышленные предприятия) в реальном (физическом) и виртуальном (цифровом) мирах.

При этом следует учитывать, что в типичном наборе сущностей MBSE подхода: требования – функции – компоненты, последние являются самыми очевидно выделяемыми, с которых можно начинать построение архитектурных моделей [1, 3]. Но при этом следует учитывать иерархию типологий моделей MBSE или по терминологии Кондратьева В.В. «опорные нотации моделирования систем» [8] (рис.14):

Онтологические модели – системы понятий

Архитектурные модели – представление основных сущностей (требований, функций и компонентов) в виде иерархической структуры. При этом формируются матрицы связанностей или, другими словами, взаимного влияния сущностей: требования-требования, требования-функции, требования-компоненты, функции-функции, функции-компоненты и, наконец, компоненты-компоненты.

Параметризированные архитектурные модели – к каждому элементу архитектурной модели добавляется один или несколько атрибутов/характеристик/параметров. При этом строятся матрицы взаимного влияния сущностей, подобных построенным в предыдущем пункте. Такой подход позволяет формализовать взаимосвязь параметров, что является важнейшим условием для следующего шага по формализации математических моделей процессов и явлений в сложной технической системе.

Математические модели процессов и явлений в сложной технической системе

Компьютерные цифровые модели, реализующие математические модели, построенные на предыдущем шаге.

Модель – это упрощенное представление некоторого аспекта реальной системы. Модели недолговечны: они создаются, объясняют явление, стимулируют обсуждение, выдвигают альтернативы, а затем заменяются новыми моделями. Инженеры знают, как построить модель, но довольно часто они пропускают несколько шагов поэтому далее представлено краткое описание процесса моделирования.

Обнаружение требований, изучение компромиссов и анализ рисков – это три различных вида деятельности по системной инженерии. Несмотря на то, что они имеют одинаковую базовую структуру процесса, они кажутся разными, поскольку используют разные словари, входы и выходы.

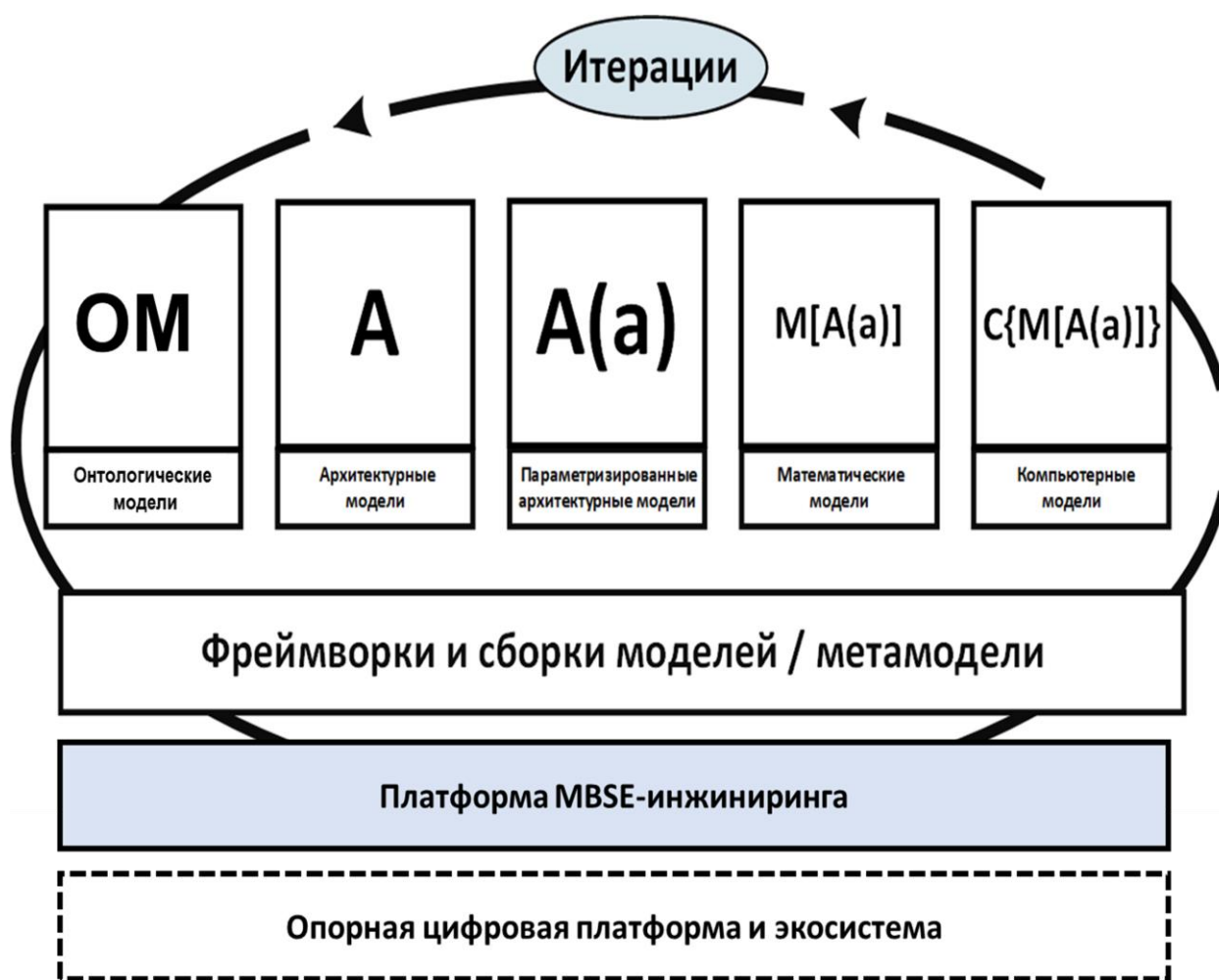


Рис. 14 – Итерационный процесс моделирования систем в MBSE.

В целом методология основана на упорядочивании методов и инструментов архитектурного моделирования целевых технических объектов как систем. Моделирование основано на формировании и заполнении текстовыми записями унифицированных справочников и матриц связанностей требований и компонент объекта. Уже на начальных стадиях системной разработки подход позволяет упорядочить представление структуры объектов и быстро унифицированно формировать и оценивать облик рассматриваемого объекта, в рамках построения цифрового представления объекта согласно общих подходов интеллектуальной цифровой экономики. Так на рисунке 15 представлен подход MBSE: до математического моделирования, алгоритмов и автоматизации – структурирование онтологических представлений системы в форме архитектурных моделей $A(a) = \{E, E(a), [E, E(a)]\}$ [10].

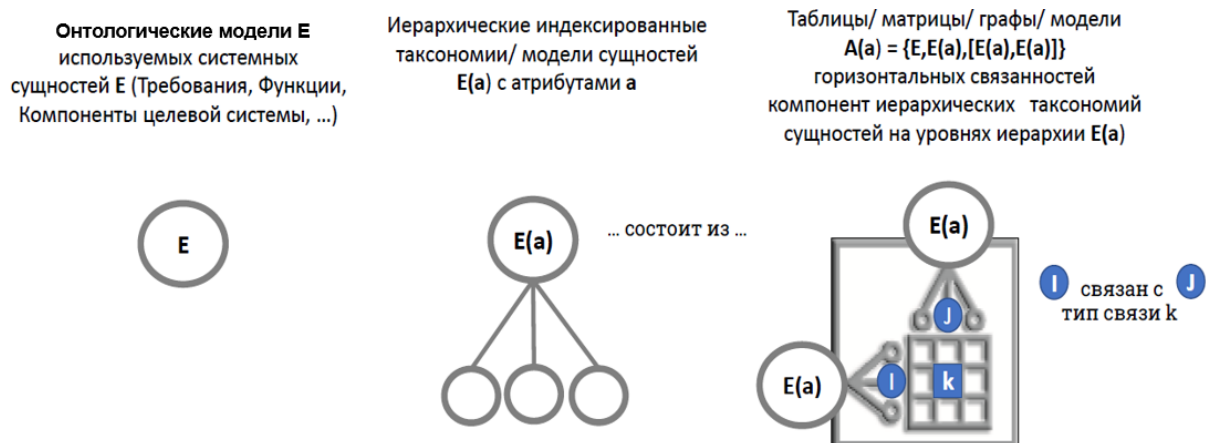


Рис. 15 – Политики и дорожные карты моделирования и применения моделей в MBSE инжиниринге.

В 1970-х годах известный британский статистик Джордж Э. П. Бокс заявил: «Все модели неверны, но некоторые полезны». Это утверждение справедливо и сегодня, особенно по мере того, как организации переходят к MBSE. MBSE требует принятия предположений и сбора данных. Аналитики используют эти предположения, данные и переменные отношения для разработки компьютерных моделей, жизненно важных для внедрения MBSE. Поскольку модели являются абстракцией физического мира, точность модели является функцией как разрешения (resolution) модели, так и входных данных. Учитывая, что программы разработки, вероятно, будут использовать модели с более высоким разрешением в процессе проектирования, качество входных данных становится первостепенным для правильного моделирования системы. Поэтому организации должны стремиться разрабатывать репозитории данных и моделей для обеспечения успешного применения методологии MBSE. Организациям следует быть осторожными при разработке этих репозиториях, обеспечивая курирование проверенных и подтвержденных данных и моделей. Кроме того, организации должны обеспечивать надлежащую документацию данных и моделей. Важно сохранять историю источника данных или модели, всю необходимую

информацию для документирования контекста для ее первоначального использования и списка лиц, которые создали или обновили модели.

Внедрение MBSE на предприятии должно преодолеть как технические, так и социальные препятствия. Технические препятствия возникают из-за отсутствия или раздробленности данных и моделей в организациях. Раздробленность может препятствовать пониманию и полезности существующих данных и моделей, ограничивая их использование. Кроме того, раздробленность увеличивает бюрократию и затраты, связанные с поддержанием и использованием данных и моделей, что приводит ко второму социальному препятствию. Организации должны изменить свою культуру, чтобы поощрять кросс-функциональный обмен и сотрудничество. Это особенно важно для интеграции моделей. Поэтому целесообразно рассмотрение предприятия как системы (рис. 16).

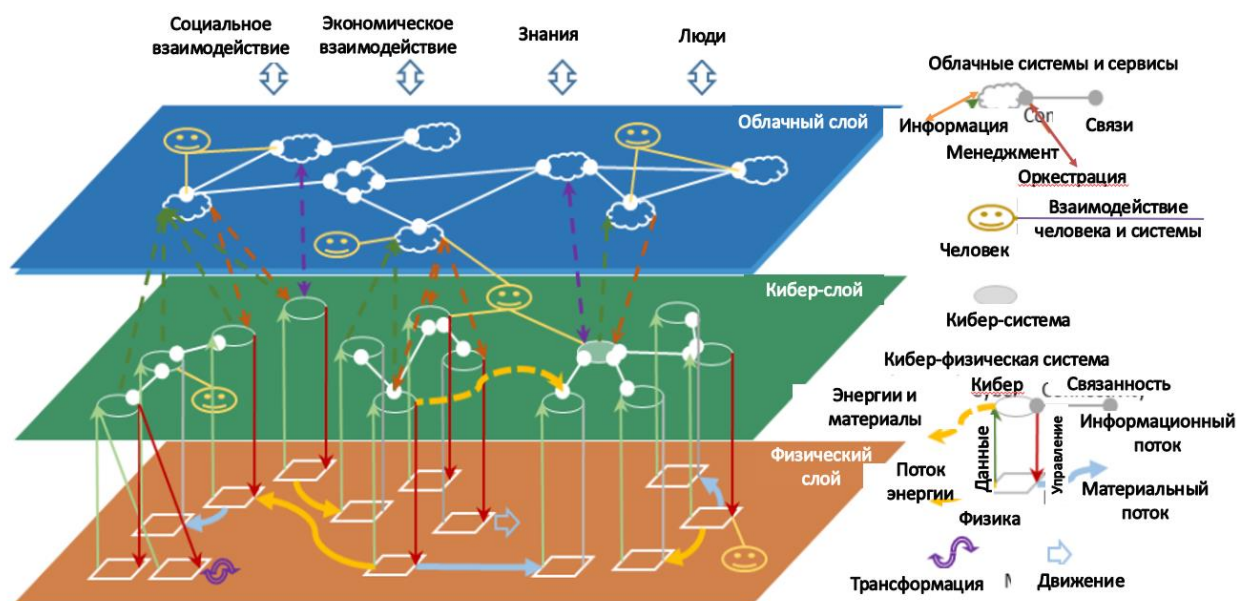


Рис. 16 – Представление предприятия как сложной технической системы.

Команды должны обмениваться форматами входных и выходных данных, потребностями и требованиями. Кроме того, инженеры и проектировщики должны перейти от предоставления только проектов к

предоставлению также и моделей, что позволит проводить системный анализ и созревание. Это требует изменения организационного мышления, помогая им увидеть ценность инженерных практик на основе моделей. Все команды в организации должны видеть эту ценность. Они должны видеть, как дополнительное время, необходимое для документирования и создания данных и моделей, в дополнение к их предоставлению, помогает организации принимать более обоснованные решения. Если все видят, как они вносят свой вклад в ценность процесса и ключевые решения, они с большей вероятностью будут следовать указаниям руководства и в конечном итоге станут сторонниками, помогающими с трансформацией организации в сторону MBSE. С учетом итерационный процесс моделирования систем в MBSE (рис. 14) схема применения MBSE на предприятии может выглядеть как на рисунке 17, который показывает как применение модельно-ориентированного подхода позволяет представить и обеспечить вертикальные и горизонтальные связи в системе.

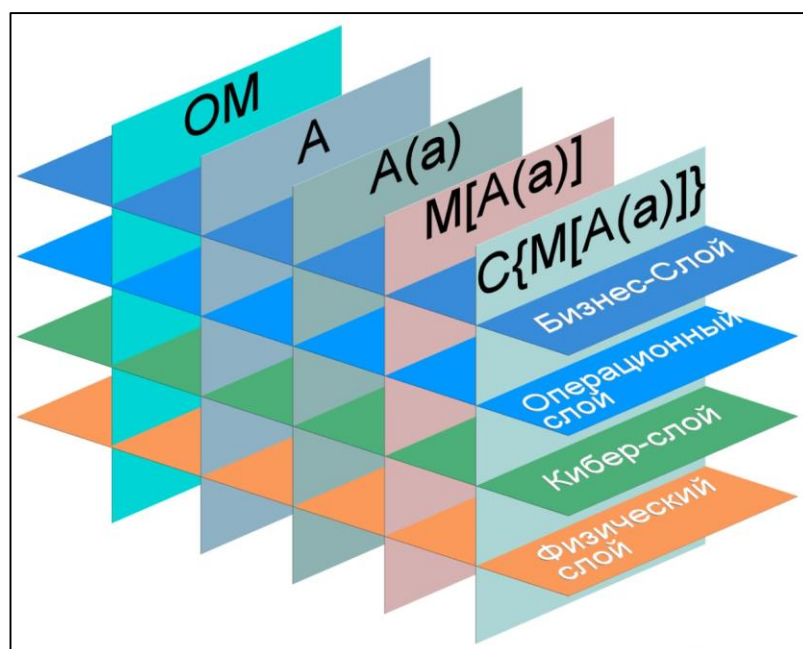


Рис. 17 – Представление сложной технической системы предприятия с учетом MBSE подхода.

Текущие инструменты MBSE должны будут развиваться для поддержки внедрения MBSE на предприятии. Им нужно будет интегрироваться с различными базами данных и различной точностью моделей. Несколько текущих инструментов обладают некоторыми из этих возможностей, но ни один из них не обладает всеми необходимыми и желаемыми возможностями для проведения анализа компромиссов и прослеживаемого процесса принятия решений.

Архитектурные фреймворки призваны помочь вам организовать модели и определить, какие представления необходимы для достаточного описания архитектуры предприятия как системы [47]. Для фреймворка характерно организовывать модели и представления с помощью «сетки», часто со столбцами, представляющими «аспекты» (aspects – различные способы, которыми вы можете охарактеризовать что-либо), и строками, представляющими «перспективы» (perspective – способы понять, как аспекты соотносятся друг с другом и с целым). Аспекты — это различные способы, которыми что-то может рассматриваться разумом в отношении торгового пространства архитектурного решения. Перспективы обеспечивают понимание всего спектра проблемного пространства и помогают раскрыть разнообразие интересов. Метод унифицированной архитектуры, проиллюстрированный в таблице 1, использует этот подход при организации моделей, которые будут разработаны.

Таблица 1. Метод унифицированной архитектуры.

	Точка зрения (аспект – aspect)			
Перспектива (perspective)	Данные	Активность (activity)	Местоположение (location)	Люди
Бизнес	Бизнес-модель сущностей	Бизнес-модель процессов	Бизнес-модель местоположения	Ролевая бизнес-модель
Логическая	Логическая модель сущностей	Логическая модель процессов	Логическая модель местоположения	Логическая модель ролей
Техническая	Техническая модель сущностей	Техническая модель процессов	Техническая модель местоположения	Техническая модель ролей

Метод унифицированной архитектуры (UAM) представляет собой типичный способ организации моделей и представлений архитектуры с использованием аспектов и перспектив.

На рисунке 18 показаны шесть шагов метода для разработки моделей, который обеспечивает структурированный путь построения правильных моделей и получения максимальной пользы от их использования. Это является общепринятым шаблоном для разработки моделей. Однако чаще всего люди начинают с шага 4 (создание модели) и сразу же начинают строить модели, не имея четкого представления о том, зачем нужны конкретные модели или какие типы данных необходимо в них вводить.

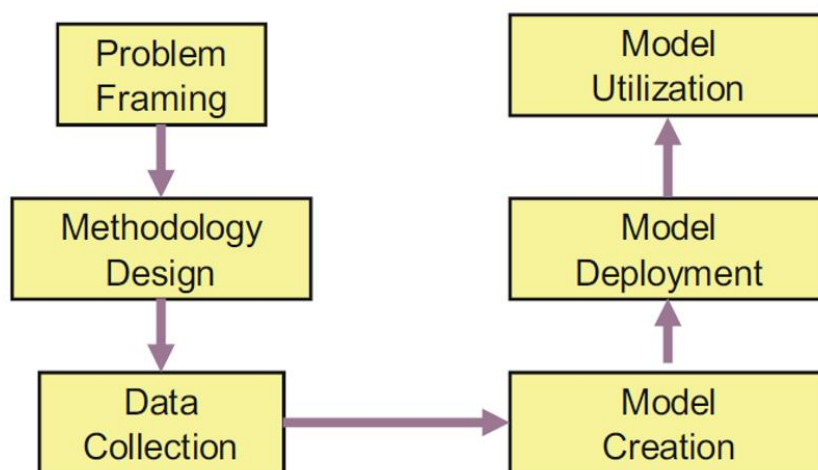


Рис. 18 – Структурированный способ построения правильных моделей и извлечения из них максимальной пользы.

Этот метод разработки модели, показанный выше, был использован в качестве основы для процесса разработки архитектуры в международном стандарте ISO 42020 по процессам архитектуры. Целью процесса разработки архитектуры является «описание или документирование архитектуры достаточно полным и правильным образом для предполагаемого использования архитектуры». Подход постановки проблемы включен в ранние этапы процесса разработки архитектуры в стандарте ISO (рис. 19).

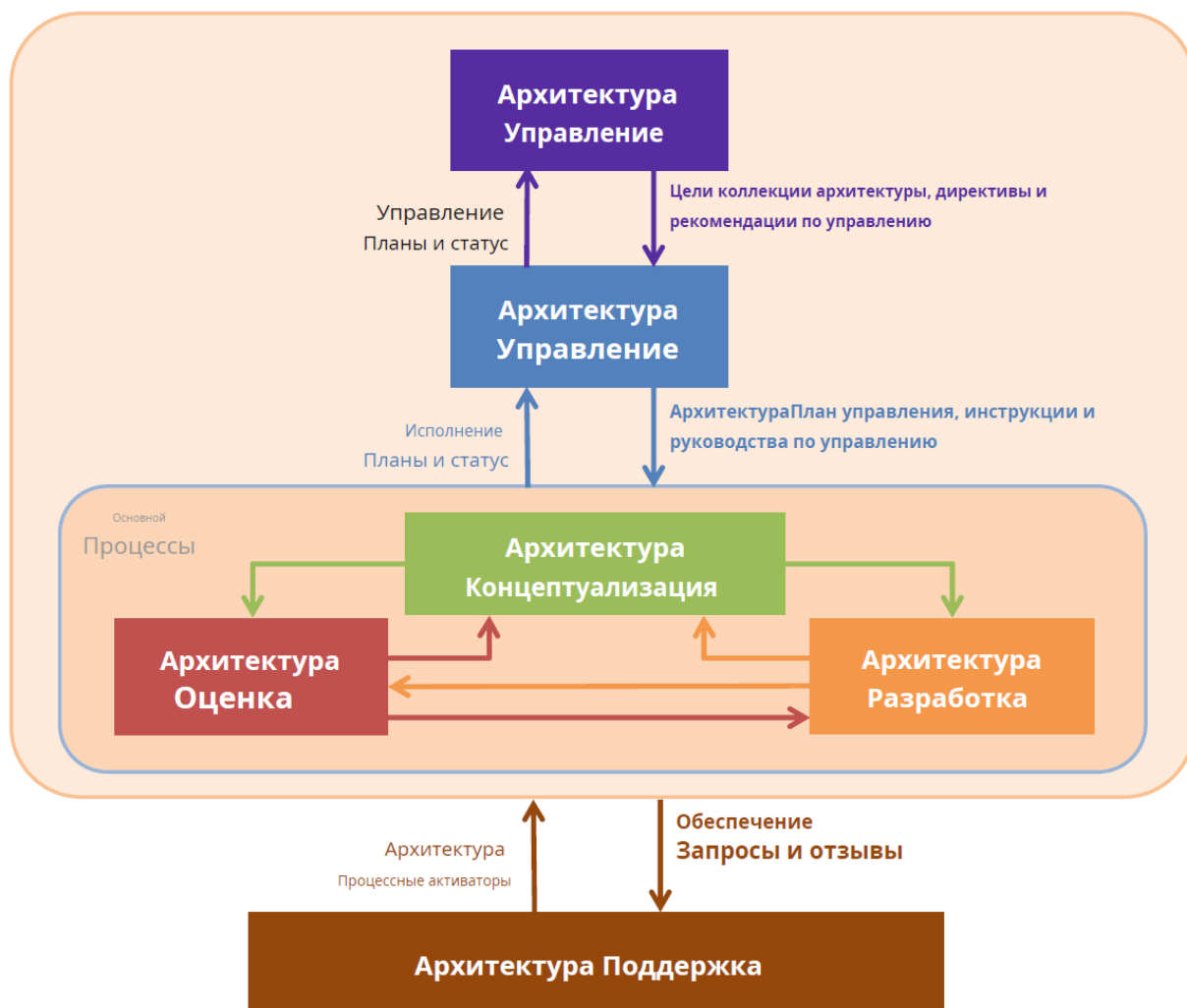


Рис. 19 – Архитектурные процессы в международном стандарте ISO/IEC/IEEE 42020

Архитектура систем на основе моделей и принятие решений

Выбор архитектуры является фундаментальной деятельностью в процессе применения MBSE для формирования концептуальной системной архитектуры. Структура системной архитектуры на основе моделей облегчает спецификацию архитектуры как процесса рассуждения, т.е. последовательного ряда действий по концептуализации и принятию решений, подкреплённых средой MBSE. Такая структура охватывает как онтологию архитектуры системы, ориентированной на заинтересованные стороны и решения, так и процесс развития архитектуры как серии шагов концептуализации через пять онтологических доменов (областей интересов): домен заинтересованных сторон, нейтральная среда решения, среда решения, интегрированная концепция и концепция операций. Такой подход смещает фокус моделирования с записи на концептуализацию, исследование, принятие решений и инновации. Хотя такой подход может изначально потребовать больших усилий, но должен обеспечить более сильное взаимодействие заинтересованных сторон, более четкое определение точек архитектурного решения, более быстрое исследование, лучшую долгосрочную жизнеспособность и повышенную надежность модели.

Выбор архитектуры для сложной системы, является задачей, полной неоднозначности. Традиционные подходы опирались на федеративную смесь неформальных, полужформальных и формальных методов. Растущие проблемы, с которыми сталкиваются системы сегодня, сделали эти «офлайн» подходы в значительной степени устаревшими. Системная инженерия на основе моделей (MBSE) [48] постепенно становится основным подходом для практики системной инженерии. Однако, в то время как традиционная системная инженерия работает над выявлением недостающих связей между подсистемами, MBSE сегодня сосредоточена на описательной записи концепций в моделях [49]. Это представление концепции необходимо для

дальнейшей обработки, анализа и представления, но это только один аспект системной инженерии. Текущая практика и исследования перегружены репрезентативными усилиями в MBSE и недооценены анализом и принятием решений. Так, инженеры-механики должны предоставлять валидные, проверенные, пригодные для сборки и обслуживания проекты деталей и компонентов, независимо от технологии проектирования, с помощью которой они проектируют и т. д. Тем не менее, в современном ландшафте цифровой инженерии [50-52] никто не представляет, что инженеры-механики, как и инженеры других специальностей, не будут использовать новейшее программное обеспечение для управления, проектирования, внедрения, тестирования и развертывания своих результатов. Системная инженерия не должна быть исключением.

MBSE может использоваться для поддержки системной архитектуры и для обеспечения того, чтобы процесс разработки и реализации проекта оставался строгим и прозрачным, т.е. понятным для всех заинтересованных сторон. Достижение хорошей системной архитектуры должно быть неотъемлемой частью любого подхода MBSE который использует модели и анализ MBSE для выбора архитектуры и не является просто вариантом MBSE для системной архитектуры. При этом много было написано об описательных аспектах MBSE, будь то каталогизация функционального потока или определение потенциальных состояний системы. Однако эта документация не обязательно поддерживает принятие архитектурных решений, если она не представляет точки принятия решений. Точка принятия решений может быть возможностью выбора решения из как минимум двух вариантов. Это архитектурные решения и позволяет оценить, как и где MBSE поддерживает принятие решений об архитектуре.

Усилия, необходимые для создания среды MBSE, и связанная с ней идеологическая трансформация подразумевают, что область применения и цель должны быть четко определены, чтобы рационализировать инвестиции в MBSE. Одной из этих целей (но ни в коем случае не единственной) является

поддержка принятия архитектурных решений, которая включает в себя следующий цикл мероприятий:

1. представление потенциальных архитектур с помощью моделей,
2. определение архитектурных решений,
3. проведение анализа в поддержку возникающих архитектурных решений,
4. принятие архитектурных решений на основе результатов анализа модели,
5. фиксация решений в модели для следующей итерации архитектуры.

Действия, выполняемые на концептуальных этапах системной инженерии, определяются как архитектурное моделирование, и его основным результатом является построение системной архитектура, определяющей целостный взгляд на всю систему [53]. Напротив, действия, выполняемые для реализации архитектуры, в частности, плановые решения с инженерными и научными знаниями, рассматриваются как проектирование, где основным результатом является проект (практический вариант реализации архитектуры): план для разработчиков по внедрению или созданию системы. Проектирование сложного компонента может представлять собой разработку архитектурных решений для этого компонента как полноценной системы, например, реактивного двигателя в самолете или коммуникационной сети, которая соединяет множество датчиков и контроллеров.

Обсуждение необходимости, актуальности и достаточности системной архитектуры, особенно в системах с интенсивным использованием программного обеспечения, продолжается, особенно с появлением подходов к итеративной и непрерывной разработке и развертыванию с коротким циклом. Ключевым аргументом остается и подтверждается многими известными неудачами [54] то, что целостная системная архитектура повышает уверенность в способности удовлетворять потребности заинтересованных сторон, разрабатывать надежную архитектуру, которая может адаптироваться

к изменениям, и сокращать объем технического дисбаланса по мере развития системы [55].

Обычно при проектировании системной архитектуры моделирование играло ключевую роль, в том числе при описании концепций [56]. При этом подход на основе моделей означает [57]:

- постановку проблемы,
- исследование альтернатив,
- моделирование системы,
- интеграцию компонентов,
- запуск системы,
- оценка эффективности и
- переоценка системы.

Архитектурные проблемы решаются в основном на ранних этапах, в которых концептуальная архитектура является основным артефактом, а физические компоненты доступны в небольшом количестве или вообще отсутствуют.

MBSE часто начинается с мозгового штурма концепции в ответ на некоторую потребность или набор потребностей и заканчивается формализованным обзором и утверждением четко определенной и выполнимой архитектуры. Между этим существует ряд концептуальных решений, однако скачок от потребностей заинтересованных сторон к четко определенной организации структурных и поведенческих элементов не происходит в одночасье. Этот ментальный «Большой каньон» просто слишком широк, чтобы перепрыгнуть через него сразу, и необходим ряд промежуточных шагов. Вопрос состоит в том, как можно мудро спланировать эти шаги, которые безопасно приведут нас на другую сторону: эта проблема проиллюстрирована на рисунке 20 [58].

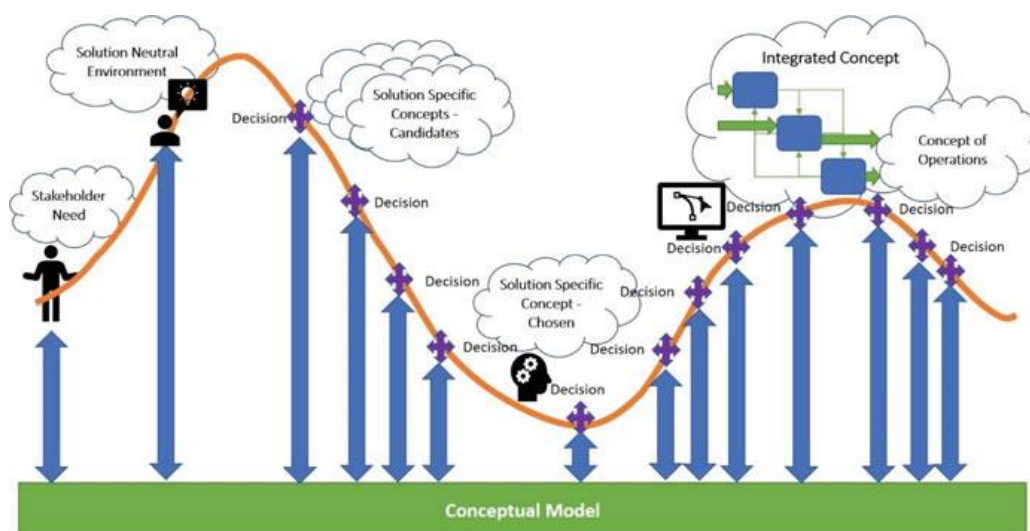


Рис. 20 – Преодоление ментального Большого каньона от потребностей к операционно-жизнеспособным решениям посредством ряда решений по архитектуре системы, используя модель в качестве базы знаний и основного механизма рассуждений.

Архитектура системы – это описание структуры и поведения системы, которые совместно предоставляют одну или несколько функций для удовлетворения потребностей заинтересованных сторон системы. MBSE опирается на формальный язык моделирования для представления и обоснования архитектуры системы, но результаты по сути такие же, как и у традиционного (не основанного на моделях) процесса: спецификация архитектуры системы, которая может служить основой для дальнейшей спецификации требований, проектирования, разработки, тестирования и эксплуатации. Эта высокоуровневая концепция MBSA проиллюстрирована на рисунке 21, на котором объекты (такие как «Заинтересованная сторона» - stakeholder и «Системный архитектор» - system architect) обозначены прямоугольниками, тогда как процесс («Построение системной архитектуры на основе моделей» - Model-Based System Architecting) обозначен овалом. Заполненный черный треугольник внутри треугольника означает, что «Потребность» является атрибутом «Заинтересованной стороны». Связь со стрелкой информирует о потреблении атрибута «Потребность» (Need)

процессом «Построение системной архитектуры на основе моделей». Связь с заполненным кружком на конце является связью агента «Системный архитектор», тогда как связь с открытым кружком на конце является связью инструмента «Построение системной архитектуры на основе моделей»).

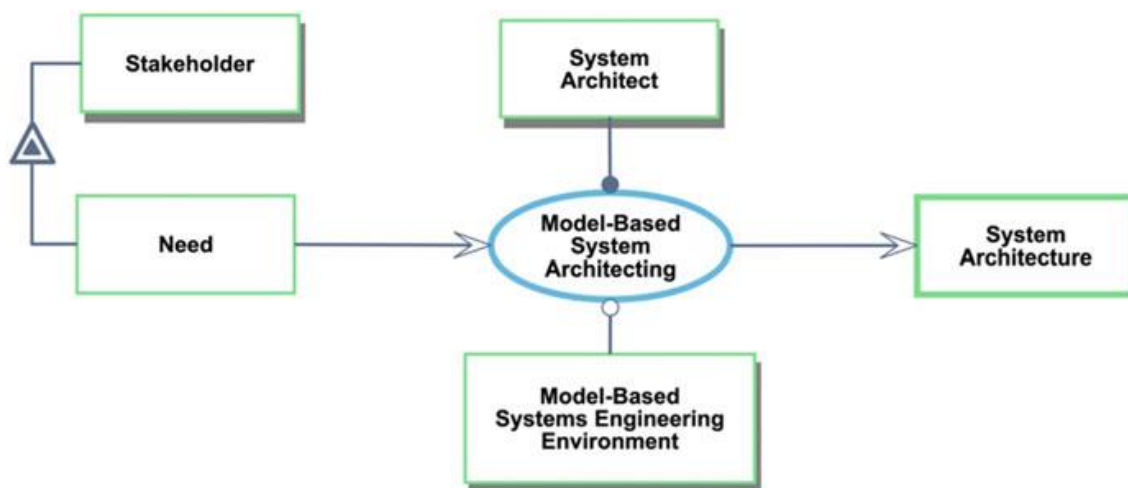


Рис. 21 – Архитектура систем на основе моделей: заинтересованная сторона (Stakeholder) демонстрирует потребность (Need). Архитектура систем на основе моделей (Model-Based System Architecting) потребляет потребность заинтересованной стороны (Need of Stakeholder). Системный архитектор (System Architect) управляет архитектурой систем на основе моделей. Архитектура систем на основе моделей требует среды системной инженерии на основе моделей. Архитектура систем на основе моделей дает архитектуру систем.

Концепция — это начальное отображение того, чего мы хотим достичь, в форму, которая будет использоваться для этого. Например, «ракета приземлится вертикально, используя стабилизаторы», «транспортное средство будет работать как на топливе, так и на электроэнергии» или «все коммуникации будут проходить через центральный узел сообщений». Концепция является частью архитектуры системы и должна быть соответствующим образом определена в рамках процесса MBSE.

Концепция отображает функцию в форму [59]. Функция системы – это процесс (действие), который обычно влияет на один или несколько операндов

(объектов, которые изменяются действием). Форма – это набор элементов, которые поддерживают эту функцию. Это аналогично трем основным частям всех языков: существительному (инструменту действия), глаголу (действию, описывающему действие) и существительному (объекту действия). Следует понимать базовую модель Концепции и связь между понятием, функцией, формой и архитектурой. Концепция — это начальное отображение того, что мы хотим достичь, в форму, которая будет использоваться для этого. Например, «ракета приземлится вертикально, используя стабилизаторы», «транспортное средство будет работать как на топливе, так и на электроэнергии» или «все коммуникации будут проходить через центральный узел сообщений». Концепция является частью архитектуры системы и должно быть соответствующим образом указано в рамках процесса MBSA.

Концепция самого высокого уровня всей архитектуры должна быть короткой фразой или предложением. Например, «Автомобиль с самостоятельным управлением осуществляет перевозку до 4 пассажиров на расстояние 500 км». В этом коротком примере мы ясно видим (а) процесс: Транспортировка, (b) форму: Автомобиль с самостоятельным управлением, (c) операнд: Пассажир (до 4). Кроме того, это утверждение включает необязательный атрибут: Расстояние (до 500 км), который может быть взят из какой-либо потребности.

Долгий путь от потребностей к решениям проходит через ряд шагов и ни в коем случае не является прямой линией. Многие из этих шагов движутся вперед и назад в том, что можно представить как танец танго. Для этой серии шагов было предложено много ментальных моделей, наиболее примечательны модель V и другие примеры, которые в основном отстаивают процедуру действий. Здесь представлена общая классификация, т.е. утверждается, что в каждой точке архитектура является одной из двух когнитивных задач: концептуализация или принятие решения. Концептуализация – это описание или указание концепций, в то время как принятие решения – это выбор концепций из доступного пула кандидатов. После принятия решения решение

становится частью решения. Концептуализация и принятие решения совместно именуются рассуждения. Каждый шаг проектирования – это шаг рассуждения, а MBSE – это серия шагов рассуждения (рис. 22) Оба типа рассуждений – концептуализация и принятие решений – могут выиграть от подхода, основанного на моделях.

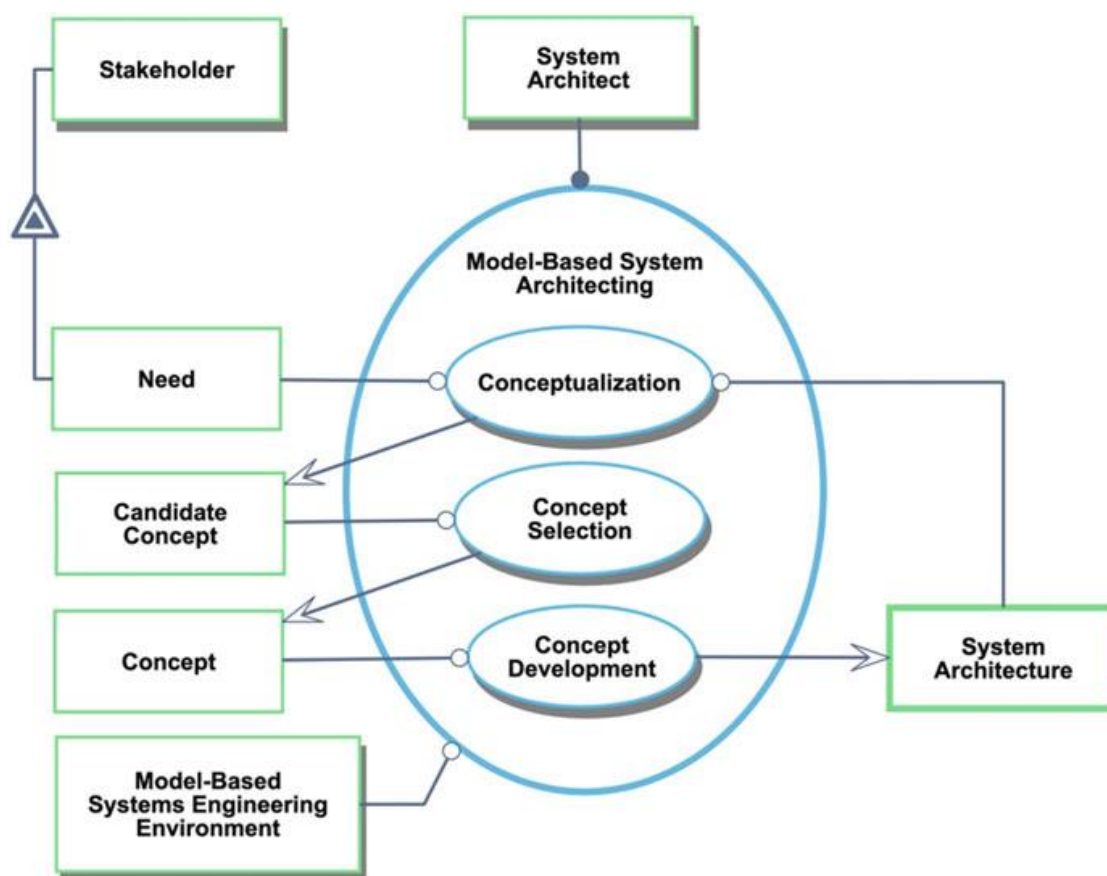


Рис. 22 – Архитектура систем на основе моделей более подробно.

Онтология – это формальный словарь концепций предметной области. Крайне важно принять и формализовать онтологию для последовательного обсуждения представления концепций, особенно для архитектур систем. Рекомендуема к использованию онтология проиллюстрирована на рисунке 23. Эта онтология лежит в основе фреймворка, который вводит основные записи в представление концепции архитектуры системы и предлагает способы кодирования этих записей, предпочтительно в среде моделирования. Онтология состоит из пяти доменов: заинтересованные стороны (D1), среда,

нейтральная к решению (D2), среда, специфичная к решению (D3), интегрированная концепция (D4) и концепция операций (D5). Домен концепта — это подмножество онтологии, которое фокусируется на определенном аспекте архитектуры и имеет отображение на другие. Домены различаются по цвету. Здесь перечислено 28 записей в этих доменах на основе фреймворка представления концепций, представленного в [59]. На эти 28 записей ссылаются, используя {EXX}, например, {E15}, ссылаясь на конкретную форму.

Domain 1 (D1) Stakeholders					
1	Stakeholder				
2	Need				
Domain 2 (D2) Solution-Neutral Environment		Domain 3 (D3) Solution-Specific Environment		Domain 4 (D4) Integrated Concept	
3	Solution-neutral operand (SNO)	8	Solution-specific operand (SSO)	17	Internal Operands (IO)
4	SNO value attribute	9	SSO value attribute	18	IO value attribute
5	SNO other attribute	10	SSO other attribute	19	IO other attribute
6	Solution-neutral process (SNP)	11	Solution-specific process (SSP)	20	Internal Processes (IP)
7	SNP attribute	12	SSP attribute	21	IP attribute
		13	Generic Form	22	Internal Elements of Form (IEoF)
		14	Generic Form attribute	23	IEoF attribute
		15	Specific Form	24	Structure
		16	Specific Form attribute	25	Interactions
				Domain 5 (D5) Concept of Operations	
				26	Concept of Operations
				27	Operator
				28	Context

Рис. 23 – Онтология архитектуры системы и структура представления концепций.

Первые три домена, D1–D3, представляют собой простейшую формулировку концепции. Четвертый и пятый домены лежат ниже по иерархии, отражая скрытое завершение, т.е. расширение D1–D3 до тех пор, пока все еще уместно продолжать детализацию архитектуры. Точное время

завершения зависит от типов решений и контекстов [59]. D1–D3 отличаются от D4–D5 абстрактным и конкретным уровнями обсуждения. Системный архитектор должен уметь переключаться с абстрактного обсуждения (D1, D2 и D3) на более конкретный уровень детализации (охваченный D4 и D5). Более того, итеративное проектирование систем означает, что D5 может влиять на D1 в цикле пересмотра, расхождения и схождения.

Домен заинтересованных сторон (D1) охватывает заинтересованных лиц и их потребности. Заинтересованная сторона {E01} — это «любая группа или лицо, которые могут повлиять или на которых влияет достижение целей системы». Другими словами, многие группы и отдельные лица могут быть заинтересованными сторонами в самом широком смысле, в зависимости от контекста. Это подчеркивает важность широкого операционного контекста, в котором система должна работать. Рассмотрим европейский «Зеленый курс» [61]: согласно данному ими определению, все люди на Земле являются заинтересованными сторонами «Зеленого курса».

Потребности заинтересованных сторон {E02} определяются как ответы на вопрос «Какие проблемы мы пытаемся решить?». Потребности должны быть ориентированы на проблему, а не на решение. Потребности часто определяются до того, как вмешивается системный архитектор. Потребности часто представляют собой нечеткие, двусмысленные заявления заинтересованных сторон. Эта нечеткость бросает вызов системным архитекторам, чтобы четко сформулировать суть потребности, например, какова ожидаемая возможность, или ожидаемый результат, или ожидаемое изменение текущего состояния. Особая важность потребностей заинтересованных сторон заключается в том, что они используются для формулирования функциональных требований в среде, нейтральной по отношению к решениям.

Потребности заинтересованных сторон могут исходить из различных источников. Первыми из них являются сами заинтересованные стороны: задача системного архитектора — организовать обсуждение с

заинтересованными сторонами таким образом, чтобы это помогло сформулировать эти потребности. Другими потенциальными источниками потребностей являются варианты использования, ограничения, требования, которые могут исходить из обширного обзора литературы. Целью системного архитектора является формулирование функционального намерения в каждой потребности заинтересованных сторон. Сначала потребности связываются с формулировкой проблемы, выражаются в нейтральной по отношению к решению среде и реализуются в процессе. Область заинтересованных сторон (D1) и потребность определяется как атрибут заинтересованных сторон.

Необходимость в среде, нейтральной к решению, является основополагающим принципом проектирования. Среда, нейтральная к решению (D2), облегчает выявление функциональных требований, которые должны быть свободны от любых предубеждений в сторону перспективных подходов к решению, конкретных технических дисциплин или стратегий реализации [62]. Поэтому системный архитектор указывает основную информацию о процессе, нейтральном к решению, до разработки концепции решения. Функциональное намерение (например, «перевозка пассажиров», «передача денег» или «воспроизведение музыки») должно быть сформулировано до того, как будут установлены возможные альтернативные решения.

Операнд, нейтральный к решению (SNO) {E03}, представляет собой интересующий объект, который подвергнется некоторой трансформации с помощью процесса, нейтрального к решению (SNP) {E06}. Процесс, нейтральный к решению, демонстрирует динамическую природу функции: он отражает действие. SNP и SNO должны быть абстрактными, чтобы возникало множество альтернатив и происходил процесс принятия обоснованных решений. Записи SNE могут иметь атрибуты, которые уместно начать разрабатывать на этом этапе.

Итак, системная архитектура, как и гражданская архитектура, является одновременно и наукой, и искусством. Эта область должна отвечать на

следующие вопросы, включая явное указание того, что лежит за пределами границ MBSE [63]:

1. Какие функции должны, должны, не должны и не должны быть охвачены?
2. Может ли интроспекция интересующих функций привести к переформулировке проблемы? Если да, то насколько общей должна быть модель?
3. Какие компоненты должны и не должны быть охвачены?
4. Как мы определяем, является ли кто-то заинтересованной стороной и следует ли их включать?
5. Как мы оцениваем синергию или конфликты в данной архитектуре?
6. Какие идеи получены в процессе проектирования помимо контура выбранной архитектуры, и как мы сохраняем эти идеи, чтобы в дальнейшем информировать процесс проектирования?
7. На каком уровне детализации достаточно разложить интересующую систему по отношению к контексту и конкретным потребностям заинтересованных сторон?

MBSE позволяет записывать ответы на эти вопросы в архитектурной модели и в контексте нашей структуры представления концепций, тем самым расширяя и расширяя когнитивный процесс, выполняемый системным архитектором для рассмотрения и ответа на эти вопросы. Действительно, так же, как картина является артефактом художественного процесса, модель может записывать возникающие предложения, которые мы включаем в архитектуру системы, такие как элегантность, расширение прав и возможностей, целостность и вдохновение, все это субъективные восприятия, которые, как мы надеемся, испытают заинтересованные стороны, когда им будет представлена выбранная системная архитектура.

Важным моментом является определение ценности процесса на основе модели для принятия архитектурных решений. Архитектурные решения — это те этапы рассуждения, которые влияют на направление развития архитектуры.

Решения принимаются на протяжении всего процесса, и некоторые из них основаны на модели. Обычно утверждается, что модели способствуют более легкому повторному использованию дизайна, оценке большего количества вариантов и автоматизации исследования пространства дизайна. Представляет интерес более глубокий вопрос о том, как можно ожидать изменения процесса принятия решений. Простое наличие моделей не изменило процесс принятия решений в целом и важно определить, что такого в возможностях MBSE, что заставило бы нас поверить, что процесс выбора среди потенциальных архитектур отличается и лучше.

Разработчики часто принимают пошаговые решения по крупным программам, прежде чем появятся подробные таблицы сравнения решений. Первоначальная видимая часть разработки проекта системы часто неоднозначна, и между концепцией-как-набросок-на-салфетке и подробными спецификациями лежит пропасть, которую нужно преодолеть. Нам нужны более строгие мосты, чем традиционные — иллюстрации в виде слайд-шоу и чрезмерно сложные черновые чертежи в виде блок-схем и т.д. Способность визуализировать информацию, поддерживающую принятие решений, отошла на второй план, пока сообщество MBSE сосредоточилось на среде моделирования и пользовательском опыте специалиста по моделированию или аналитика. Менеджеры по продуктам и инженеры были несколько забыты и утратили способность смотреть на модель, распознавать дилемму, понимать варианты и принимать или, по крайней мере, советовать решение.

Итак, решения — это выбор, который человек делает относительно чего-либо после рассмотрения нескольких возможностей. Это определение подчеркивает: а) каждое решение возникает из нескольких альтернатив, и б) выбор должен быть сделан после рассуждения и рассмотрения. Решение — это результат процесса принятия решения [64]. Архитектурные решения — это важные и критически важные решения, которые оказывают существенное влияние на концепцию, т.е. существенное преобразование структуры и поведения системы [65]. Наши когнитивные и умственные способности, а

также субъективные предубеждения могут сделать самые важные решения неотличимыми от менее важных. Подход на основе модели помогает расположить заинтересованных лиц в одной среде и гарантировать, что приоритеты, воздействия и последствия понятны всем, стандартизированы и объективны как часть процесса принятия решений. Принятие решений – это процесс достижения решения. Обычно он состоит из трех фаз, применяемых итерационно: Определение проблемы принятия решения, Принятие решения и Выполнение решения. Также иногда говорят о таких итерационных фазах Размышление, Варианты решения, Принятия решения (Intelligent – Design – Choice) [66, 67].

Основная роль системного архитектора – принятие решений, а принятие решений – суть архитектуры, однако все больше архитектурных решений принимается в группах, и роль архитектора становится одной из функций содействия, модерирования, информирования и регистрации архитектурных решений [68]. Это понятие подчеркивает важность подходящей платформы, которая будет помогать системному архитектору на протяжении всего процесса принятия архитектурных решений. Возможности поддержки принятия решений включают управление информацией, формулирование, рекомендации, выбор, выполнение и обучение.

Рассмотрение нескольких альтернатив в общем случае естественна для людей. От самой тривиальной работы до самых насущных и фундаментальных проблем нашей жизни всегда есть как минимум два варианта, и даже когда есть один видимый вариант, есть также вариант по умолчанию «ничего не делать» (DN). Когда мы рассматриваем медицинское лечение, мы определяем альтернативные клиники, врачей, медицинские подходы, доступные дни и часы, медицинское покрытие и риск ухудшения нашего состояния здоровья. Сложная системная архитектура ничем не отличается: когда мы проектируем новый самолет, мы оцениваем желаемую вместимость, расход топлива, дальность полета, возможности автоматизации пилотирования, ситуационную

осведомленность и т. д. Альтернативы возникают из ключевых атрибутов, соответствующих значений и возможных комбинаций.

Анализ решений является научной основой принятия решений. Он коренится как в точных, так и в социальных науках, порождая две парадигмы DN: аналитическую и поведенческую. Аналитические, ориентированные на модели подходы возникли из классических вероятностных и теоретико-полезных подходов и сосредоточены на рациональном выборе [69]. Поведенческие теории принятия решений рассматривают принятие решений как ненормативный, ориентированный на человека процесс со всеми вытекающими из него проблемами [70]. Согласно поведенческой школе, человек или группа людей принимают решения в условиях различных ограничений, неопределенности и предвзятости. Поведенческий подход имеет дело с эвристикой, рациональностью и рационализацией, аналитическим параличом и множеством других аспектов и явлений человеческого познания. Управленческие аспекты, такие как отслеживание и обеспечение решений, считаются в основном поведенческими.

Два основных руководства по системной инженерии (INCOSE и NASA [14]) рассматривают принятие решений как инженерный процесс, касающийся как программных, так и архитектурных аспектов. Программные решения принимаются на этапе принятия решений, чтобы упростить управление проектами и рисками. Архитектурные решения касаются таких аспектов, как функциональность, дизайн, технология и выбор поставщика.

Рассуждения об альтернативах являются фундаментальной частью наших действий. Необходимо поддерживать процесс рассуждений посредством записи и визуализации альтернатив, композиций альтернатив и сравнений альтернатив. Подход на основе моделей расширяет возможности рассуждений, сводя все альтернативы к общему формальному языку и представлению. Представления альтернатив не всегда просты, а необходимость формулировать их в соответствии с синтаксисом и семантикой языка моделирования заставляет альтернативные концепции и их атрибуты

появляться через интерактивный когнитивно-вычислительный процесс, а не через детерминированный механистический процесс преобразования идей в модели.

Архитектурные решения представляют собой шлюз между проблемно-ориентированной формулировкой точки решения архитектуры и спецификацией и разработкой архитектуры, ориентированной на решение. Хотя решения принимаются на основе соображений затрат и выгод, мы помним, что архитектурный морфизм также учитывает набор доступных операций в данной модели архитектуры. Однако размер этого набора операций совершенно произволен. А именно, мы можем выполнить любое количество операций, которое, по нашему мнению, достаточно для того, чтобы удостовериться в способности архитектуры предоставлять затраты и выгоды, которые мы оцениваем для нее. В мире неполной информации, постоянных изменений и ограниченных ресурсов мы должны делать вызов во многих случаях и часто в конечном итоге пересматривать и пересматривать наши прежние решения.

Таким образом было показано, что системная архитектура по сути является процессом рассуждения, который состоит из этапов концептуализации и принятия решений. MBSE облегчает, но также и изменяет процесс архитектуры. Стоит сделать шаг назад, чтобы суммировать, как MBSE отличается и развивается от устаревшей, «офлайновой» системной архитектуры. Процесс архитектуры (будь то устаревшая или основанная на модели) является источником открытий и прозрения. Он решает потребность в понимании посредством достаточной спецификации и полноты охвата на заданном уровне абстракции анализа. Эти драйверы позволяют нам судить, когда архитектурное моделирование завершено.

Была представлена модельно-ориентированная структура представления концепций, которая формализует процесс концептуализации для создания архитектуры системы. Структура предлагает путь к решению, учитывающему все основные проблемы в архитектуре системы.

Действительно, это не линейный, а итеративный процесс, изобилующий пересмотрами, расхождениями и сближениями. Структура учитывает как текущие концептуализации, так и принятие решений, когда необходимо принять критические решения.

В заключение приведем три ключевых принципа MBSE:

1. MBSE – это итеративный процесс рассуждений, в котором модель записывает и информирует развивающуюся концептуальную архитектуру.

2. MBSE способствует расхождению перед сходимостью: позволяя вариантам появляться из нейтральной по отношению к решению среды и сходясь к решению после рассмотрения нескольких подходов, специфичных для решения.

3. Проекты MBSE могут фокусироваться на соответствующих концептуализациях и не обязательно следовать подходу от А до Я – объем смоделированной информации должен быть достаточным для принятия решения.

Процесс проектирования «офлайн» основан на изолированном анализе и обсуждении, и мы обнаружили, что требуется значительно больше усилий для создания модели для ответа на архитектурные вопросы. Часть этих усилий обусловлена необходимостью перевода теоретических концепций принятия решений на язык концептуального моделирования. Однако после того, как модель разработана и может служить основой, результирующее исследование становится быстрее с каждой итерацией. Подход на основе модели может обеспечить возврат инвестиций благодаря возможности повторного использования и развития активов модели в ходе нескольких итераций. Однако мы предупреждаем, что процесс проектирования не следует рассматривать как процедурный: наша структура — это не просто контрольный список. Учитывая, что ни одна архитектурная модель не сможет охватить подробную информацию о проекте, мы обнаружили, что процесс моделирования и осмысления важен для создания общего понимания среди членов команды архитекторов. Таким образом, процесс остается как

когнитивным, так и вычислительным, и два потока дополняют друг друга благодаря общему артефакту, который представляет собой модель, которая поддается как человеческому мышлению, так и машинной обработке.

Основные преимущества MBSE при разработке архитектуры следующие:

Обеспечение коммуникации с/между заинтересованными сторонами. Этот процесс является неотъемлемой частью процесса принятия решений. Привлечение заинтересованных сторон, вовлечение их в обсуждения, поддержка их приверженности миссии системы и информирование о ходе разработки имеют решающее значение для разработки решения, соответствующего потребностям заинтересованных сторон. Общий язык для обсуждения концепций, проблем, вариантов и решений с использованием эталонной модели играет важную роль в содействии, формализации и документировании обсуждения.

Повторное использование знаний и составляющих системной архитектуры. Растущая обеспокоенность в предприятиях, работающих над развивающимися системами в гибком мире, повторное использование существующих знаний о предметной области и дизайне существующих конструкций, становится критическим аспектом MBSE. Повторное использование знаний, архитектуры и артефактов дизайна имеет два основных влияния на процесс принятия решений. Опора на существующие решения и разработки снижает неопределенность, неоднозначность и риск разработки (график, бюджет, качество и т. д.). Кроме того, сокращение исследуемого пространства решений путем повторного использования существующих конструкций компонентов является подходом к преодолению проблем, связанных с комбинаторным количеством решений. Решения по архитектуре системы, основанные на формальной модели, которая включает повторно используемый контент предметной области и определенные составляющие, имеют более высокую степень уверенности и могут с большей вероятностью

преодолеть возникновение новых ограничений, проблем или реализации риска.

Последовательное отображение проблемного пространства. MBSE облегчает концептуальное отображение проблемного пространства в доменно-агностической манере. После этого можно указать потребности проблемного домена, ключевые метрики оценки и показатели производительности, а также составить архитектурное пространство решений, которое может содержать по крайней мере одно возможное решение проблемы. Развитие понимания проблемы, частично параллельно с принятием решений, помогает переформировать и воссоздать проблемный домен. Мы считаем, что эта последовательность в отображении проблемного пространства помогает нам распознавать закономерности в дизайнах.

Постепенная трансформация формулировок проблем в архитектуре. MBSE обеспечивает плавный переход от проблемной области к области решений, позволяя прослеживать отображение кандидатов для выбранных решений проблем. В общем случае можно было бы описать проблемную область множеством менее формальных способов, а решение – множеством подходов, подход на основе модели формализует переход и прослеживаемость трансформацию аспектов области решений к аспектам проблемной области.

Последовательное сравнение и эволюция альтернативных архитектур. Сравнение и рассуждения об архитектурах в рамках моделирования являются сложной задачей. MBSE позволяет это сделать, сопоставляя концептуальные архитектурные решения с количественными показателями полезности для заинтересованных сторон, с одной стороны, и с количественными показателями обоснованности проекта, с другой. Можно построить концептуальную архитектуру, которая помогает заинтересованным сторонам во многих отношениях, но отсутствие проверки архитектуры может быть плохим для бизнеса. MBSE способствует плавному переходу между методами моделирования, оценки, верификации и валидации, с

концептуальной моделью в качестве фокусной точки. Альтернативный подход перефразирует проблему принятия решений в математических терминах анализа решений, которые не полагаются на модель. Здесь есть два риска: (а) полный объем архитектуры не будет достаточно понят при разработке выбранного решения, и (б) проверка архитектуры решения путем прослеживания до «чисел» будет невозможна.

Документирование как процесса разработки архитектуры, так и его результата – системной архитектуры. Документируя процесс принятия решений в модели, создается отображение всего процесса, а не результата. Не следует гадать, почему была выбрана конкретная архитектура. MBSE будет намного сильнее в захвате функционального намерения. Ориентированная на принятие решений модель может обосновать решение на основе соображений, которые питали процесс принятия решений, приведший к этому конкретному решению. Хотя это не является распространенной практикой MBSE, в рамках MBSE возможно, достижимо и желательно гарантировать, что процесс надлежащим образом документирован, и что прослеживаемость решений к вариантам, проблемам, потребностям и заинтересованным сторонам становится ясной.

Стремление к достаточности. Обеспечение достаточности модели имеет особую важность, связанную с осознанием того, что смоделировали достаточно, чтобы принять архитектурное решение. Рассуждали так, применяя закон Миллера: использовать около 7 элементов в каждом наборе в качестве порога достаточности. Мы можем измерить соответствие различных наборов объектов и процессов этому критерию и определить уверенность в модели и готовность полагаться на нее.

Заключение

Поскольку MBSE продолжает расширять свое влияние на более поздние фазы жизненного цикла систем, становится очевидным, что MBSE может многое выиграть, используя концепции из других дисциплин, таких как анализ решений, цифровая инженерия, искусственный интеллект, машинное обучение, социальные сети и т.д. Поэтому важно развивать синергию между MBSE и этими другими дисциплинами по многим измерениям, включая моделирование людей и групп, количественную оценку неопределенности, фиксацию человеческих предпочтений, рассуждения о проблемном пространстве, агрегацию информации, использование цифровых двойников для увеличения охвата жизненного цикла системы, использование краудсорсинга для облегчения доступа к экспертным знаниям и использование сторителлинга для увеличения участия заинтересованных сторон в совместном планировании и принятии решений [74-81]. При этом системная инженерия переживает долгожданную трансформацию, подпитываемую достижениями в области системной инженерии на основе моделей (MBSE)? появлением Industry 4.0 и Интернета вещей (IoT). Поскольку MBSE продолжает расширять свое влияние на более поздние фазы жизненного цикла систем, становится очевидным, что MBSE может использовать концепции из других дисциплин. Использование синергии системной инженерии с другими дисциплинами лежит в основе трансдисциплинарной системной инженерии, которая стремится использовать конвергенцию инженерии с другими дисциплинами для решения проблем социотехнических систем, которые кажутся неразрешимыми, если рассматривать их исключительно с инженерной точки зрения.

Анализ решений предлагает строгий подход к ранжированию и оценке альтернативных решений. Он успешно применяется в различных отраслях, например, в фармацевтике, нефтегазовой промышленности, производстве и на крупных предприятиях и используется для:

- определения предпочтений, ценностей и уровней толерантности к риску,
- количественной оценки неопределенностей и предоставления методов представления и обновления убеждений в свете новой информации с использованием соответствующих методов для формулирования решения,
- методов исследования новых и осуществимых альтернатив,
- методов для формулирования решения,
- метода включения человеческого фактора и
- методов анализа ситуации с использованием когнитивной логики.

Анализ решений также предоставляет средства для устранения когнитивных предубеждений при принятии решений [82, 83].

Цифровая инженерия (DE) – это интегрированный цифровой подход, который использует авторитетные источники системных данных и моделей в континууме по всем дисциплинам для поддержки жизненного цикла деятельности от концепции до утилизации. Цифровой двойник и цифровой поток – две важные концепции в DE, которые могут позволить MBSE охватить более поздние этапы жизненного цикла системы, такие как верификация и валидационные испытания. Концепция цифровых двойников уже используется в MBSE [16, 18, 84]. Так на рисунке 24 представлены этапы построения цифровой модели по методологии MBSE.

Онтологическое описание предметной области (термины, определения, ...)	Архитектурные иерархические модели объекта	Архитектурные иерархические параметризованные модели объекта	Матрицы связанностей сущностей архитектурных параметризованных моделей	Математические модели Матрицы связанностей математических моделей	Компьютерные модели	Цифровые модели	Интеллектуальные модели
--	--	--	--	--	---------------------	-----------------	-------------------------

Рис. 24 – Этапы построения цифровой модели системы согласно MBSE.

Первые 3 этапа на рисунке 23 представляют собой так называемое «Обликовое» проектирование системы [8], включающее построение онтологии метамодели, онтологии предметной области, справочников

сущностей архитектурных моделей: требований, компонентов, функций, процессов. Построение таблицы связанностей сущностей архитектурных моделей представляют обычно большие трудности и позволяют проводить балансировку параметров требований и ограничений. Цифровое и компьютерное проектирование включают сборки, цифровые платформы и полигоны моделирования и виртуальных испытаний. В общем случае использование MBSE в цифровой трансформации предприятий [9] позволяет следующее:

- Весь состав используемых моделей возможно системно упорядочить и взаимоувязать между собой.
- Модели начинают рассматриваться совместно с физическими объектами как полноценные компоненты гибридного представления физической и виртуальной Вселенной, с двусторонними связями.
- Такой подход позволяет системно и упорядоченно представлять целевые объекты в разных взаимодополняющих и взаимопроникающих слоях моделирования.
- Создание такого описания системы позволяет оперативно встраивать систему в другие системы, достраивать, развивать цифровое представление предприятия как систему систем.

Искусственный интеллект/машинное обучение (ИИ/МО) представляет собой значительный прогресс в компьютерной науке и интеллектуальной обработке данных, который быстро преобразует несколько отраслей. ИИ обычно относится к процессам и алгоритмам, которые имитируют человека, имитируя когнитивные функции, такие как восприятие, обучение и решение проблем. МО и глубокое обучение (DL) являются подмножествами ИИ. ИИ/МО имеют потенциал для расширения MBSE для моделирования сложных систем с частично известным пространством состояний, работающих в неопределенных, динамических средах.

Литература

1. Кондратьев В.В., Тищенко Е.Б. Архитектурный инжиниринг гибридных моделей, включающих цифровые двойники и машинное обучение // Экономические стратегии. – 2023. – № 5(191). – С. 94–99.
2. Романов А.А. Прикладной системный инжиниринг. М.: ФИЗМАТЛИТ, 2015. 555 с.
3. Баденко В.Л., Кондратьев В.В., Ядыкин В.К. Модельно-ориентированная системная инженерия и интеллектуальная цифровая экономика // Сборник трудов X Международной научно-практической конференции «Интеллектуальная инженерная экономика и Индустрия 5.0» (ИНПРОМ), 25-28 апреля 2024, Санкт-Петербург. В 2 т. Т.1 / Под ред. д-ра экон. наук Родионова Д.Г., д-ра экон. наук Бабкина А.В. – СПб.: Издательство ПОЛИТЕХ-ПРЕСС, 2024. – стр. 169-173. ISBN 978-5-7422-8535-9
4. Henderson K., Salado A. Value and benefits of model-based systems engineering (MBSE): Evidence from the literature // Systems Engineering. – 2021. – Т. 24. – №. 1. – С. 51-66.
5. Badenko V.L. et al. Integration of digital twin and BIM technologies within factories of the future // Magazine of Civil Engineering. – 2021. – №. 1 (101). – С. 10114.
6. Боровков А.И., Рябов Ю.А., Гамзикова А.А. Цифровые двойники в нефтегазовом машиностроении // Деловой журнал Neftegaz. RU. – 2020. – №. 6. – С. 30-36.
7. Гаричев С.Н. и др. Модельно-ориентированный инжиниринг физико-технических, информационных и интеллектуальных систем // Труды Московского физико-технического института. – 2022. – Т. 14. – №. 2 (54). – С. 149-161.
8. Кондратьев В.В. Модельно-ориентированный системный инжиниринг 2.0. – М.: МФТИ, 2021. 102 с.

9. Баденко В.Л., Тищенко Е.Б., Ядыкин В.К. Модельно-ориентированный системный инжиниринг как инструмент цифровой трансформации промышленного предприятия // Стратегическое управление цифровой трансформацией интеллектуальной экономики и промышленности в новой реальности: монография / под ред. д-ра экон. наук, проф. А.В. Бабкина. – СПб.: ПОЛИТЕХ-ПРЕСС, 2024. стр. 603-625 ISBN 978-5-7422-8606-6. DOI 10.18720/IEP/2024.3/23
10. Баденко В.Л., Большаков Н.С., Федотов А.А., Ядыкин В.К. Цифровые двойники сложных технических систем в индустрии 4.0: базовые подходы // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Экономические науки. – 2020. – Т. 13. – №. 1. – С. 20-30.
11. Левенчук А.И. Системное мышление. М: Издательские решения. – 2018. – 398 с.
12. Кондратьев В., Лоренц В. Даешь инжиниринг! Методология организации проектного бизнеса. М: Эксмо. – 2009. – 576 с.
13. Walden D. D., Roedler G. J., Forsberg K. INCOSE systems engineering handbook version 4: updating the reference for practitioners //INCOSE International Symposium. – 2015. – Т. 25. – №. 1. – С. 678-686.
14. INCOSE Systems engineering principles; M. D. Watson - Chair et al., 2022; ISBN 978-1-937076-08-5
15. Прохоров А., Лысачев М., Боровков А. Цифровой двойник. Анализ, тренды, мировой опыт. М.: АльянсПринт. – 2020. – 401 с.
16. Madni A. M., Madni C. C., Lucero S. D. Leveraging digital twin technology in model-based systems engineering //Systems. – 2019. – Т. 7. – №. 1. – С. 7.
17. Стрельцова Г.А., Исаева Г.Н. Применение MBSE-методологий при создании аэрокосмических систем // Информационно-технологический вестник. – 2023. – № 4(38). – С. 123-136.

18. Madni A.M., Sievers M. Model-based systems engineering: Motivation, current status, and research opportunities //Systems Engineering. – 2018. – Т. 21. – №. 3. – С. 172-190.
19. Yang L., Cormican K., Yu M. Ontology-based systems engineering: A state-of-the-art review //Computers in Industry. – 2019. – Т. 111. – С. 148-171.
20. Сухомлин В.А. Анализ исследований математических основ модельно-ориентированной системной инженерии // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. – 2024. – Т. 20, № 3. – С. 350-365.
21. Yonglin L., Zhi Z., Qun L. An ontological metamodeling framework for semantic simulation model engineering //Journal of Systems Engineering and Electronics. – 2020. – Т. 31. – №. 3. – С. 527-538.
22. Введение в модельно-ориентированную системную и программную инженерию (MBSSE): учебник / В.А. Сухомлин, В.Ю. Романов, Д.А. Гапанович. – Москва : Фонд «Лига интернет-медиа»; МАКС Пресс, 2024. – 672 с.
23. Sievers M. Semantics, metamodels, and ontologies //Handbook of Model-Based Systems Engineering. – 2023. – С. 15–46
24. Badenko V. et al. Method for Developing the System Architecture of Existing Industrial Objects for Digital Representation Tasks //Systems. – 2024. – Т. 12. – №. 9. – С. 355.
25. Dunbar D. et al. Driving digital engineering integration and interoperability through semantic integration of models with ontologies //Systems Engineering. – 2023. – Т. 26. – №. 4. – С. 365-378.
26. Tudorache T. et al. WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web //Semantic web. – 2013. – Т. 4. – №. 1. – С. 89-99.
27. Wardhana H., Ashari A., Sari A.K. Transformation of sysml requirement diagram into owl ontologies //International Journal of Advanced Computer Science and Applications. – 2020. – Т. 11. – №. 4. – С. 106-114

28. Sure Y., Staab S., Studer R. Ontology engineering methodology //Handbook on ontologies. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. – С. 135-152.
29. Авдеев Т. В., Бакаев М. А. Гибридная модель представления знаний для реализации вывода во фреймовой онтологии //Научный вестник Новосибирского государственного технического университета. – 2013. – №. 3. – С. 84-90.
30. Corcho O., Fernández-López M., Gómez-Pérez A. Methodologies, tools and languages for building ontologies. Where is their meeting point? //Data & knowledge engineering. – 2003. – Т. 46. – №. 1. – С. 41-64.
31. Lara J. D., Guerra E., Kienzle J. Facet-oriented modelling //ACM Transactions on Software Engineering and Methodology (TOSEM). – 2021. – Т. 30. – №. 3. – С. 1-59.
32. Пашковский Д.А., Быков А.А. Методические подходы и особенности построения фасетно-иерархической классификации рисков вертикально интегрированной компании // Газовая промышленность. 2021. № 11 (824). С. 104-117.
33. Estefan J. A. et al. Survey of model-based systems engineering (MBSE) methodologies //INCOSE MBSE Focus Group. – 2007. – Т. 25. – №. 8. – С. 1-12.
34. Estefan, J.A., Weilkiens, T. MBSE Methodologies. In: Madni, A.M., Augustine, N., Sievers, M. (eds) Handbook of Model-Based Systems Engineering. Springer, Cham. – 2023. – С. 48-81.
35. Weilkiens T. et al. Evaluating and comparing MBSE methodologies for practitioners //2016 IEEE International Symposium on Systems Engineering (ISSE). – IEEE, 2016. – С. 1-8.
36. Баденко В.Л. Модельно-ориентированный системный инжиниринг для строителей: основные понятия и принципы: учебное пособие. – СПбПУ. – 2023. – 48 с.

37. Щедровицкий Г.П. Схема мыследеятельности-системно-структурное строение, смысл и содержание //Системные исследования. Методологические проблемы. – 1987. – Т. 1986. – С. 124-146.
38. Almeida J.P.A. et al. An Analysis of the Semantic Foundation of KerML and SysML v2 //International Conference on Conceptual Modeling. – Cham : Springer Nature Switzerland, 2024. – С. 133-151.
39. Keskin B., Salman B., Koseoglu O. Architecting a BIM-Based Digital Twin Platform for Airport Asset Management: A Model-Based System Engineering with SysML Approach //Journal of Construction Engineering and Management. – 2022. – Т. 148. – №. 5. – С. 04022020.
40. Royce W.W. Managing the development of large software systems: concepts and techniques //Proceedings of the 9th international conference on Software Engineering. – 1987. – С. 328-338.
41. Boehm B.W. A spiral model of software development and enhancement //Computer. – 2002. – Т. 21. – №. 5. – С. 61-72.
42. Forsberg K., Mooz H. Application of the ‘vee’to incremental and evolutionary development //INCOSE International Symposium. – 1995. – Т. 5. – №. 1. – С. 848-855.
43. ГОСТ Р 59194-2020 Управление требованиями. Основные положения.
44. Халл Э., Джексон К., Дик Д. Инженерия требований. – М: ДМК Пресс, – 2017. – 218 с.
45. Косяков А., Свит У., Сеймур С., Бимер С. Системная инженерия. Принципы и практика. – М: ДМК Пресс, – 2014. – 624 с.
46. Badenko V. et al. Principles for Sustainable Integration of BIM and Digital Twin Technologies in Industrial Infrastructure //Sustainability. – 2024. – Т. 16. – №. 22. – С. 9885.
47. Макареня Т.А., Быстрая Ю.С. Архитектура предприятия: инструмент разрешения противоречий управления предприятием

//Государственное и муниципальное управление. Ученые записки. – 2015. – №. 4. – С. 48-52.

48. Mordecai Y., Fairbanks J.P., Crawley E.F. Category-theoretic formulation of the model-based systems architecting cognitive-computational cycle //Applied Sciences. – 2021. – Т. 11. – №. 4. – С. 1945.

49. Madni A.M., Purohit S. Economic analysis of model-based systems engineering //Systems. – 2019. – Т. 7. – №. 1. – С. 12.

50. Barykin S.E. et al. Model-Based Holistic Multidimensional View on Digital Engineering Investments Appraisal //Engineered Science. – 2024. – Т. 31. – С. 1187.

51. Lu J. et al. Design ontology supporting model-based systems engineering formalisms //IEEE Systems Journal. – 2021. – Т. 16. – №. 4. – С. 5465-5476.

52. Badenko V. et al. Model-based system engineering approach for existing industrial enterprise digital transformation //Journal of Infrastructure, Policy and Development. – 2024– Т. 8. – №. 14. – С. 7983

53. Stirgwolt B.W., Mazzuchi T.A., Sarkani S. A model-based systems engineering approach for developing modular system architectures //Journal of Engineering Design. – 2022. – Т. 33. – №. 2. – С. 95-119.

54. Terry Bahill A., Henderson S.J. Requirements development, verification, and validation exhibited in famous failures //Systems engineering. – 2005. – Т. 8. – №. 1. – С. 1-14.

55. Lane J. A., Koolmanojwong S., Boehm B. Affordable systems: balancing the capability, schedule, flexibility, and technical debt tradespace //INCOSE International Symposium. – 2013. – Т. 23. – №. 1. – С. 1385-1399.

56. MacCalman A. D., Goerger S. R. Leveraging a design of experiments methodology to enhance impacts of modeling and simulations for engineered resilient systems //The Journal of Defense Modeling and Simulation. – 2018. – Т. 15. – №. 4. – С. 383-397.

57. Wach P. et al. A systematic literature review on the mathematical underpinning of model-based systems engineering //Systems Engineering. – 2025. – Т. 28. – №. 1. – С. 134-153.
58. Menshenin Y. et al. Model-based system architecting and decision-making //Handbook of Model-Based Systems Engineering. – 2023. – С. 289–330
59. Crawley E., Cameron B., Selva D. System architecture: strategy and product development for complex systems. – Prentice Hall Press, 2015. – 452 с.
60. Menshenin Y., Crawley E. A system concept representation framework and its testing on patents, urban architectural patterns, and software patterns //Systems Engineering. – 2020. – Т. 23. – №. 4. – С. 492-515.
61. Olczyk M., Kuc-Czarnecks M. European Green Deal Index: a new composite tool for monitoring European Union’s Green Deal strategy //Journal of Cleaner Production. – 2025. – С. 145077.
62. Rauch E., Matt D. T., Dallasega P. Application of axiomatic design in manufacturing system design: a literature review //Procedia CIRP. – 2016. – Т. 53. – С. 1-7.
63. Mordecai Y., James N. K., Crawley E. F. Object-process model-based operational viewpoint specification for aerospace architectures //2020 IEEE Aerospace Conference. – IEEE, 2020. – С. 1-15.
64. Баденко В.Л. Анализ экологических рисков в ГИС на основе нечетких множеств // Информация и космос. – 2013. – №3. – С. 78-84
65. Dickerson C. E. et al. Architecture definition in complex system design using model theory //IEEE Systems Journal. – 2020. – Т. 15. – №. 2. – С. 1847-1860.
66. Malczewski J. GIS-based multicriteria decision analysis: a survey of the literature //International journal of geographical information science. – 2006. – Т. 20. – №. 7. – С. 703-726.
67. Баденко В.Л. Геоинформационные технологии для решения задач природообустройства: Учеб. пособие. Санкт-Петербург: Изд-во СПбПУ, – 2013. – 147 с.

68. Weinreich R., Groher I. The architect's role in practice: From decision maker to knowledge manager? //IEEE Software. – 2016. – Т. 33. – №. 6. – С. 63-69.
69. Saaty T.L. How to make a decision: the analytic hierarchy process //Interfaces. – 1994. – Т. 24. – №. 6. – С. 19-43.
70. Kahneman D.A Perspective on Judgment and Choice: Mapping Bounded Rationality //American Psychologist– 2003. – Т. 58. – №. 9. – С. 697-720
71. Баденко В.Л., Зотов Д.К., Федотов А.А. Построение информационных моделей существующих зданий на основе данных лазерного сканирования: учебное пособие. Санкт-Петербург: Изд-во СПбПУ, – 2019. – 74 с.
72. Баденко В.Л., Зотов Д.К., Федотов А.А. Информационные технологии для выбора места размещения объекта капитального строительства: учебное пособие. Санкт-Петербург: Изд-во СПбПУ, – 2019. – 164 с.
73. Бабкин А.В. Стратегическое управление развитием цифровой экономики на основе умных технологий. Санкт-Петербург: ПОЛИТЕХ-ПРЕСС, – 2021 – 794 с.
74. Nardelli P. H. J. Cyber-physical Systems: Theory, Methodology, and Applications. – John Wiley & Sons, 2022. – 271 с.
75. Watson M.D., Mesmer B.L., Farrington P.A. Engineering Elegant Systems: Theory of Systems Engineering. – 2020. NASA_TP_20205003644, NASA, Washington, D.C. – 278 с.
76. Месарович М., Такахара Я. Общая теория систем: математические основы. М: Мир, – 1978. – 311 с.
77. Гаричев С.Н., Горбачев Р.А., Давыденко Е.В. Джапаров Б.А., Кондратьев В.В. Модельно-ориентированный инжиниринг физико-технических, информационных и интеллектуальных систем //Труды Московского физико-технического института. – 2022. – Т. 14. – №. 2 (54). – С. 149-161.

78. Баденко В.Л., Большаков Н.С., Ядыкин В.К. Информационное моделирование как инструмент управления активами //Цифровые технологии в экономике и промышленности (ЭКОПРОМ-2019). – 2019. – С. 630-637.
79. Arefiev N., Terleev V., Badenko V. GIS-based fuzzy method for urban planning // Procedia Engineering. – 2015. – v. 117. – С. 39-44.
80. Oliver E., Mazzuchi T., Sarkani S. A resilience systemic model for assessing critical supply chain disruptions //Systems Engineering. – 2022. – Т. 25. – №. 5. – С. 510-533.
81. Henderson K., McDermott T., Salado A. MBSE adoption experiences in organizations: Lessons learned //Systems Engineering. – 2024. – Т. 27. – №. 1. – С. 214-239.
82. Milanov Y. et al. Method for clustering and identification of objects in laser scanning point clouds using dynamic logic //The International Journal of Advanced Manufacturing Technology. – 2021. – Т. 117. – №. 7. – С. 2309-2318.
83. Perlovsky L. I. Toward physics of the mind: Concepts, emotions, consciousness, and symbols //Physics of Life Reviews. – 2006. – Т. 3. – №. 1. – С. 23-55.
84. Боровков А.И., Кулемин В.Ю. Цифровой инжиниринг для создания изделий высокой степени технологической сложности на основе цифровых двойников //Актуальные проблемы защиты и безопасности: Тр. XXVII Всеросс. научн.-практ. конф. СПб. – 2024. – С. 116-123.