

Теория эволюционного моделирования

Аннотация

В работе сформулированы принципы *теории эволюционного моделирования*, предназначенной для построения моделей широкого класса *сложных, открытых, динамических, слабо детерминированных систем*. Теория совершенствует концепции моделирования, разработанные в общей теории систем, в объектно-ориентированном проектировании и в универсальном языке моделирования (UML). Проведен критический анализ подхода UML. Описаны основные понятия и методы новой теории эволюционного моделирования. В отличие от UML, где нет единой модели системы, новая теория эволюционного моделирования строит **единую виртуальную модель**. Моделирование в предлагаемой теории осуществляется на основе фиксированного *набора базовых классов*, замены итерационного проектирования модели на эволюционный и усовершенствованного диаграммного представления. **Моделью является гиперграф классов**. Приведена его структура и принципы проектирования.

Теория эволюционного моделирования позволяет решать сложные задачи, возникающие при описании поведения, планировании и организации (самоорганизации) экономической, политической, административной деятельности предприятий, корпораций и других социальных систем, в том числе государства, а также других задач: Интернета, жизнедеятельности биологических организмов, климатических систем...

Теория эволюционного моделирования формирует новые принципы дальнейшего развития вычислительных средств и *средств разработки программного обеспечения*, в том числе систем управления базами данных.

Изложенные идеи и принципы теории эволюционного моделирования были эффективно использованы в Компании «Цефей» при разработке Эталон-CASE-технологий для эволюционного проектирования моделей сложных, открытых, динамических слабо детерминированных систем и Корпоративных Систем Управления «Эталон». Реализованные проекты, использующие Эталон-решения, работают в крупнейших корпоративных структурах и органах государственного управления.

Содержание

1. ВВЕДЕНИЕ	3
2. ОБЩИЕ ПОЛОЖЕНИЯ	6
3. ЭВОЛЮЦИОННОЕ МОДЕЛИРОВАНИЕ – ПРИОРИТЕТ РАЗВИТИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ	10
4. КРИТИЧЕСКИЙ АНАЛИЗ UML	18
4.1. КРАТКОЕ ОПИСАНИЕ UML	18
4.2. ДИАГРАММЫ UML	19
4.3. ПРОБЛЕМЫ UML	20
5. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ЭВОЛЮЦИОННОГО МОДЕЛИРОВАНИЯ.	23
5.1. ПРЕДМЕТ, ПРОЦЕССЫ	23
5.2. СИСТЕМА, УПРАВЛЕНИЕ	25
5.3. МОДЕЛЬ, ЭВОЛЮЦИОННОЕ МОДЕЛИРОВАНИЕ	28
5.4. ЕДИНОЕ ИНФОРМАЦИОННОЕ И ФУНКЦИОНАЛЬНОЕ ПРОСТРАНСТВО - РЕПОЗИТОРИЙ МОДЕЛИ	34
6. МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ – ГИПЕРГРАФ КЛАССОВ	36
6.1. КЛАССЫ ГИПЕРГРАФА	39
6.2. ОТНОШЕНИЯ КЛАССОВ В ГИПЕРГРАФЕ	40
7. ЭВОЛЮЦИОННОЕ МОДЕЛИРОВАНИЕ	45
7.1. АРХИТЕКТУРА ЭВОЛЮЦИОННОЙ МОДЕЛИ - ГИПЕРГРАФА КЛАССОВ	45
7.1.1. Фундаментальный уровень	45
7.1.2. Информационно-технологический уровень	46
7.1.3. Системно-ориентированный уровень	46
7.1.4. Предметно-ориентированный уровень	48
7.2. ПРИМЕР ПРОЕКТИРОВАНИЯ КЛАССОВ ПРЕДМЕТНО-ОРИЕНТИРОВАННОГО УРОВНЯ ЭВОЛЮЦИОННОЙ МОДЕЛИ	49
8. ОСНОВНЫЕ ПРЕИМУЩЕСТВА ЭВОЛЮЦИОННОГО МОДЕЛИРОВАНИЯ	52
9. ЗАКЛЮЧЕНИЕ	57
10. ЛИТЕРАТУРА	59

Автор благодарен

С.А. Хохлову, всегда верившему в этот проект, за безусловную постоянную поддержку.

И.В. Воловичу, Н.Е. Яковлеву, за многочисленные плодотворные обсуждения и помощь в работе.

Всем, кто поддерживает данный проект, невзирая на зачастую неблагоприятные обстоятельства и активные противодействия.

*I want it all,
I want it now.
Queen*

1. Введение

Общая теория систем (Л.Берталанфи) и *кибернетика* (Н.Винер) изучают, соответственно, структуру и функционирование систем с точки зрения теории управления и теории информации, отвлекаясь от их материальных и пространственных характеристик.

Важными принципами в общей теории систем являются *редукционизм* (сведение системы к ее составным частям), *холизм* (рассмотрение системы как целого) и *иерархия* системы и ее подсистем. Теория систем основана на предположении, что существуют универсальные принципы организации, справедливые для всех систем, включая физические, химические, биологические, ментальные или социальные системы. Механистический подход приходит к универсальности, редуцируя свойства систем к их материальным составляющим. Системный подход, напротив, приходит к универсальности, игнорируя конкретный материал, из которого сделана система, и концентрируя внимание на ее абстрактной организации.

Объектно-ориентированное программирование/проектирование позволяет применять индустриальные, технологические методы коллективной работы при разработке сложных проектов и систем программного обеспечения на основе понятий *класса*, *объекта* и принципов *наследования*, *инкапсуляции*, *полиморфизма*.

Моделирование используется для решения задач исследования, анализа, планирования, управления в самых разнообразных областях деятельности. В последние десятилетия моделирование стало важной составной частью проектирования программного обеспечения, основанного на CASE-технологиях (*Computer Aided Software Engineering*) [12; 16]. Хорошо известны структурный и объектно-ориентированный подходы к моделированию, в которых используется нотация диаграмм потоков данных и «сущность-связь» для первого подхода и диаграмм вариантов использования, классов и др. для второго подхода [16]. Активно продвигаются такие средства моделирования, как Silverrun, BPwin, ERwin, Oracle Designer, Rational Rose, Microsoft.NET, Paradigm Plus, G2, ReThink и др. и такие технологии, как DataRun, RUP, AgentBuilder и т.д.

Множество противоречивых определений, суждений, понятий, энергичный поиск принципов структуризации окружающего мира, сотни тысяч созданных разрозненных программных продуктов, реализующих идентичные задачи, говорят о **кризисе**, который переживают современные теории *систем управления и моделирования*.

Особенно отчетливо приметы такого кризиса отразились в области проектирования и использования бизнес-систем управления финансово-экономической деятельностью в условиях глобализации.

Эпоха структурного подхода и модульной реализации программных продуктов таких компаний, как SAP (R/3), Oracle (Oracle Applications), Microsoft (MS Business Solution: Axapta, Navision) и др. безвозвратно проходит.

Реанимация этих **концептуально и архитектурно устаревших решений** не носит характер принципиального технологического совершенствования, разве что в части хранения, доступа и обработки данных с использованием интернет-технологий.

В настоящее время подобные решения удерживают позиции в основном за счет:

во-первых, экономико-финансовых рычагов дружественных и недружественных слияний и поглощений, политического протекционизма - *захват рынков*,

во-вторых, поддержки “независимых” консультантов, которые либо предлагают жесткий ограниченный список фаворитов, либо собирают комплексные системы из тех или иных “лучших” программных модулей различных производителей и обсуждают бесперспективную концепцию интерфейсов и платформ обмена данными - *диктаторский маркетинг и агрессивный PR*,

и, наконец, за счет того, что закостенелые гиганты осознанно окружили себя бесконечным множеством небольших появляющихся и исчезающих внедренческих фирм, которые, многократно повторяясь, пишат рядом с упомянутыми “базовыми” программными продуктами свои собственные отдельные программы – интерпретации реальных задач конечного пользователя – *технологический империализм и политико-экономическое распределение прав и ответственности*.

Однако этот далеко не полный список глобальной монополизации управления потребностями в информационно-технологическом секторе бизнес-систем не в состоянии скрыть растущие проблемы не только эффективного управления экономикой, но и адекватности отражения ее состояния.

Провозглашенная финансово-экономическая прозрачность и эффективность на деле оборачивается неожиданными и непредсказуемыми для большинства участников и наблюдателей кризисами и банкротствами, ряд причин которых лежит как в принципах моделирования предметной области – экономики, так и в ограниченности технологических инструментов их воплощения.

В настоящей работе приводятся основные понятия и методы **новой** теории эволюционного моделирования. В отличие от UML, где нет единой модели системы, новая теория эволюционного моделирования строит **единую модель**.

Моделирование в предлагаемой теории осуществляется на основе фиксированного набора базовых классов, замены итерационного проектирования на эволюционное и усовершенствованного диаграммного представления.

Базовый набор классов был открыт при разработке бизнес-систем управления деятельностью предприятий и крупнейших холдингов, но на самом деле он имеет универсальное значение и применим для описания гораздо более широкого круга систем.

Теория эволюционного моделирования позволяет решать сложные задачи, возникающие при описании поведения, планировании и организации (самоорганизации) экономической, политической, административной деятельности предприятий, корпораций и других социальных систем, в том числе государства, а также других задач: Интернета, жизнедеятельности биологических организмов, климатических систем...

2. Общие положения

В настоящей работе формулируются принципы новой теории эволюционного моделирования (ТЭМ), предназначенной для построения *моделей широкого класса систем* со следующими основными характеристиками:

- **Сложные** – состоящие из большого числа элементов и взаимодействий – связей (мера сложности по Колмогорову – это длина минимального описания системы).
- **Динамические** – изменяющие свои структурные и функциональные характеристики, включая возможное расширение.
- **Слабо детерминированные** – элементы системы могут по-разному реагировать на одинаковое воздействие. Понятие слабой детерминированности может частично формализоваться и выявляться, в частности, с использованием теории вероятности, нейронных сетей, DataMining-технологий и др.
- **Открытые (незамкнутые)** – обменивающиеся информацией с окружающей средой (в том числе, включающие сознание человека).

Существуют традиционные подходы к моделированию сложных систем, которые реализуют последовательную технологию многоуровневой декомпозиции, усреднения и упрощения, выделения ряда существенных (ключевых, с наибольшим весом...) характеристик, выявления основных законов и зависимостей. На их основе и строятся модели для управления, мониторинга, прогнозирования.

Однако такой подход эффективен только для описания поведения формализованных, жестких, динамически неизменяемых, механистических систем в ограниченном пространстве в ограниченные периоды времени между точками бифуркации. При усложнении систем, их динамическом изменении, увеличении частоты бифуркаций теряется адекватность поведения модели.

Именно попытки создания адекватных моделей социально-экономических систем проявили и обострили проблемы высочайшего динамизма изменений *структуры, состава, поведения, состояний* их элементов.

Особенности заключаются еще и в том, что наряду с высокой степенью непознанности законов этой предметной области, формализованные открытые законы проявляют новые иные свойства, к тому же в системы постоянно вносятся новые, в разной степени исполняемые законы, декларируемые произвольными группами социума.

Короткое время «постоянства» в жизни модели и необходимость воплощения ее в едином пространстве воздействий, взаимодействий и восприятий заставляют по-иному рассматривать технологию и архитектуру проектирования модели.

Изменения в жизни структуры и элементов модели, их описание и эволюция явились *метаданными* для изучения и формализации.

В отличие от традиционных подходов в теории эволюционного моделирования предметной области ставится в соответствии единое описание элементов и их взаимодействий. Это описание поддерживает и отражает историю эволюционных изменений в структуре, составе, алгоритмах поведения, формах представления, состояниях моделируемой предметной области. Изменения модели могут быть продиктованы углублением знаний о предметной области.

Моделью предметной области в теории эволюционного моделирования является **единый гиперграф классов**. В настоящей работе описана его архитектура и алгоритм построения.

Основными целями использования единой эволюционной модели в ТЭМ являются:

- познание (исследование, изучение, выявление законов);
- эффективное управление;
- гармонизация процессов;
- накопление знаний в общей для социума нотации (на едином языке);
- объединение знаний и законов разрозненного множества научных дисциплин в едином информационно-функциональном пространстве;
- изменение принципов обучения: энциклопедичность суммы знания усиливается технологией выращивания знания;
- прогнозирование эволюционного появления нового порядка-предмета более высокого уровня интеграции.

Теория эволюционного моделирования формирует новые принципы дальнейшего развития вычислительных средств и средств разработки программного обеспечения, в том числе систем управления базами данных.

Теория совершенствует концепции моделирования, разработанные в общей теории систем, в объектно-ориентированном проектировании и при построении универсального языка моделирования UML [10; 14; 19] (в работе приведен критический анализ основных концепций UML).

Теория эволюционного моделирования качественно изменяет существующие подходы к созданию и использованию *автоматизированных систем управления*, когда:

- либо системы автоматизации последовательно итерационно строятся на основе наборов описаний имеющихся, порой не оптимальных, бизнес-процессов,
- либо бизнес-процессы приспосабливаются, порой также не оптимально, под приобретенные типовые наборы систем автоматизации.

Хотя чаще используется сложный паллиатив этих подходов.

Использование ТЭМ для реализации, например, бизнес-приложений и ERP (Enterprise Resource Planning)-систем создает **новые технологии эволюционного проектирования единой виртуальной модели управления** отдельными предприятиями, транснациональными компаниями, отраслями, регионами, государствами и другими социально-экономическими объединениями.

Единая виртуальная модель управления позволяет увидеть всю архитектуру управления, рассмотреть различные варианты и обоснованность реструктуризации, создать технологию осознанного реинжиниринга, гармонизировать методы централизации и децентрализации управления, минимизировать потоки данных, оптимизировать организационные структуры, повысить эффективность управления ресурсами, сбалансировать цели и методы управления и т.п.

Идеи и принципы ТЭМ были эффективно использованы в Компании «Цефей» при разработке *Эталон-CASE-технологий* для эволюционного проектирования моделей сложных, открытых, динамических, слабо детерминированных систем и *Корпоративных Систем Управления «Эталон»*.

Для анализа справедливости положений теории эволюционного моделирования и с целью ее дальнейшего совершенствования на *Эталон-CASE-технологии* была спроектирована *Эталон-ERP-система* с возможностью создания **библиотеки реальных используемых моделей управления** в различных предметных задачах, отраслях, странах и т.п.

Сформированная библиотека моделей-гиперграфов служит для формализованного хранения знаний, обобщения эволюционно накапливаемого опыта, многостороннего анализа, оптимизации, дальнейшего использования прототипов, обучения, исследования...

Реализованные проекты, использующие *Эталон-решения*, работают в крупнейших корпоративных структурах и органах государственного управления.

Эталон-решения на основе теории эволюционного моделирования реализуют полный технологический цикл создания модели предметной области и получения программных приложений конечного пользователя:

- *Эталон-язык* – структуризация и описание предметной области,
- *Эталон-CASE* средства сетевого эволюционного проектирования, в том числе модернизации и развития модели предметной области – гиперграфа классов,
- *Эталон-формирование виртуальных приложений* – динамически настраиваемых виртуальных систем управления – рабочих мест доступа (desktops) конечного пользователя,
- *Эталон-генерация* информационных хранилищ,
- *Эталон-компиляция* исполняемого программного кода,
- *Эталон-исполняемая модель* – использование программного продукта конечным пользователем.

Одним из основных показателей качественно высокой эффективности применения эволюционного моделирования является **соотношение функциональности модели, сроков и стоимости ее создания и развития**.

Например:

Модель-гиперграф управления бизнесом, предположим, транснациональной компании с диверсифицированными видами

деятельности насчитывает *сотни тысяч элементов*: десятки тысяч атрибутов, тысячи таблиц, десятки тысяч форм ввода, отчетов, документов, тысячи методов и процедур обработки данных.

Она хранит и обрабатывает информацию об экземплярах предметной области в объеме *миллионов записей*.

При использовании Эталон-CASE-технологии, реализующей принципы теории эволюционного моделирования, единая модель-гиперграф проектируется в сетевом режиме группой *3–10 человек – системных аналитиков – за 2-3 недели*, в отличие от аналогов, проектируемых годами тысячами разработчиков.

При этом программный код един для различных моделей и составляет около 100 Мб, в то время как программный код аналогов, реализующих обработку подобных информационно-функциональных задач, составляет около $n * 100$ Гб, где n – число компаний, предлагающих ERP-системы.

Но главным является то, что полученные Эталон-модели имеют концептуально новую архитектуру, принципиально по-иному отражают жизнь самой модели, ее эволюцию во времени и пространстве. Они создают **единую модель состояний, воздействий, взаимодействий и восприятий**.

Применение теории эволюционного моделирования позволило выявить проблемы в современных подходах экономических наук и концептуальное несовершенство существующих международных стандартов финансовой отчетности IAS и GAAP. Исследование моделей и результатов их использования явно выявили дисбалансы управления:

- общемировыми природными ресурсами;
- реальным сектором экономики, созданием прибавочной стоимости в условиях глобализации;
- сектором виртуальных финансовых инструментов эксплуатации денежных заимствований всего дееспособного населения;
- мировыми тенденциями налоговых законодательств, в том числе специальных экономических зон;
- интеллектуальными и психологическими ресурсами социума,
- и т.п.

3. Эволюционное моделирование – приоритет развития информационных технологий

В современном мире труд самого многочисленного слоя населения большинства развитых стран – «пролетариата» – связан не с реальными производственными процессами, механизмами, инструментами, материалами и т.п., а с информацией о течении производственных процессов, о действиях механизмов...

Труд подавляющего большинства профессий также невозможен без использования современных информационных технологий.

Рассмотрим, например, предметную область, которая на сегодняшний день занимает более 60% рынка прикладных программных продуктов: бизнес-системы управления финансово-экономической деятельностью различных предприятий и корпоративных структур.

Применение передовых высоких технологий обеспечивает существенное снижение себестоимости основного производства различных товаров и услуг. Разрабатываются многочисленные параметризованные алгоритмические модели управления производственными процессами, позволяющие оптимизировать производственные программы и необходимые для их реализации производственные ресурсы. Сформированы и постоянно развиваются методы контроля эффективности производства. Планирование, учет и анализ прямых затрат на производство готовой продукции может вестись по различным разработанным и совершенствующимся методикам.

Минимизация затрат на оплату труда производственного персонала, соответствующих отчислений и налогов привели к эффективной модели глобального перераспределения производства.

На сегодняшний день в этой области в центре внимания находятся проблемы *повышения производительности и эффективности труда именно управленческого персонала, снижения издержек управления бизнес-процессами.*

Простое переложение существующих управленческих моделей бизнес-процессов на язык электронного документооборота уже показывает свою неэффективность.

Шумные банкротства транснациональных компаний, перекосы в оценке (скорее, в самооценке) труда менеджеров, дисбалансы целей и методов управления, «мыльные пузыри» целевых капитализаций и банкротств и т.п. приводят к востребованности эффективных сбалансированных моделей управления, прежде всего, бизнес-процессами.

Сейчас мало рассматривать возможности современных информационных технологий, как технологий удобства и ускорения исполнения уже существующих функций, бизнес-процессов.

Множество документов, инстанций, подписей, визирований, регламентов, вынужденных регистраций промежуточных решений было в большой степени

продиктовано распределенностью необходимой информации в пространстве и во времени.

Новые информационные технологии и предлагаемое в настоящей работе **эволюционное моделирование бизнес-процессов в едином информационном и функциональном пространстве** в виде гиперграфа классов выявляют «лишние», не оптимальные процессы и сущности. Они позволяют последовательно трансформировать традиционные понятия «поток документов» в системы единого восприятия состояний ресурсов и процессов во времени, возможно, в реальном времени.

Эволюционное динамическое моделирование, открытость, наглядность единой модели-гиперграфа могут стать основой систематизации, исследования, изучения, универсализации, оптимизации «искусства» управления.

Кроме того, при сегодняшнем уровне глобализации *необходимо рассматривать сложную совокупность различных экономически связанных управленческих моделей множества финансово-экономических организационных единиц с целью их гармонизации.*

Создание корпоративных систем управления ресурсами конкретного предприятия или холдинга в условиях глобализации необходимая, но не достаточная для эффективного управления задача. Каждая бизнес-единица является частью целого – единого глобального финансово-экономического, политического, социального, экологического и т.п. пространства.

Эволюционное познание и динамическое моделирование целостной картины – единственный метод гармонизации развития.

Теория эволюционного моделирования не только концептуально изменяет подходы к проектированию и описанию моделей предметной области, но и четко **формирует новый сегмент** стандартов развивающихся современных информационных технологий.

Основные элементы структуры современных информационных технологий (ИТ) приведены на Рис. 1.

Для эффективного обеспечения решения задач конечного пользователя в настоящее время согласованно развиваются три основных направления информационных технологий:

- *Среда передачи данных,*
- *Аппаратное обеспечение,*
- *Программное обеспечение.*

Среда передачи данных формировалась с использованием различных физических объектов – почта, дискеты, медные провода, оптика, радиоволны...

В настоящее время идет глобальная унификация стандартов передачи, мультиплексирования, защиты, помехоустойчивости и т.п. информации и поиск новых принципов передачи данных.

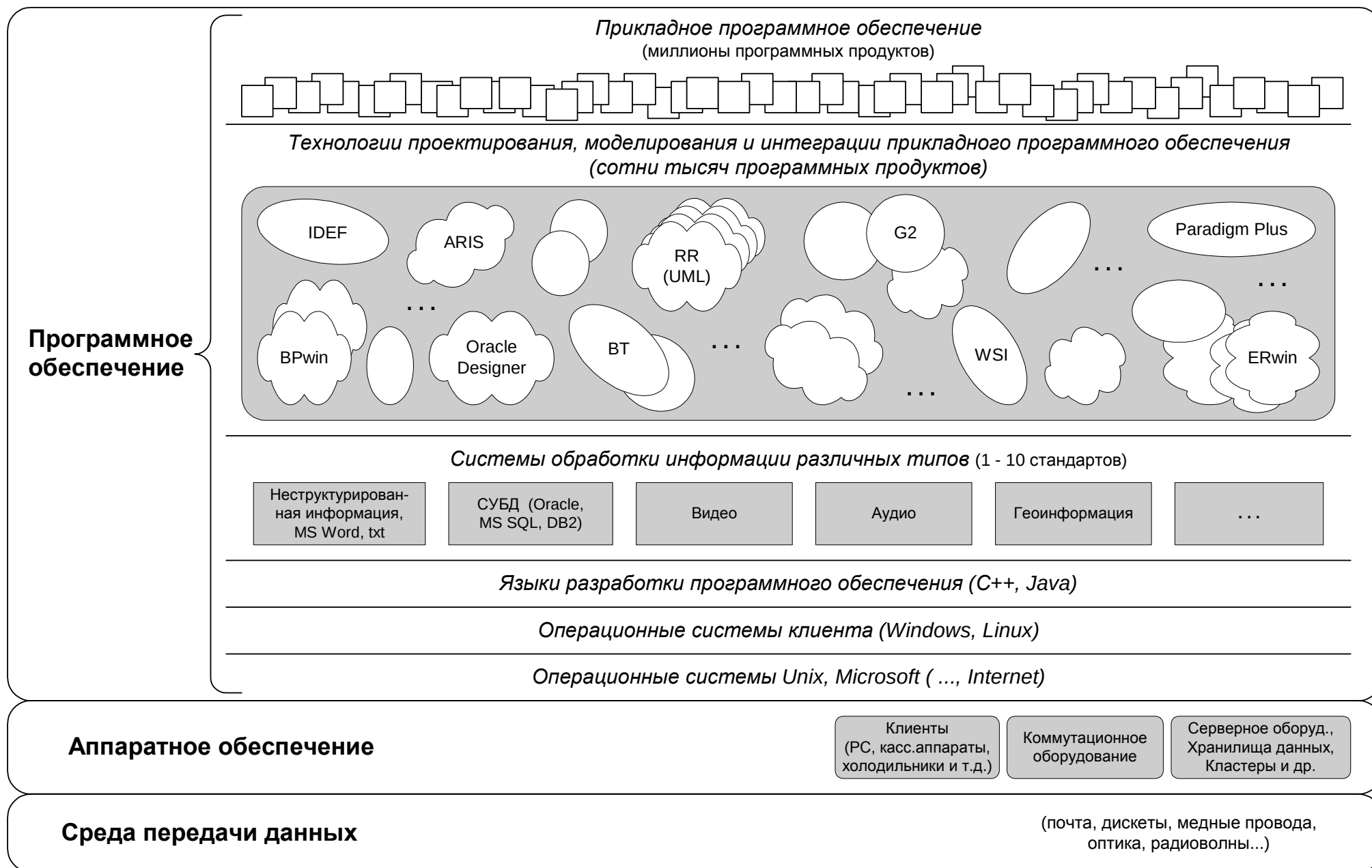


Рис. 1. Основные элементы структуры современных информационных технологий

Аппаратное обеспечение хранения, обработки, передачи, использования информации можно разделить на три основных вида:

- Аппаратно-вычислительные комплексы, предназначенные для логически централизованного, а физически, возможно, территориально распределенного хранения больших объемов информации и обработки их по сложнейшим алгоритмов вычислений;
- Аппаратно-вычислительные комплексы, предназначенные для коммутации и различного преобразования передаваемой информации;
- Клиенто-ориентированные аппаратно-вычислительные комплексы или вычислительные средства, встроенные в любое оборудование и механизмы (персональные компьютеры, кассовые аппараты, автомобили, телефонные аппараты, холодильники, печи, утюги и т.п.).

В этом сегменте также интенсивно ведется глобальная унификация стандартов. Появление инновационных продуктов в течение короткого времени порождает хаотическое предложение тех или иных вариантов содержательных решений. Но вскоре все решения стандартизуются, и вариативной остается, в основном, только форма представления и использования.

Структура программного обеспечения несколько нарушает стройную систему глобальной стандартизации элементов информационных технологий.

Программное обеспечение (ПО) состоит из следующих основных направлений:

- *Сетевые операционные системы*, где более 80% рынка потребителей ИТ используют всего две группы систем;
- *Операционные системы аппаратно-вычислительных средств клиентских частей*, где более 80% рынка потребителей ИТ используют всего две системы;
- *Языки разработки программного обеспечения*, где более 80% современных разработчиков в настоящее время используют всего два языка;
- *Системы управления и обработки информации различных типов* (неструктурированная информация, структурированная информация, аудиоинформация, видеоинформация, геоинформация и т.п.), где более 80% рынка потребителей ИТ используют 1-10 стандартных систем обработки;
- *Программные средства проектирования, моделирования, интеграции* – на рынке представлены сотни тысяч различных программных продуктов;
- *Прикладное программное обеспечение конечного пользователя*, специализированное для решения конкретных целей и задач – на рынке существуют миллионы разрозненных прикладных программных продуктов (ППП).

Именно последние два направления развития программного обеспечения из вышеприведенных:

- ✓ *программные средства проектирования, моделирования и интеграции;*
- ✓ *прикладное программное обеспечение,*

являются наиболее интересными для исследования и познания.

Приоритет их интенсивного развития очевиден и обоснован.

Традиционные подходы к созданию прикладных программных продуктов, описанию, моделированию, проектированию любой системы приводят к тому, что разработчик или группа разработчиков начинают свою деятельность практически каждый раз с нуля, т.е. с личного опыта, накопленных заготовок (порой устаревших), наличия произвольной информации об аналогах.

Качество проектируемых прикладных программных продуктов зависит только от талантов, психологических, личностных характеристик конкретных людей, а также от организационных, технологических, финансовых, временных аспектов.

В результате мы имеем *глобальные баснословные финансовые, материальные, технологические, интеллектуальные издержки* на создание миллионов различных прикладных программных продуктов по сути своей реализующих в той или иной степени одинаковые задачи предметной области.

На *Рис. 2* представлены проблемы диспропорции создания прикладных программных продуктов, когда для решения одного круга задач в глобальных масштабах создано огромное количество разрозненных программ, а многие другие предметные области полностью обойдены вниманием.

Кроме того, эти различные созданные и используемые программные продукты «не понимают» друг друга. Они не совместимы.

Большинство усилий в современной науке направлено на создание универсальных шлюзов и технологий формирования и сопровождения интерфейсов – «среды общения» между несопоставимыми программами. Ожидается открытие секрета такого универсального «клея» для воды, камня, газа, бумаги, кислоты и т.п.

Есть, несомненно, тенденции кажущегося упрощения этого многотрудного занятия.

Глобализация подготовила предпосылки к унификации сетевых операционных систем, операционных систем клиентского программного обеспечения конечного пользователя, систем управления базами данных, языков программирования и т.д.

Но даже, если, предположить, что среда интеграции прикладных программных продуктов будет создана, то это никак не затронет решение проблем и снижение издержек сопровождения самих разрозненно спроектированных систем, их оптимизации, динамического развития...

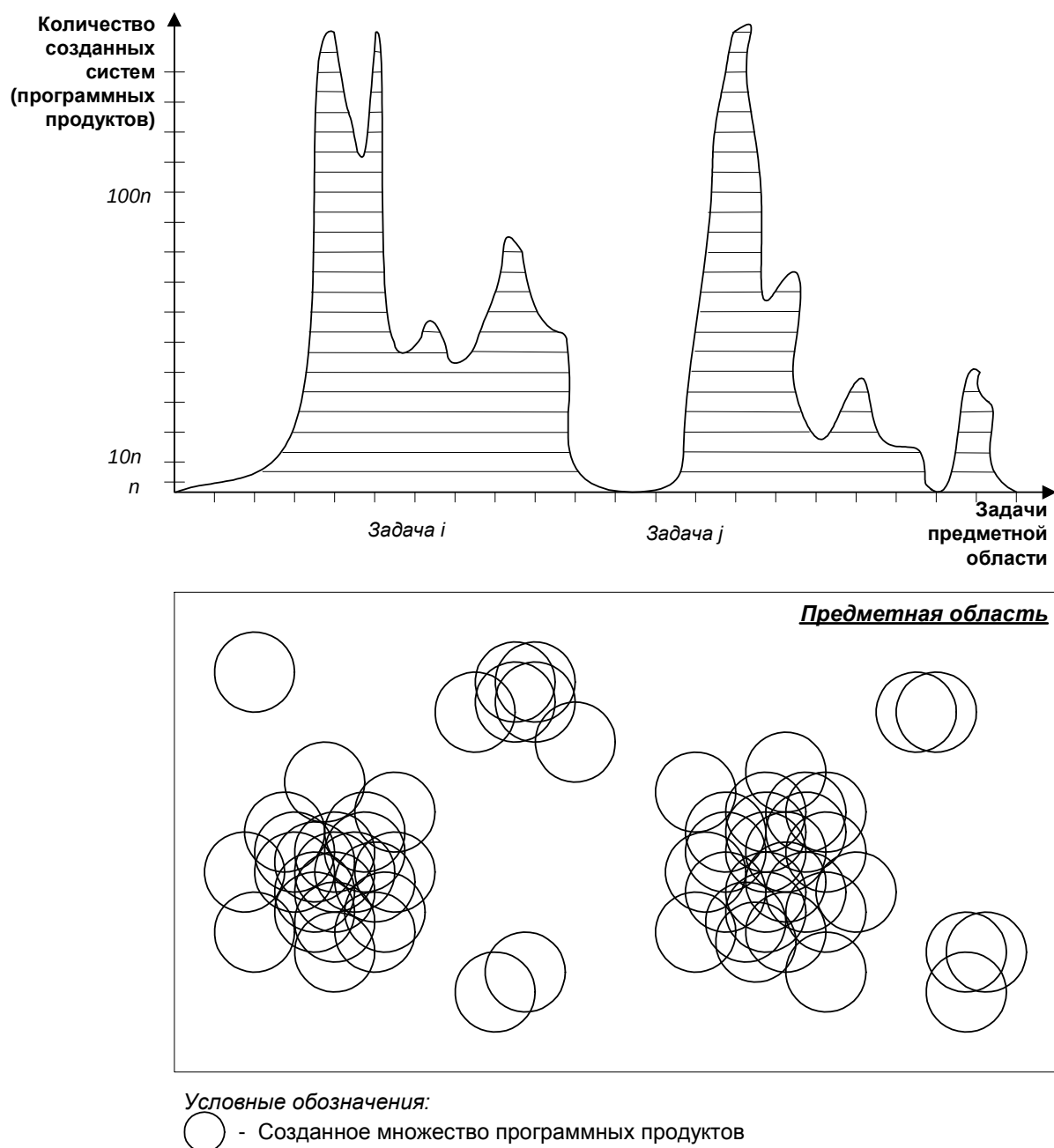


Рис. 2. Диспропорция создания систем управления

Чтобы понимать и описывать реакции химического взаимодействия веществ, нужна периодическая таблица Менделеева.

Чтобы решить описанные проблемы развития информационных технологий, необходимо согласованно эволюционно моделировать предметную область, понимая друг друга и используя накапливаемые в моделях знания. Необходим единый эволюционирующий язык проектирования моделей.

Создание единой среды эволюционного моделирования – приоритет развития ИТ (см. Рис. 3).

Среда эволюционного моделирования формирует основы создания единой модели предметной области и виртуального описания прикладных исполняемых систем, как зон видимости и доступа к любым произвольным наборам элементов этой модели.



Рис. 3. Предлагаемая технология создания прикладного программного обеспечения

Созданные и используемые Эталон-решения эволюционного моделирования показали, что возможна согласованная, эффективная сетевая работа сотен проектировщиков в пространстве единой модели.

Разработаны технологии распределенного проектирования частей модели с дальнейшим объединением – слиянием гиперграфов и анализом избыточности единой модели.

Средства проектирования, сопровождения и развития моделей просты, их легко осваивают специалисты партнеров и клиентов, приобретающих Эталон-продукты для реализации конкретных задач, их продвижения, развития и использования.

Представленный проект Эталон-решений эволюционного моделирования масштабируем и может работать на основе большинства из описанных информационных технологий.

Используя технологии B2B, можно обмениваться моделями, встраивать дополнительные части моделей-прототипов, анализировать и обучаться на иных решениях однотипных задач и т.п.

Результаты применения Эталон-решений обосновали несомненные преимущества новых технологий проектирования приложений конечного пользователя, основные принципы которых и рассмотрим в настоящей работе.

4. Критический анализ UML

4.1. Краткое описание UML

Вначале дадим краткое описание UML. Универсальный язык моделирования UML был разработан как методика визуального моделирования, используемая при разработке программных систем на основе объектно-ориентированного подхода. Примером использования стандарта UML является программный продукт Rational Rose.

Основными в UML являются следующие понятия: *система*, *модель*, *диаграмма*.

Система – это множество элементов, организованных для достижения конкретной цели.

Модель – это семантически законченная абстракция системы.

Диаграммы – это специальные графические конструкции.

Моделирование в UML – это последовательное построение статических и динамических диаграмм.

Модель системы в нотации UML задается *совокупностью* следующих диаграмм: прецедентов (use case diagram; другой перевод – диаграммы вариантов использования), классов, состояний, деятельности, последовательностей, кооперации, компонентов, развертывания.

На метамодельном уровне UML организован в три пакета: «Основные элементы», «Элементы поведения» и «Общие механизмы». Пакеты состоят из подпакетов.

В частности в подпакет «Элементы ядра» пакета «Основные элементы» входят основные метаклассы языка UML: класс, атрибут, ассоциация, классификатор, интерфейс и т.д.

Пакет «Элементы поведения» содержит подпакеты «Общее поведение», «Кооперация», «Варианты использования», «Автоматы».

В подпакет «Общее поведение» пакета «Элементы поведения» входят такие элементы, как объект, действие, связь, экземпляр, сигнал и др.

В подпакет «Варианты использования» входят такие элементы, как актант (actor), прецедент (Use Case) и др.

Пакет «Общие механизмы» содержит подпакет «Управление моделями», который специфицирует основные элементы языка UML. В нем определяются следующие базовые контейнеры: Модель, Пакет, Подсистема.

Пакеты со всем своим содержимым определены в некотором пространстве имен, которое позволяет единственным образом использовать имена всех элементов модели.

4.2. Диаграммы UML

Моделирование системы в UML начинается с построения диаграммы прецедентов, а затем диаграммы классов. Диаграмма прецедентов является наиболее общей концептуальной моделью системы, на ее основе строятся все остальные диаграммы. Диаграмма классов – это логическая модель системы, описывающая статические свойства ее структуры.

Основные элементы статического представления модели – классификаторы и их отношения. В число классификаторов входят классы, интерфейсы, типы данных, варианты использования, актанты (actors), компоненты, узлы и подсистемы.

Для построения диаграммы прецедентов используется актант – это идеализированная внешняя сущность, вступающая во взаимодействие с системой, подсистемой или классом. На диаграмме прецедентов изображается последовательность сообщений между системой и одним или несколькими актантами.

Диаграмма классов представляет собой граф, вершинами которого являются элементы типа «классификатор», связанные структурными соотношениями.

Набор диаграмм UML:

1. *Диаграммы вариантов использования (use case diagrams)*
 - для моделирования бизнес-процессов (требований к системе);
2. *Диаграммы классов (class diagrams)*
 - для моделирования статической структуры классов системы и связей между ними;
3. *Диаграммы поведения (behavior diagrams)*
4. *Диаграммы взаимодействия (interaction diagrams)*
 - для моделирования процессов обмена сообщениями между объектами. Имеется два вида диаграмм взаимодействия:
 - диаграммы последовательности (sequence diagrams);*
 - кооперативные диаграммы (collaboration diagrams);*
5. *Диаграммы состояний (statechart diagrams)*
 - для моделирования поведения объектов системы при переходе из одного состояния в другое;
6. *Диаграммы деятельности (activity diagrams)*
 - для моделирования поведения системы в рамках различных вариантов использования или моделирования деятельности;
7. *Диаграммы реализации (implementiotion diagrams)*. Имеется два вида диаграмм реализации:
 - диаграммы компонентов (component diagrams)*

- для моделирования иерархии компонентов (подсистем) системы;

диаграммы размещения (deployment diagrams)

- для моделирования физической архитектуры системы.

4.3. Проблемы UML

На основе анализа UML можно сделать следующие замечания:

1. UML не дает ответа на главный вопрос – что такое модель системы. Вместо единой модели системы предлагается разрозненный набор диаграмм, описывающих различные аспекты структуры и поведения системы. Таким образом, универсальный язык моделирования UML не решает им самим поставленную задачу – построить модель системы.
2. По существу UML является не языком, а набором трафаретов, поскольку набор основных понятий (слов) – в первую очередь классов – не определен и не фиксирован. Нет списка основных (базовых) классов. Это отсутствие формализации придает UML большую общность, но в тоже время лишает его определенности при построении моделей. Можно сказать, что UML – это язык без слов!
3. В UML используется чрезмерно широкие безграничные определения, которые приводят к применению различными группами проектировщиков субъективных внутренних соглашений для UML нотаций. Нотации перестают быть стандартными.
4. UML за понятием «универсальный» прячет отсутствие формализованного подхода к описанию классов и проектированию самого информационного пространства, максимально перенеся процесс структурного и системного подхода на проблемы этапности итераций, их спецификацию, контролируемость, документированность, авторизацию и т.п. Большинство признается, что при первой же попытке построить какую-либо модель приходится изобретать и использовать «стереотипы» – средства, выходящие за рамки UML.
5. В UML не приведены формальные правила или критерии выделения классов в моделируемой предметной области.
6. UML представляет собой формальный синтез (объединение) огромного числа общих понятий, вследствие чего он сложен как в использовании, так и в изучении. Правильнее было бы выделить элементарные базовые понятия, а следующие строить уже в их терминах, чтобы новый концепт опирался на существующие, имеющие только одно значение.
7. Моделирование системы в UML опирается на построение, прежде всего, диаграммы прецедентов (use case diagram), которая описывает поведение системы с точки зрения целей ее пользователя, ошибочно полагая, что предмет, его свойства и поведение зависят от целей использования. Это привносит в моделирование элементы субъективизма и, в конечном счете, риск неадекватного отражения существенных свойств системы.

8. Деление предметной области на актанты и прецеденты является субъективным и произвольным, что может привести к построению модели, неверно отражающей состояние и поведение описываемого предмета. Кроме того, создаются предпосылки для порождения многочисленных интерпретаций одного предмета в несопоставимых нотациях в зависимости от различных рассматриваемых и используемых свойств.
9. Отсутствует формализованная методика построения диаграмм прецедентов, классов и других диаграмм.
10. Некоторые диаграмм UML являются производными и не должны содержать дополнительную семантику, которая непрозрачна в диаграммах источников – теряется согласованность диаграмм.
11. Хотя UML предназначен для описания моделирования, а не непосредственно для проектирования работающих приложений, архитектура построенных на его основе процедур существенно использует *итеративный* процесс разработки. Методика итераций не всегда эффективна, одним из ее недостатков является многократное повторение одних и тех же процедур, многократно дискретно перерабатывающих описываемую модель, что препятствует созданию моделей сложных систем.
12. В UML многое зависит от «правильности» выделения «основных элементов», но определения и метода структуризации элементов нет, они субъективны.

Кроме того, многократно усложняется процесс логически распределенной разработки проектных решений или использования «чужих» шаблонов проектирования.
13. Наряду с итеративным процессом разработки нескольких видов моделей, в UML существует понятие интеграции, которое существенно усложняет процессы согласования, что делает практически невозможным создание *сложных динамических* систем.
14. Применение в UML множественного наследования приводит к путанице и ошибкам. Как отмечено в [10, с. 95], «трудно определить, что откуда взялось».

Принцип множественного наследования несвойственен процессу эволюции классов. Он разрушает целостность проектирования. Наличие подобных множественных отношений характерно процессу жизни объекта (мутация свойств).
15. Диаграммы UML поддерживают моделирование систем с точки зрения процессов. Такая технология моделирования приводит к возможности многократного повторения одной сущности в разных аспектах функционирования и разрушает целостность модели (например: поставщик, подрядчик, контрагент, эмитент, кредитор..., юридическое лицо – единая сущность в разных нотациях).
16. Модель UML отделена от проектирования фактической системы конечного пользователя, и может не отражать всех свойств и связей, которые

появляются в процессе разработки самой системы, как программного продукта. То есть, UML – не адекватен.

17. Существует непоследовательность и даже противоречивость определения ряда ключевых понятий в UML.

В частности:

- «Иногда трудно отличить объект от класса» [10, с. 48]. Объект определен [14, с. 430] как «индивидуальная, уникальная сущность», что противоречит данному там же описанию объекта, как конечного автомата. Если объект – это частный случай класса, то непонятно, каким образом объекты могут обмениваться сообщениями на диаграмме последовательностей, если для классов эта операция не определена.
- Взаимозависимые определения агрегации и ассоциации [14], а также ассоциации, обобщения, зависимости [14, с. 78].
- Определение наследования как специализации приводит к трудностям при его использовании.
- И т.д.

Выводы: UML – не язык.

Он не проектирует единую модель, не адекватен, не исполним, не ясен, не точен, не изучим, итеративен, не эффективен.

Предлагаемая теория эволюционного моделирования призвана устранить эти недостатки, то есть *разрешить конфликт между простотой, которую требует семантика языка моделирования, и богатством проектируемых нотаций.*

Отметим, что создание модели системы может быть как конечной целью разработки, так и промежуточным этапом для построения реальной информационной системы, программных приложений для решения задач конечного пользователя.

5. Основные понятия теории эволюционного моделирования

Чтобы изложить принципы теории эволюционного моделирования, введем несколько групп используемых понятий:

- *Предмет, процессы;*
- *Система, управление;*
- *Модель, эволюционное моделирование;*
- *Единое информационно-функциональное пространство.*

Последовательность рассмотрения понятий (предмет, система, модель), сохранена в соответствии с традиционными подходами теории систем управления для возможного сопоставления и анализа их концептуальной трансформации в новой теории эволюционного моделирования.

Для теории эволюционного моделирования логически более приемлема следующая последовательность понятий: предмет, модель, система.

5.1. Предмет, процессы

В теории эволюционного моделирования выделяется некоторая *предметная область* (возможно, вся вселенная), для которой необходимо создать адекватную модель.

При этом изменяется традиционный подход к моделированию, который прежде всего начинался с постановки *целей*, формирования *задач*, описания *функций*...

Новая теория эволюционного моделирования предполагает, что можно выделить у предмета свойства, которые не зависят от целей использования и управления. ТЭМ предлагает описание свойств предмета проводить эволюционно по мере их выявления и осознания, но однократно.

Понятия целей, задач, функций и т.п. будут вводиться на уровне формулирования систем управления. В данном случае они влияют на приоритеты рассматриваемых в определенный период времени свойств, их вес, существенность, алгоритмы управленческих воздействий и взаимодействий, доступ к исполняемой модели и т.п.

Предмет – это понятие, включающее реальные предметы: физические структуры, биологические организмы, организационные структуры (предприятие, государство...), а также виртуальные, мысленные конструкции и т.п. Предмет может быть материей, полем, энергией, сознанием... Предмет может состоять из других предметов.

Предметы находятся в динамическом взаимодействии (функционировании).

Предметная область – это совокупность предметов и процессов взаимодействия.

В ТЭМ предметная область описывается свойствами. Свойства включают следующие составляющие: параметры (характеристики) и методы (алгоритмы взаимодействия), которые могут эволюционно познаваться и изменяться во времени и пространстве с различной динамикой.

Традиционно предметные области объединяли предметы и процессы с однотипными свойствами, описывающими схожие по содержанию, форме и поведению конкретные экземпляры. Такой подход обосновал дисциплинарное деление предметных областей.

В теории эволюционного моделирования нет дисциплинарного деления рассматриваемых предметных областей.

Процессы взаимодействия делятся на два вида:

- *процессы жизни* предметов,
- *процессы управления* предметами и процессами их жизни, далее управление.

Деление процессов на такие виды является одной из существенных предпосылок к формализации базовых свойств элементов моделирования любой предметной области.

Процессы жизни предмета – это последовательность трех этапов действий (поведения):

- I. рождение (дар, появление, создание, синтез...)
- II. движение (рост, убыль, ремонт, модернизация...)
- III. смерть (трансформация, передел, разрушение, синтез...)

Однозначное выделение процессов жизни предметов, структуризация процессов, формализация и обобщение свойств является основой одного из существенных элементов моделирования – баланса, включающего объекты и методы автоматического балансирования, выявления дисбалансов процессов и состояний предметной области в проектируемых моделях.

Именно специфика динамики процессов жизни предметов является основой формирования общих подходов отражения в моделях свойств времени и истории преобразований и состояний.

Процесс управления – целевое воздействие разума, сознания и/или технологий, созданных разумом, на предметы и процессы жизни. В частности, управление может быть представлено наблюдением, описанием и т.п. Более детально управление описывается в следующем подразделе.

Формализация процессов управления создает основы архитектуры проектируемой модели предметной области и ряда базовых элементов.

Свойства любой предметной области, включающие, в том числе, описание процессов, бесконечны.

5.2. Система, управление

Функционирование предметов, их жизненные циклы, процессы трансформации, взаимодействия формализуются (описываются) или выделяются человеком в виде набора систем.

Для каждого предмета и процесса функционирования можно рассматривать неограниченное множество систем, как его составляющих, так его и использующих, сформулированных на основе различных принципов, целей, задач, функций...

Например:

«Предприятие и его функционирование» может являться для нас совокупностью множества систем: система управления производством, система налогового учета, система бухгалтерского учета, система управления ТМЦ (товарно-материальными ценностями), система бюджетирования, система мониторинга деятельности, система управления отношениями с клиентами и т.п., которые в свою очередь могут содержать другие подсистемы.

«Человек и его жизнь» — является элементом множества систем: биологической системы, системы образования, системы пенсионного обеспечения, системы занятости, системы здравоохранения и т.п.

Традиционное применение принципа структурного редукционизма для сложных динамических систем приводит к *порождению множества проблем, отсутствующих в реальной жизни предметов.*

Любая «оптимальная» структуризация и последовательная, иерархическая декомпозиция любой предметной области на составные части бесчисленно многовариантна в зависимости от субъективных целей и задач.

При этом неизбежно разрушаются и теряются реальные отношения элементов предметной области.

В то же самое время для каждой составной части формируются многочисленные интерфейсы взаимоотношений с другими составными частями, в той или иной степени описывающие связи субъективно разъединенных элементов единого целого, полученных вынужденной многоуровневой структуризацией сложной предметной области.

В традиционных подходах создание, сопровождение, развитие разрозненно структурированных систем для каждой составной части итерационно.

Каждый шаг итерации порой влечет за собой чрезвычайные, непредсказуемые последствия.

Любое изменение элементов одной из частей может порождать сложнейшие сетевые реакции необходимых соответствующих изменений в структуре отношений и элементах других связанных составных частей. В то же время, часть изменений может быть вообще не учтена, порождая противоречивость.

Для каждой из составных частей системы возможно бесконтрольное многократное параллельное формирование несопоставимых и неоднозначных

описаний одних и тех же сущностей и процессов, либо повторений, либо под одним и тем же именем могут быть описаны различные понятия.

Структурный, модульный подход на этапе создания разрозненных систем составных частей предполагает сложные процессы их дальнейшей интеграции, даже при условии неукоснительного выполнения организационно контролируемых предварительных общесистемных соглашений.

Любые регламентные механизмы согласования и исполнения общесистемных изменений, неизбежно порождаемых при последовательном познании и описании любой из частей предметной области, многократно увеличивают сроки создания и развития сложных систем.

Эти и многие другие проблемы являются основой необъяснимых «синергетических» эффектов при интеграции составных частей в единое целое, отсутствия целостности данных, их противоречивости, несопоставимости и т.п.

Для решения описанных проблем в теории эволюционного моделирования **принципиально** изменяется подход к понятию систем, так как они действительно обладают супердинамикой изменений, специфичностью условий формирования и использования, субъективизмом, «настроением»...

Именно человек определяет: для каких целей и, следовательно, под каким информационным углом будет рассматриваться предмет с точки зрения использования части свойств и процессов функционирования (процессов жизни и управления). Система – это субъективный взгляд человека на предмет и процессы. Системы по входящим в них множествам элементов могут пересекаться или быть частью других систем без ограничений.

Система – это множество элементов и процессов, сознательно выделенных и/или организованных для достижения нужной цели. Система в ТЭМ тождественна понятию *система управления*.

В теории эволюционного моделирования **системы управления виртуальны**.

В ТЭМ предметная область не разбивается на части, как бы она ни была сложна, для каких целей и задач она бы ни использовалась.

Описание познаваемых свойств предметной области ведется в любой последовательности эволюционно в единой целостной модели.

При виртуальной организации и описании систем управления определяются объемы требуемых полномочий и доступ к элементам единой модели предметной области (описание модели приведено в следующих разделах). То есть, выделяется состав элементов модели и назначается их использование одному из органов управления. Системы управления легко подвергаются любой виртуальной реструктуризации и другим изменениям.

Сопоставимость понятий «предмет», «система» и «модель» будет рассмотрена в следующем подразделе.

Системы управления имеют следующую структуру (см. Рис. 4):

- *Органы управления,*
- *Объекты управления,*
- *Процессы управления.*

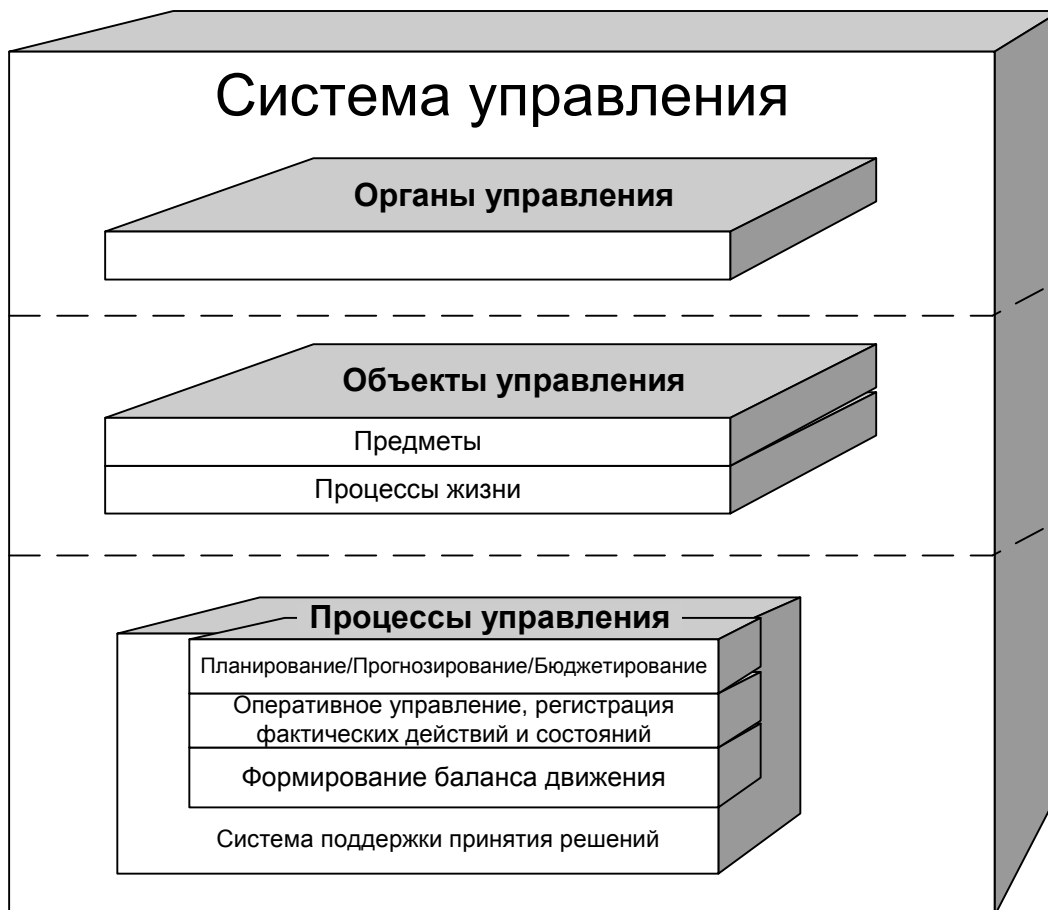


Рис. 4. Структура системы управления

Органы управления – организационная структура (иерархическая, матричная, сетевая и т.п.), в том числе автоматическая, генерирующая управляющие воздействия на объект управления. Органы управления могут быть как частью самого предмета, так и другим предметом.

Органы управления формируют часть общесистемных базовых свойств элементов модели, требования к их архитектурной структуризации и ряд технологических аспектов процесса моделирования для задания и отражения полномочий, авторизации действий, доступа, степени открытости и т.п.

Объекты управления – некоторая совокупность предметов и процессов их жизни.

Все объекты управления обладают рядом обязательных ключевых свойств, которые их идентифицируют и определяют независимо от целей использования в системах управления.

Выделение и спецификация объектов управления ведется методами нормализации свойств-атрибутов, их описывающих.

Органы управления при наличии целого ряда специфических общесистемных свойств являются также одним из объектов управления.

Процессами жизни предметов в системе являются:

- I. приход в систему/рождение/дар/появление/создание/синтез/...
- II. движение/рост/убыль/ремонт/модернизация/...
- III. уход из системы/смерть/трансформация/разрушение/синтез/...

Часть процессов жизни предмета II этапа, например движения, такие как приход-уход, рассмотрение-упрощение, приобретение-продажа и т.п., относительно рассматриваемой системы являются соответственно I и III этапом процесса жизни предмета для этой системы.

Процессы управления – отражают текущий уровень представлений о технологии и этапах управления. Они могут дополняться и развиваться.

Процессы управления включают:

- планирование, прогнозирование, бюджетирование,
- оперативное управление, регистрация фактических действий и состояний,
- формирование баланса движения,
- система поддержки принятия решений.

В следующих работах процессы жизни предметов и процессы управления будут рассмотрены подробнее.

Любые объекты управления (предметы и процессы их жизни) могут быть подвергнуты одному или всей совокупности видов управленческих воздействий.

Органы управления и процессы управления могут изменяться, оптимизироваться независимо (с другой динамикой) от объекта управления.

5.3. Модель, эволюционное моделирование

Понятие *модели* – в современной науке используется в различных нотациях, например, модель, как:

идея, пилотное решение, проект, прототип, проверка предположений, временный программный код, основа эксперимента, функция от весомых аргументов и т.п.

При этом принципы сопоставимости *предмета, системы и модели* не формализованы. Однако чаще всего (в том числе и в UML) их можно отобразить иерархическим графом (см. Рис. 5).

Предмет описывается некоторым множеством систем, а для каждой системы проектируется некоторое множество моделей. При этом предмет, в свою очередь, может состоять из других предметов, а система может быть совокупностью множества других систем. Для систем проектируются модели по различным принципам их формирования.

Например:

модели организационные, технологические, функциональные, событийные, структурные, регламентные, информационные, документооборота, ресурсные, процессные, классов, прецедентов и т.д., а также модели как совокупности различных комбинаций принципов создания.

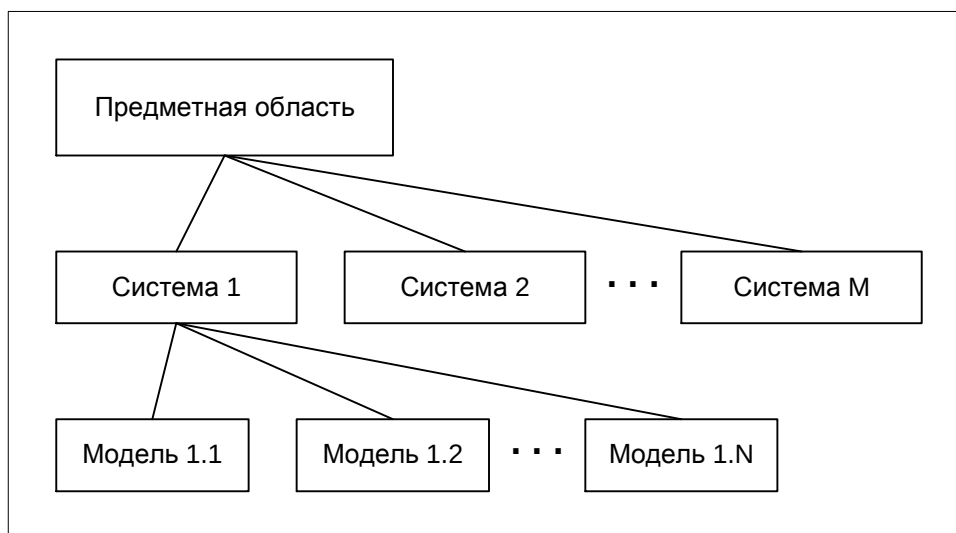


Рис. 5. Принципы сопоставимости предмета, системы, модели в традиционном подходе

Традиционно элементы систем и моделей представляются как конечные автоматы.

Конечный автомат A [14; 17] задается следующим набором данных: $A = \{X, Q, Y, f, g\}$. Здесь X - множество входных сигналов, Y - множество выходных сигналов, Q - множество состояний автомата, отображение f из $Q \times X$ в Y - функция выхода, отображение g из $Q \times X$ в Q - функция перехода.

Геометрически конечный автомат задается ориентированным графом. Вершины графа - состояния автомата; около каждого ребра (q_i, q_j) ставится буква входного алфавита x_k , вызывающая переход из состояния q_i в q_j (т.е. $q_j = g(q_i, x_k)$), и буква выходного алфавита y_s , которая появляется на выходе конечного автомата.

Отметим, что на сегодняшний день искусство определения сущностей на языке конечных автоматов интуитивно, субъективно и неформально. Элементарные конечные автоматы $\{A_i\}$ описывают стандарты представления результатов открытий, исследований, интуиции, предположений при формальном отображении предметов и процессов в элементы модели.

В дальнейшем проектирование сводится к синтезу – структурному объединению элементарных конечных автоматов.

Таким образом создавались сложные системные комплексы и модели, носящие в большей степени статический, детерминированный, механистический характер.

Изменение существа и параметров элементарных конечных автоматов приводило к новой итерации перепроектирования всего комплекса.

Есть ли общие законы формализации и проектирования элементарных конечных автоматов?

Как учесть неопределенность динамики их изменений?

Как эффективней, грамотней, на едином междисциплинарном языке представить все многообразие предметов и процессов?

Возможно ли моделировать прогнозы будущей эволюции предметов и процессов?

В отличие от традиционных подходов для решения поставленных вопросов предлагается **новая теория эволюционного моделирования адекватной, единой модели предмета и процессов взаимодействия на основе определенных базовых классов.**

Модель предмета – это отображение предмета в другой предмет (в том числе виртуальный). Модель описывается множеством элементов (классов) и связей между ними.

В ТЭМ **модель задается гиперграфом классов.** Определение и структура гиперграфа классов приведена в следующих разделах.

Проектирование единой модели предметной области может выполняться как в процессе познания предметной области, так и в случае необходимости создания систем управления для достижения конкретных целей и решения поставленных задач.

Требования к системам управления в данном случае *формируют приоритеты* в расширении, углублении изучения и исследования свойств рассматриваемой предметной области и описания элементов единой модели.

Принципы сопоставимости *предмета, системы и модели* в ТЭМ приведены на Рис. 6.

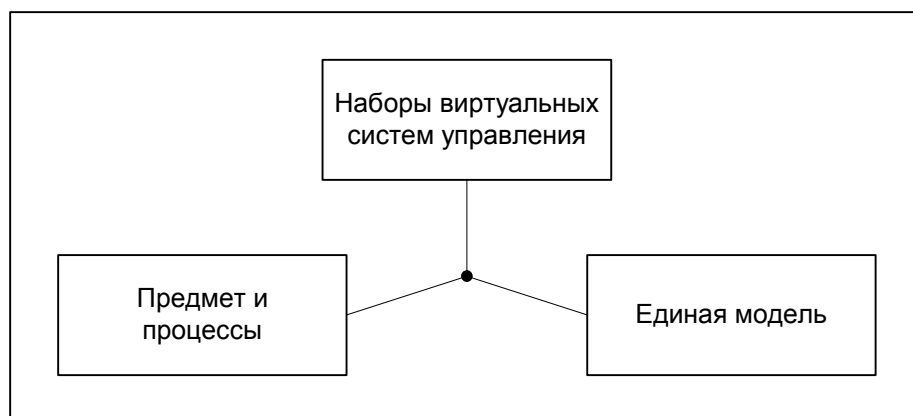


Рис. 6. Принципы сопоставимости предмета, системы, модели в ТЭМ

Предметная область проектируется в единой модели – гиперграфе классов.

Виртуальные системы управления представляют наборы подграфов единого гиперграфа – модели предметной области. При этом предмет может состоять из других предметов. Системы управления могут быть совокупностью элементов модели и других систем. В модели – гиперграфе классов можно выделить любое множество подграфов.

Процесс моделирования в ТЭМ может быть начат с реализации проектирования модели любой части предметной области, но должен вестись по определенным формализованным правилам, рассматриваемым ниже. При проектировании каждой новой задачи предметной области используется накопленный формализованный опыт – единая библиотека моделей, гиперграфов. Из имеющихся гиперграфов выделяются уже спроектированные необходимые элементы (формируется контейнер), в полученную основу модели лишь эволюционно добавляются новые элементы, свойства, связи.

Даже практический опыт показывает, что **прирост элементов и связей модели-гиперграфа с проектированием каждой новой задачи предметной области существенно сокращается**, следовательно, многократно увеличивается эффективность моделирования: скорость, надежность, качество, функциональность.

Моделирование. Применение термина «моделирование» имеет множество смысловых интерпретаций.

В предлагаемой теории эволюционного моделирования будем выделять следующие виды моделирования:

I порядка. Моделирование воздействий, состояний и реакций.

II порядка. Моделирование структур.

III порядка. Эволюционное моделирование.

Моделирование воздействий, состояний и реакций (моделирование данных) – моделирование входных сигналов и выходных сигналов элементов существующей модели. При этом модель в каждую единицу времени отражает состояние с учетом всей совокупности воздействий на нее и результатов взаимодействия элементов.

Моделирование структур (синтез элементов модели) – моделирование состава элементов и топологии их связей из множества имеющихся элементов модели.

Эволюционное моделирование – проектирование модели: создание и изменение элементов модели - классов, их свойств и связей. Проектирование осуществляется с использованием принципов объектно-ориентированного проектирования (ООП) – *наследования, инкапсуляции, полиморфизма*.

Эволюционное моделирование проектирует единую модель – гиперграф классов некоторой предметной области.

Эволюционное моделирование снимает проблемы необходимости структурного многоуровневого деления сложных систем на составные части – подсистемы с целью их раздельного модульного проектирования.

Эволюционное моделирование окончательно устраняет проблемы создания и динамической актуализации внутренних интерфейсов между составными частями – их (внутренних интерфейсов) просто нет.

Эволюционное моделирование для всей сложной модели создает и поддерживает стандарты внешних интерфейсов исполнения и использования, что многократно повышает системность, скорость их проектирования, обучаемость и взаимозаменяемость пользователей, прозрачность, простоту и т.п.

Эволюционное моделирование полностью исключает процесс интеграции составных частей сложных систем и появление необъяснимых эффектов от использования комплексных решений.

Эволюционное моделирование использует следующие понятия: **класс**, **объект**, **экземпляр**, которые соответственно характерны моделям, системам, предметной области.

В связи с реальным текущим научно-технологическим уровнем развития средств коммутации, хранения и обработки информации модель в ТЭМ имеет два состояния (см. Рис. 7):

1. **Проектируемая эволюционная модель** – элементами модели являются классы (исполнение возможно в технологии интерпретации или компиляции);
2. **Исполняемая модель** – элементами модели являются объекты.

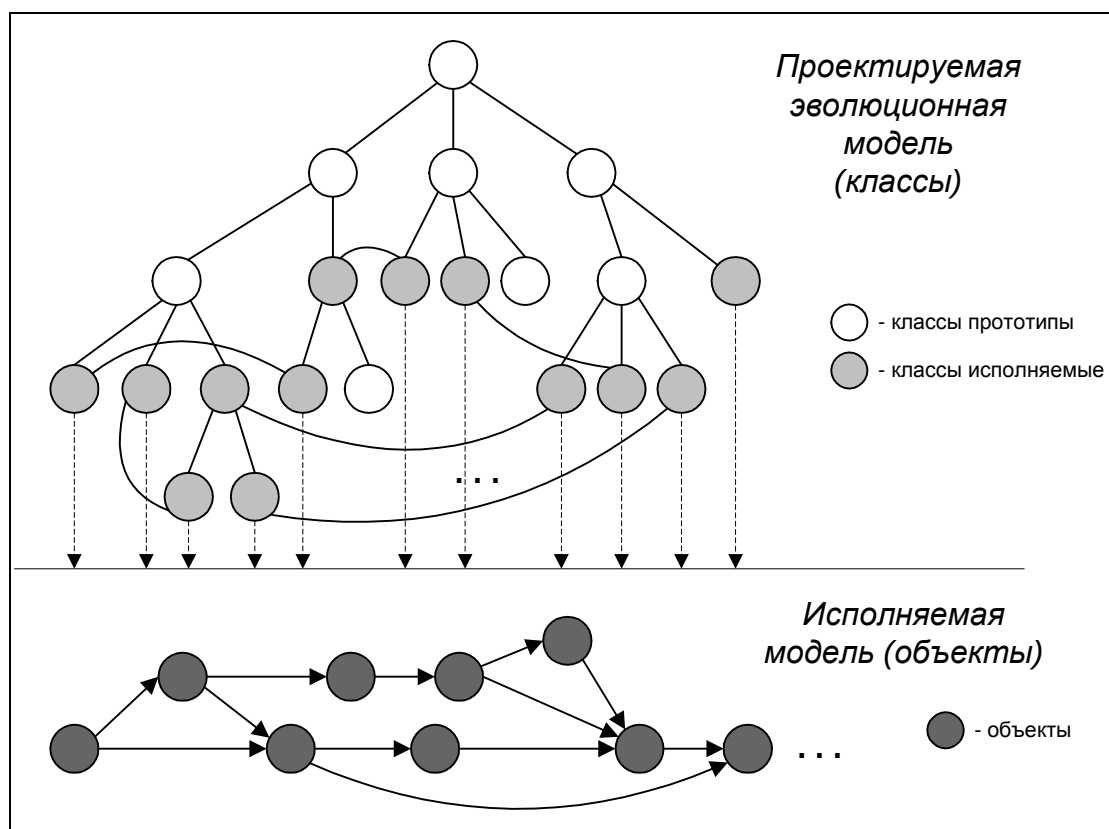


Рис. 7. Два состояния модели в ТЭМ

Эволюционное моделирование поддерживает специфические особенности свойств времени эволюции гиперграфа классов проектируемой модели и свойств времени предметов и процессов исполняемой модели.

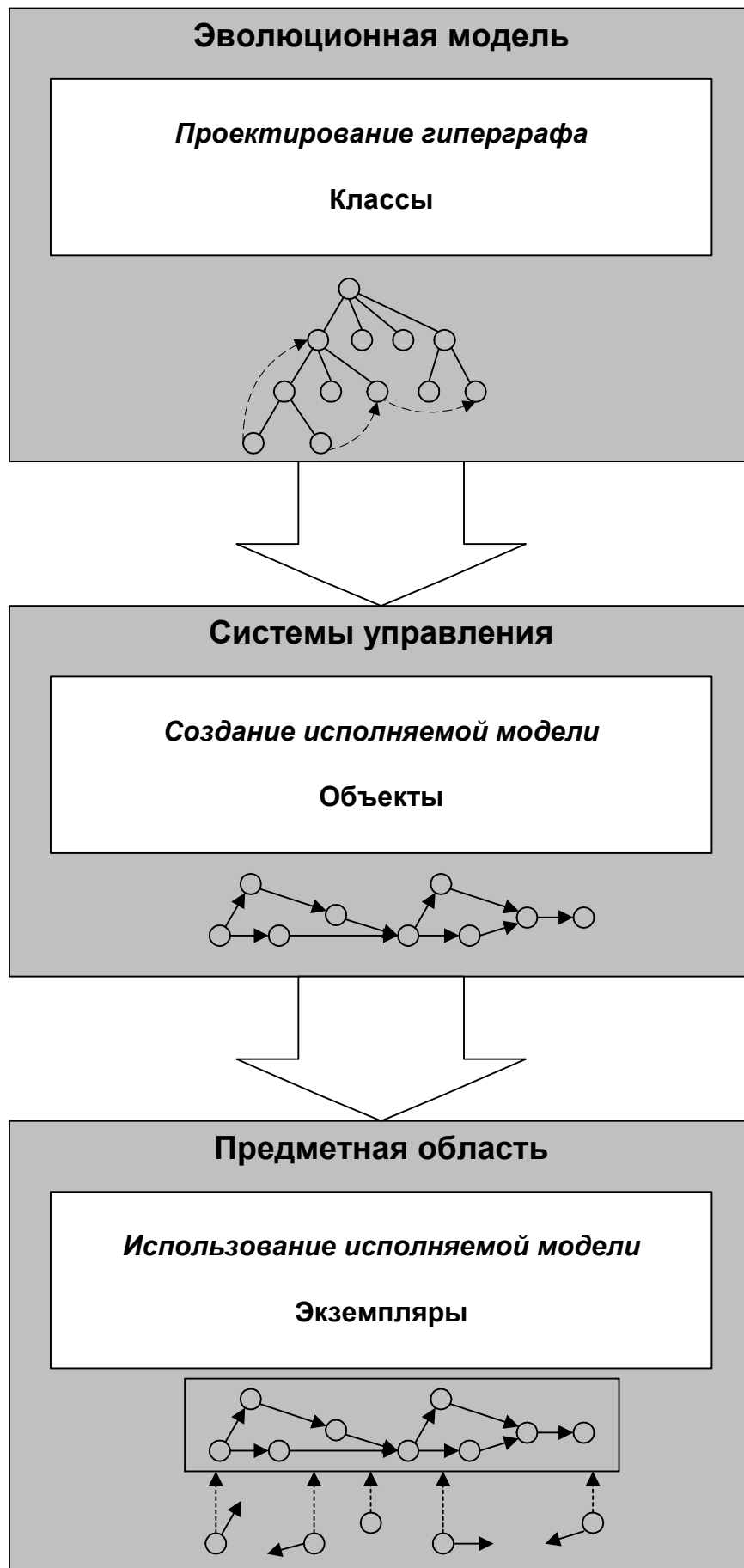


Рис. 8. Три стадии модели ТЭМ

На Рис. 8 представлены стадии проектирования, создания и использования модели в теории эволюционного моделирования.

Модель в ТЭМ проходит три стадии:

- 1) **Проектирование эволюционной модели** (гиперграфа) предметной области – создание *классов* и формирование всех видов связей. Формирование групп подграфов для создания систем управления.
- 2) **Создание исполняемой модели** – модели *объектов*, т.е. генерация хранилищ (памяти) и формирование (компиляция) исполняемых объектов на основе спроектированной эволюционной модели.

Исполняемая модель представляется набором сформированных виртуальных систем управления – описаний доступных элементов в программном приложении конечного пользователя.

- 3) **Использование исполняемой модели** – модели *объектов* для формирования системы состояний экземпляров предметной области, а также для управления *экземплярами* предметной области и процессами их жизни с помощью сформированных систем управления.

5.4. Единое информационное и функциональное пространство - репозиторий модели

Проектирование эволюционной модели осуществляется в *едином информационно-функциональном пространстве - репозитории модели*.

Эволюционное моделирование систематизирует и создает все виды связей между типизированными элементами единой модели – классами гиперграфа и формирует единое информационно-функциональное пространство отражения всех видов диаграмм зависимостей: организационных, технологических, функциональных, событийных, структурных, информационных, документооборота, ресурсных, процессных и т.д., а также совокупности различных их комбинаций.

Единое информационно-функциональное пространство – некоторое множество слов в некотором алфавите (напомним, что алфавит – это конечный набор символов; слово – это конечная последовательность символов; язык – это множество слов и принципов их соединения).

Единое информационное и функциональное пространство – единый репозиторий, содержит описание всей модели – гиперграфа классов.

Единое информационное и функциональное пространство модели содержит структурированную, нормализованную, целостную и непротиворечивую информацию обо всех элементах и связях и обеспечивает однократный ввод, регистрацию информации по местам возникновения и многократное использование одной и той же информации в различных задачах.

Программные инструменты эволюционного моделирования, в частности Эталон-CASE-технологии, предполагают сетевую (on-line) и распределенную (off-

line) работу неограниченной группы проектировщиков в едином информационно-функциональном пространстве одной модели, т.е. с единым гиперграфом.

Единое информационное и функциональное пространство любого проекта обрабатывается единым Эталон-программным (exe) модулем вне зависимости от состава, структуры и динамики изменений этого пространства (репозитория).

Все конечные пользователи работают в исполняемой области единого информационно-функционального пространства, но зона видимости каждого из них ограничена только той частью, которая определена правами доступа к элементам модели.

Эволюционное моделирование включает применение **различных операций над моделями (гиперграфом классов) и данными: сравнение, слияние, выделение, консолидацию, репликацию, экспорт, импорт и т.п. для распределенно проектируемых и исполняемых моделей в пространстве и времени.**

6. Модель предметной области – гиперграф классов

Результатом эволюционного проектирования модели является единый гиперграф классов. Набор всех возможных гиперграфов образует единое информационно-функциональное пространство.

Приведем сначала структурное описание гиперграфа классов. Любой гиперграф (эквивалентный термин – «сеть») задается множеством V , элементы которого называются вершинами, и семейством E наборов элементов множества V , называемых ребрами гиперграфа. Гиперграф обозначается $\langle V, E \rangle$.

Дадим теперь математическое определение гиперграфа классов, используемого в предлагаемой теории эволюционного моделирования. Напомним сначала некоторые стандартные определения. Все рассматриваемые множества предполагаются конечными. Множество $V = \{v_1, v_2, \dots\}$ и семейство $\{E_1, E_2, \dots\}$, в котором каждое E_i есть набор (возможно с повторениями) элементов из V , т.е. $E_i = \{v_{s_1(i)}, v_{s_2(i)}, \dots\}$, называется *гиперграфом* и обозначается $X \langle V, E_1, E_2, \dots \rangle$. Элементы множества V называются *вершинами*, а наборы $E_1, E_2, \dots$ – *ребрами* гиперграфа.

На Рис. 9 приведен пример гиперграфа $X = \{V, E', E''\}$.

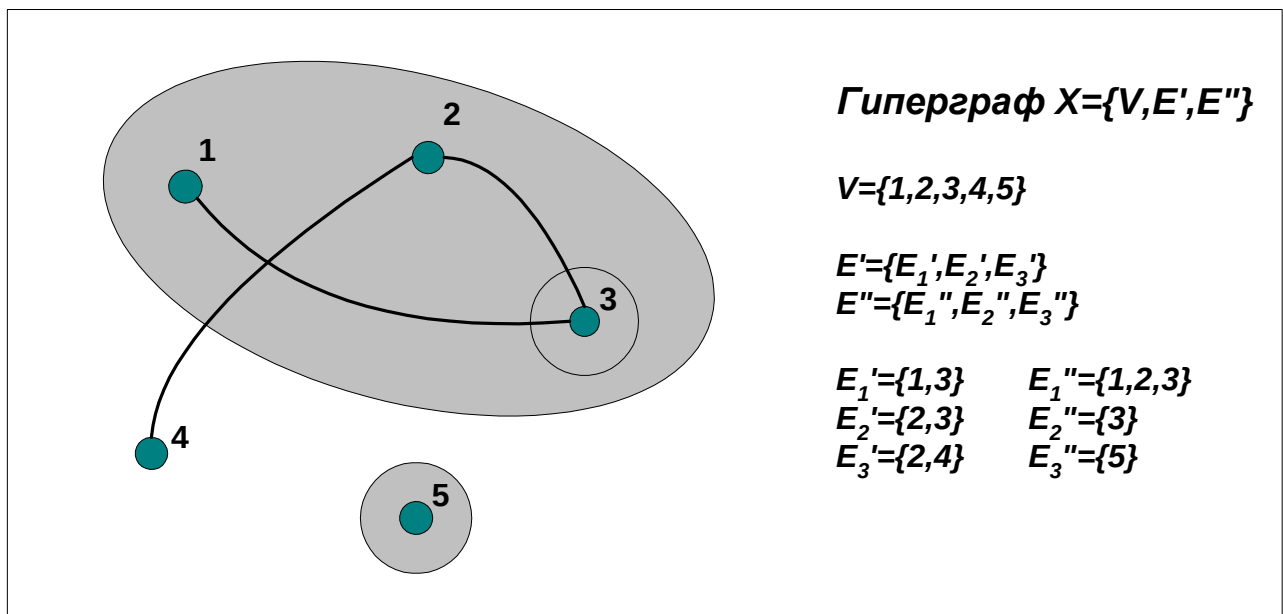


Рис. 9. Пример гиперграфа

Частным случаем гиперграфа является граф. *Графом* называется пара $\langle V, E \rangle$, где V множество, а набор E состоит из двух элементов множества V . Про ребро (v_i, v_j) говорят, что оно *соединяет* вершины v_i и v_j . Система ребер $\{(v_i, v_j), (v_j, v_k), (v_k, v_l), \dots, (v_m, v_n)\}$ называется *путем*, соединяющим

вершины v_i и v_n . Путь, не содержащий дважды одно ребро, называется *циклом*, если $v_i = v_n$. Граф называется *связным*, если любые две его вершины соединены некоторым путем. *Деревом* называется связный граф, не содержащий циклов, с выделенной вершиной, именуемой *корнем*.

Эволюционная модель в ТЭМ задается следующим гиперграфом $X = \{V, E', E''\}$.

Вершины, образующие множество V – это классы, порожденные от единого прокласса и соответствующие элементам модели. Любой класс имеет один и только один из трех указанных ниже типов $V = \{V_1, V_2, V_3\}$

Ребра, т.е. подмножества вершин, образующие семейство E' – это отношения (связи) перечисленных ниже семи видов между элементами модели. $E' = \{E_1, E_2, E_3, E_4, E_5, E_6, E_7\}$

Ребра, т.е. подмножества вершин, образующие семейство E'' – это множественные связи архитектуры гиперграфа, объединение элементов модели – классов, в группы – контейнеры по трем перечисленным ниже принципам. $E'' = \{E_8, E_9, E_{10}\}$

Типы классов

Вершины гиперграфа – классы, порожденные от единого прокласса и имеющие тип, задаваемый в ТЭМ первым, фундаментальным уровнем эволюционного моделирования $V = \{V_1, V_2, V_3\}$ (см. Рис. 10):

1. Класс Содержание (С).
2. Класс Форма (Ф).
3. Класс Поведение (П).

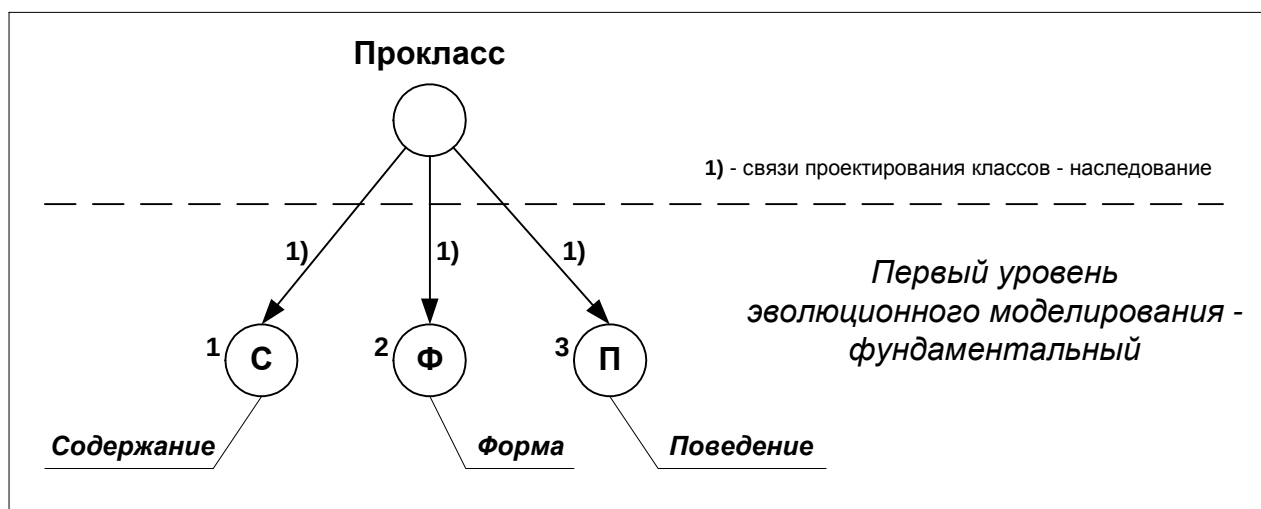


Рис. 10. Типы классов и связи проектирования

Архитектура гиперграфа классов, включающего структуру уровней наследования и соответствующие им базовые классы эволюционной модели, приведена в следующем разделе.

Виды отношений

Ребра гиперграфа – отношения (связи) классов в модели следующих видов:

Связи проектирования классов:

- 1) Наследование.

Связи структуризации классов (соединяют классы одного типа):

- 2) Связи классов типа содержание-содержание ($C \rightarrow C$).
- 3) Связи классов типа форма-форма ($\Phi \rightarrow \Phi$).
- 4) Связи классов типа поведение-поведение ($\Pi \rightarrow \Pi$).

Связи синтеза классов (соединяют классы различных типов):

- 5) Связи классов типа содержание-форма ($C \leftrightarrow \Phi$).
- 6) Связи классов типа содержание-поведение ($C \leftrightarrow \Pi$).
- 7) Связи классов типа форма-поведение ($\Phi \leftrightarrow \Pi$).

Связи структуризации архитектуры гиперграфа классов:

- 8) Контейнеры проектирования.
- 9) Контейнеры использования (Рабочие столы).
- 10) Контейнеры управления событиями.

На Рис. 11 представлены связи наследования, структуризации и синтеза классов.

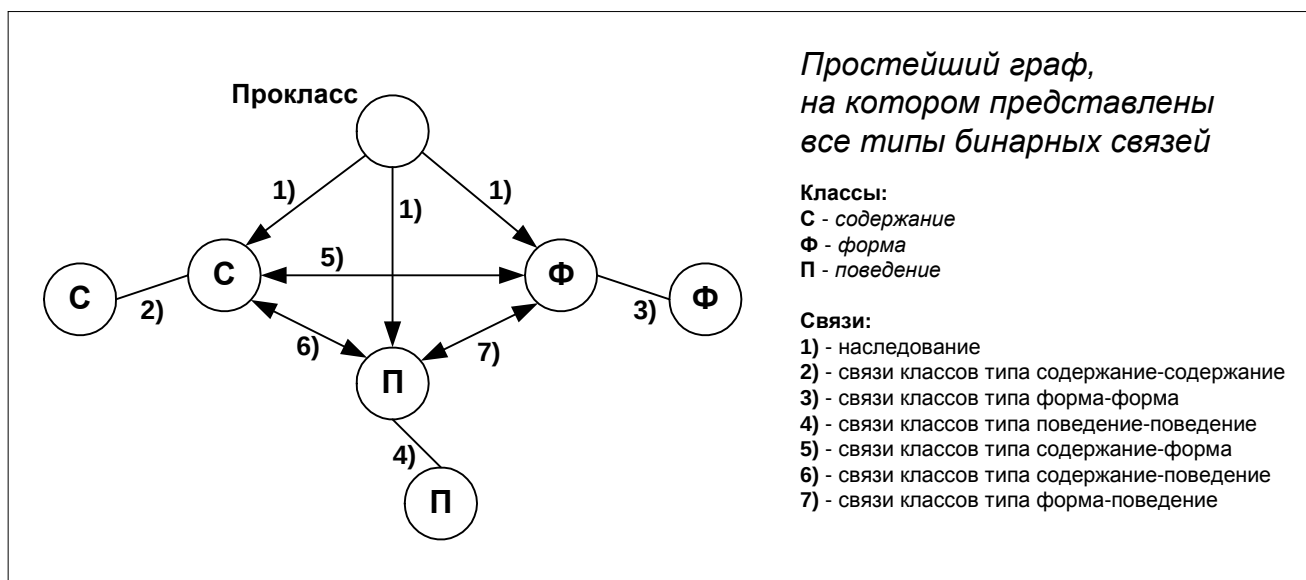


Рис. 11. Связи наследования, структуризации и синтеза классов

Моделирование в предлагаемом подходе заключается в эволюционном проектировании классов с использованием принципа наследования, реализующем связь вида 1), структуризацией их с использованием связей вида 2), 3), 4), с дальнейшим *синтезом* спроектированных классов, с использованием связей вида 5), 6), 7). Логическая навигация в получаемой модели, проектирование, анализ и ее исполнение осуществляются с использованием связей вида 8), 9), 10).

Гиперграф содержит следующий важный подграф. Если рассмотреть его вершины и для них только отношения наследования, то будет получено дерево наследования. Это дерево используется для эволюционного проектирования гиперграфа классов.

6.1. Классы гиперграфа

Класс задается совокупностью свойств (членов класса) – описаний некоторых *сущностей* (атрибутов) и *методов* их изменения. Формально – это множество, состоящее из пары подмножеств. Одно из них – подмножество *переменных*, другое – подмножество *методов* их изменения.

Проектирование классов базируется на следующих принципах объектно-ориентированного проектирования:

- **Инкапсуляция.** Объединение структур данных с функциями (методами), предназначенными для обработки этих данных.
- **Наследование.** Процесс создания нового класса на основе уже имеющегося. Наследование наделяет новый **производный** класс всей полнотой функциональных возможностей **класса-родителя**. Производный класс расширяет возможности класса-родителя для решения специфических задач.
- **Полиморфизм.** Полиморфизм (множественное поведение) позволяет на этапе выполнения выбирать нужную операцию в зависимости от класса объекта.

В модели-гиперграфе все классы принадлежат к одному из трех типов:

- *класс-содержание*,
- *класс-форма*,
- *класс-поведение*.

Класс-содержание – описание моделируемой предметной области в виде дискретных характеристик – метаданных для накопления и хранения информации об отдельных экземплярах и процессах функционирования этой предметной области.

Классы типа содержание отражают актуальное состояние модели, историю состояний и прогнозы развития.

Класс-форма – описание свойств формы представления информации для ввода, отображения хранимых и преобразованных данных об экземплярах и процессах моделируемой предметной области.

Классы типа форма определяют внешние интерфейсы – лицо (внешний образ) модели.

Класс-поведение – описание преобразований вводимой, хранящейся и отображаемой информации об экземплярах и процессах моделируемой предметной области.

Классы типа поведение описывают систему законов, по которым функционируют элементы модели.

При проектировании эволюционной модели созданные классы могут в дальнейшем быть использованы двояко: с одной стороны – как классы прототипы, с другой стороны – как классы, исполняемые и классы генерации. При этом все классы являются классами прототипами, и только часть из них является исполняемыми или классами генерации.

Классы прототипы – классы всех типов: *содержания, формы, поведения*, проектируемые в модели, в том числе создаваемые для удобства, прозрачности, увеличения скорости проектирования различных задач, имеющих сходные общие свойства. Они позволяют автоматически распространять изменения общих свойств во всех многочисленных наследниках, увеличивая эффективность, надежность проектирования, анализа и сопровождения эволюционирующей модели.

Классы исполняемые – классы типа *форма* или *поведение*, которые являются источниками порождения исполняемых объектов для формирования и отображения состояний и взаимодействий экземпляров предметной области.

Из исполняемых классов формируются виртуальные системы управления.

Классы генерации – классы типа *содержание*, которые являются источниками для формирования реальных информационных хранилищ воздействий и состояний – *памяти*.

Анализ исполняемых классов и классов генерации в модели позволяет выделять неиспользуемые (тупиковые) эволюционные ветви в проектируемой модели. Анализ использования объектов позволяет выявлять мутацию (в том числе деградацию) экземпляров предметной области.

6.2. Отношения классов в гиперграфе

Отношение – простая (бинарная), ориентированная или множественная связь между классами, отражающая различные принципы взаимодействия классов в моделях.

Виды отношений (связей) классов в модели:

Связи проектирования классов:

- 1) Наследование.

Связи структуризации классов:

- 2) Связи классов типа содержание-содержание ($C \rightarrow C$).
- 3) Связи классов типа форма-форма ($\Phi \rightarrow \Phi$).
- 4) Связи классов типа поведение-поведение ($\Pi \rightarrow \Pi$).

Связи синтеза классов:

- 5) Связи классов типа содержание-форма ($C \leftrightarrow \Phi$).
- 6) Связи классов типа содержание-поведение ($C \leftrightarrow \Pi$).
- 7) Связи классов типа форма-поведение ($\Phi \leftrightarrow \Pi$).

Связи структуризации архитектуры гиперграфа классов:

- 8) Контейнеры проектирования.
- 9) Контейнеры использования (Рабочие столы).
- 10) Контейнеры управления событиями.

Наследование – связи, устанавливаемые при проектировании нового класса и отражающие процесс нормализации свойств классов. Связь – бинарная, ориентированная.

При проектировании каждого нового класса всегда указывается класс-родитель (прототип), от которого генетически будут использоваться уже спроектированные свойства (переменные и методы).

При проектировании классов можно использовать *копирование* – создание копии класса с аналогичным предком, как частный случай наследования, при котором копирование класса A_2 (класс-родитель A_1) в класс A_3 означает формирование для класса A_3 связи наследования от класса A_1 и перенос всей совокупности добавленных свойств класса A_2 в класс A_3 .

Для повышения скорости и эффективности при эволюционном проектировании удобно использовать *композиционное наследование* и *композиционное копирование*.

Подграф, элементами которого являются все классы всех типов, а связи только вида - наследование, является иерархическим графом (граф без циклов) - деревом эволюции классов (деревом наследования).

Связи классов типа содержание-содержание отражают систематизацию и структуризацию атрибутов (переменных) и методов классов типа содержание. Они формируются процессами нормализации данных и реализацией, например, реляционных связей структурированной информации «один ко многим», «многие к одному», «многие ко многим».

Подграф, элементами которого являются классы типа содержание (классы генерации хранилищ), а связи только вида содержание-содержание, является подграфом структуры данных исполняемой модели.

Связи классов типа форма-форма отражают процессы систематизации и структуризации свойств классов по различным визуальным формам. Связи форм описываются событиями форм, результатом выполнения которых является инициализация других форм.

Событиями форм являются, например, следующие: открыть форму, закрыть форму, позиционироваться на элементах формы, определить состояние элементов формы, активизировать элементы формы и т.п.

Подграф, элементами которого являются исполняемые классы типа форма, и связи данного типа 3), является подграфом структуры оказываемых внешних воздействий и представления состояний исполняемой модели.

Связи классов типа поведение-поведение отражают зависимости, полученные в результате структуризации сложных алгоритмов по группам действий, описывающих законы поведения предметной области. Они

устанавливают логические, в том числе зависящие от времени, последовательности исполнения классов типа поведение.

Связи классов типов содержание-форма отражают возможности:

- единой сложной формой ввести или отобразить информацию нескольких независимых классов типа содержание,
- разграничить действие с одним классом типа содержание любым количеством независимых форм ввода и отображения информации.

Использование и анализ таких связей приводит к оптимизации и стандартизации интерфейсов исполняемой модели, формирует иные принципы реализации ряда задач, например, документооборота.

Связи классов типа содержание-поведение и форма-поведение отражают связи класса типа поведение с классами типа форма и содержание, являющимися либо источниками операндов, либо приемниками или формами представления результата действия алгоритма проектируемого процесса.

Контейнеры – средства диаграммного представления *архитектуры гиперграфа классов*, использующие различные принципы объединения классов в группы. Контейнеры используются для навигации по гиперграфу.

Контейнеры могут содержать как классы, так и другие однотипные контейнеры.

Контейнеры проектирования – объединение классов для эффективного и оптимального проектирования (см. Рис. 12).

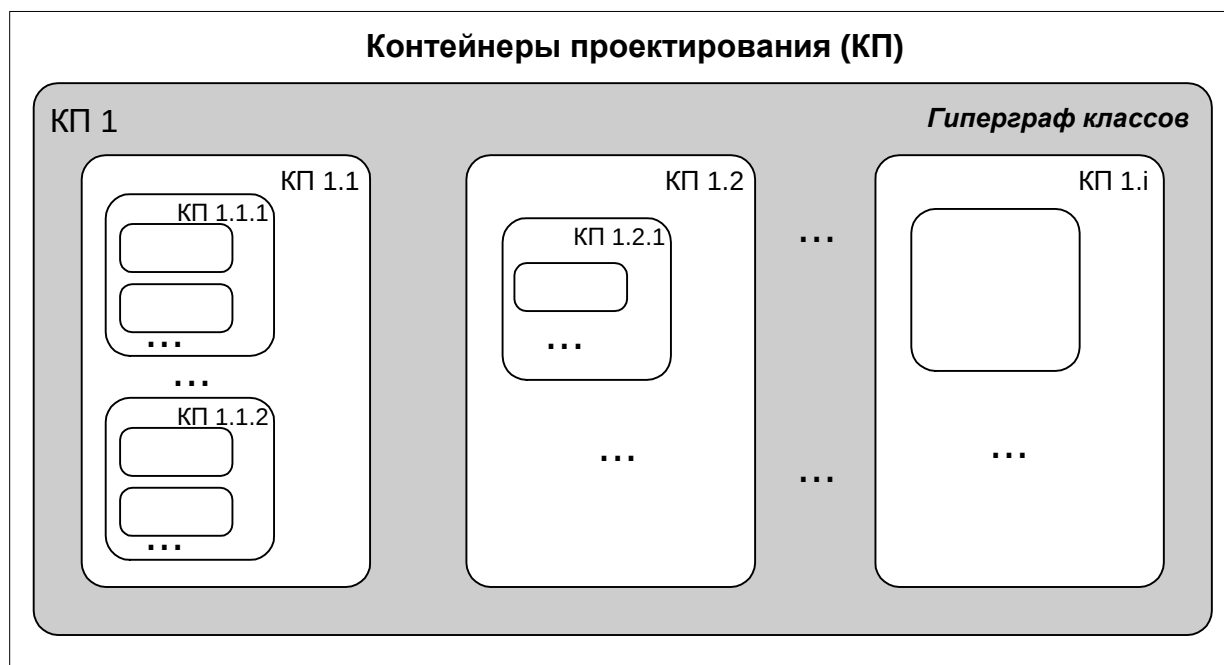


Рис. 12. Контейнеры проектирования

Контейнеры проектирования однократно содержат все созданные классы модели (прототипы, в том числе исполняемые классы и классы генерации).

Задачи выделения и формирования контейнеров проектирования:

- обеспечение прозрачности процессов эволюции классов при их наследовании;
- обеспечение однократности использования классов и их свойств (членов класса) при проектировании;
- визуальное объединение совокупности проектируемых классов всех типов (содержания, формы и поведения) в архитектурные уровни гиперграфа (см. следующий раздел).

Контейнеры использования (Рабочие столы) – объединение классов для эффективного и оптимального определения зоны использования (доступа и ответственности) конечным пользователем (см. Рис. 13).

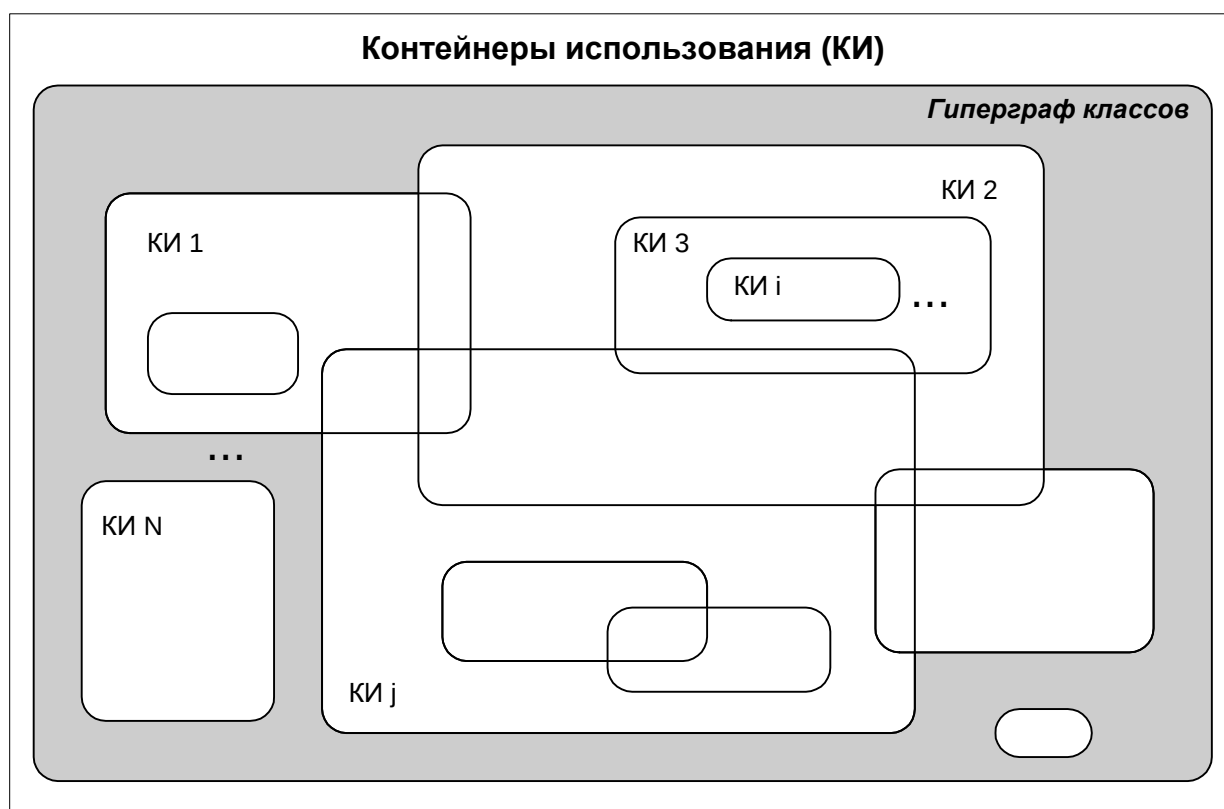


Рис. 13. Контейнеры использования

Контейнеры использования определяют виртуальное описание объектов и процессов управления систем управления.

Органам управления систем управления ставятся в соответствие (назначаются) контейнеры использования с регламентами и парольными технологиями доступа.

Контейнеры использования включают только исполняемые классы типа форма и поведение.

Принципы формирования контейнеров использования (рабочих столов):

- допускается многократное включение исполняемых классов в различные контейнеры использования;

- логический подход группирования классов по предметным областям и процессам жизни предметов и управления;
- обеспечение защиты, разграничения доступа, авторизации и контроля ответственности в использовании спроектированной модели.

Контейнеры событий – сетевое описание зависимости исполняемых классов и проекция их на временные интервалы (см. Рис. 14).

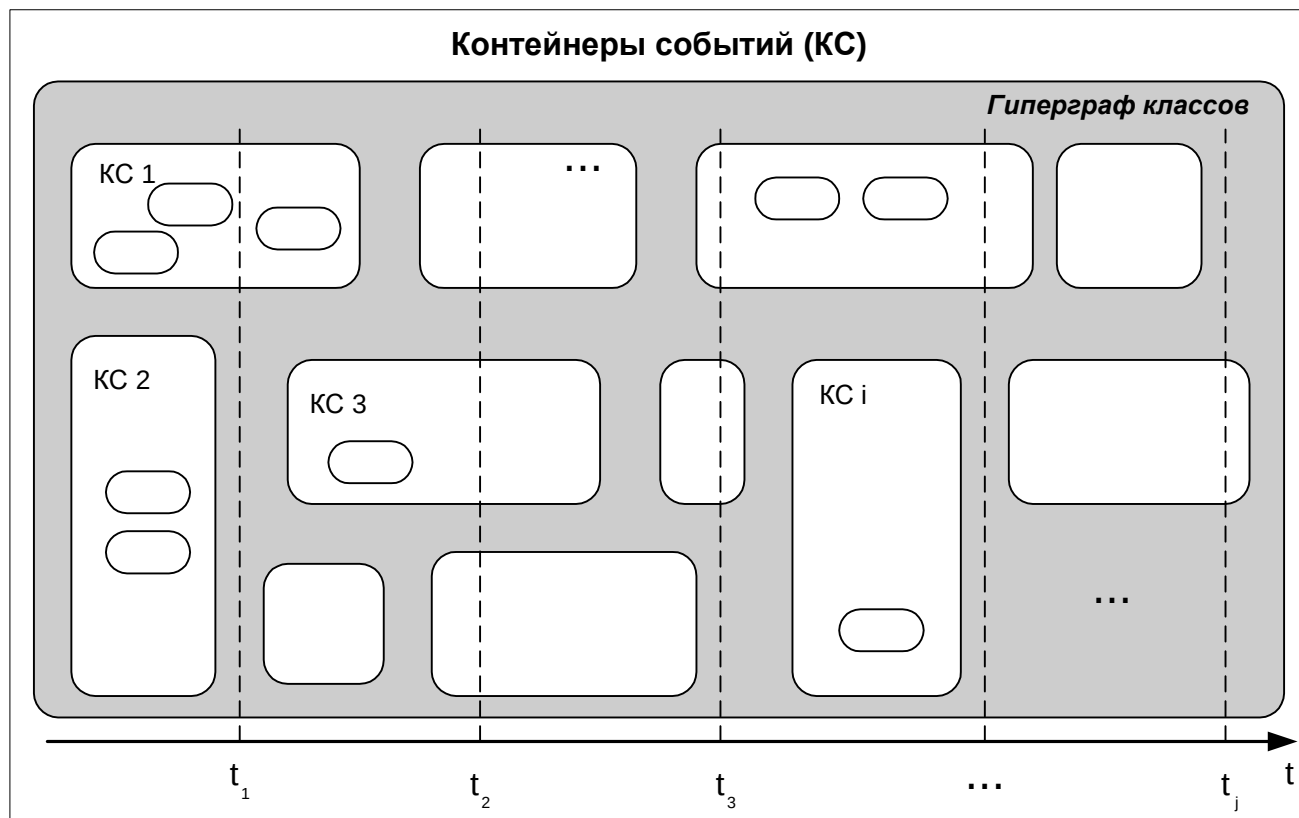


Рис. 14. Контейнеры событий

Контейнеры событий содержат только исполняемые классы. Исполняемые классы в таком контейнере используются однократно. Контейнер событий может включать другие контейнеры событий вложенного временного интервала.

Для описания временных зависимостей исполняемых классов в контейнерах событий используется описание связей структуризации и синтеза: связи классов типа содержание-содержание, связи классов типа форма-форма, связи классов типа поведение-поведение, связи классов типа содержание-форма, связи классов типа содержание-поведение, связи классов типа форма-поведение.

Контейнеры событий представляют в эволюционной модели, например, в финансово-экономической предметной области, спроектированные бизнес-процессы.

Кроме того, выбирая для многостороннего анализа модели и навигации различные совокупности групп типов элементов и видов связей, можно получать неограниченное множество диаграмм – подграфов.

7. Эволюционное моделирование

7.1. Архитектура эволюционной модели - гиперграфа классов

В ТЭМ моделью любой предметной области является гиперграф с элементами, описанными в предыдущих разделах – классами и отношениями.

Для проектирования классов и отношений в ТЭМ разработан единый эволюционный язык, частью которого они являются.

Одной из частей этого языка является единая архитектура модели – гиперграфа классов.

Архитектура модели содержит уровни и базовые классы уровней, являющиеся основой технологии описания предметов и процессов.

На основе единой архитектуры модели и базовых классов соответствующих уровней ведется *последовательное, многопользовательское, междисциплинарное, сетевое и распределенное эволюционное проектирование классов и отношений* моделируемой предметной области и процессов функционирования.

В ТЭМ предлагается следующий универсальный набор уровней метаклассов или типов классов:

Первый уровень «Фундаментальный»

Второй уровень «Информационно-технологический»

Третий уровень «Системно-ориентированный»

Четвертый уровень «Предметно-ориентированный»

Каждый уровень может содержать несколько последовательных связей типа наследование.

7.1.1. Фундаментальный уровень

Классы первого *фундаментального уровня* наследования созданы от единого прокласса и могут быть трех типов: **Содержание, Форма, Поведение.**

Определения классов этих типов даны в разделе 5.

В гиперграфе все классы имеют некоторую определенную характеристику, называемую типом - это либо содержание, либо форма, либо поведение.

Связи проектирования классов типа содержание, форма, поведение могут связывать только классы с именами $A_1/A_2/...$, у которых первое слово в имени класса A_1 одинаковое.

7.1.2. Информационно-технологический уровень

Информационно-технологический уровень наследования классов по существу используемых свойств и принципам их выделения зависит от уровня развития информационных технологий по хранению, представлению и обработке данных.

Класс *типа содержание* на современном информационно-технологическом уровне может быть представлен, например, реляционными базами данных, многомерными хранилищами и другими реализациями реального хранения данных.

Класс *типа форма* на современном информационно-технологическом уровне может быть представлен графическим, буквенно-цифровым, мультимедийным интерфейсом (сетевого или интернет-клиента) по вводу, корректировке и отображению данных (карточки, таблицы, документы, графики, геоинформационные системы, деревья, аудио, видео и т.п.).

Формы могут быть простыми и составными, синхронными и асинхронными, модалными и независимыми и т.д.

Класс *типа поведение* на современном информационно-технологическом уровне может быть представлен процедурами (в т.ч. хранимыми), схемами, комплексом процедур и т.п.

7.1.3. Системно-ориентированный уровень

Классы *системно-ориентированного* уровня наследования построены на основе следующих принципов:

- ✓ Непрерывность,
- ✓ Своевременность,
- ✓ Нейтральность,
- ✓ Преобладание сущности над формой,
- ✓ Баланс движения ресурсов (двойная транзакция),
- ✓ Сопоставимость,
- ✓ Понятность,
- ✓ и другие.

На основе этих принципов сформированы специальные свойства базовых классов системно-ориентированного уровня.

Например:

Непрерывность. Допущение непрерывности процессов жизни предметов проявляется в том, что модель строится в предположении, что нет намерения и необходимости прекращать или существенно сокращать их деятельность. Это допущение не противоречит возможности динамического изменения предметов и процессов с условием необходимости сохранения истории и

возможности в любой момент времени актуализировать прошлое состояние модели.

Своевременность. Этот принцип формирует требования к свойствам классов модели и правилам отражения временных характеристик прогнозируемых и фактических действий в реальной жизни и в исполняемой модели.

В ряде дальнейших работ будут описаны все принципы и свойства базовых классов системно-ориентированного уровня.

В ТЭМ на этом уровне выделены следующие **системно-ориентированные базовые классы**:

- 1) *Универсальный класс*,
- 2) *Классификаторы*,
- 3) *Нормативы*,
- 4) *Картотеки*,
- 5) *Регистраторы*,
- 6) *Движение состояний*,
- 7) *Балансы*,
- 8) *Прогнозы*,
- 9) *Показатели*.

Универсальный класс служит для обобщенного представления в модели непознанных на момент проектирования, неспецифицированных сущностей.

Классификатор - это глоссарий используемых понятий. Он представляет некоторый конечный язык описания экземпляров и процессов предметной области.

Классификаторы представляют собой набор словарей, в которых слова структурируются по сущностям предметной области. На основе свойств и данных классификаторов описываются другие классы и сведения об экземплярах предметной области, их состояние и процессы.

Особого внимания заслуживают словари синонимов, в том числе логических и фразеологических.

Норматив - наборы параметров, фиксированных на определенный период времени или другими условиями их актуальности.

Картотека - энциклопедия используемых сущностей рассматриваемой предметной области.

Регистратор - осуществляет регистрацию, учет и хранение информации о входных и выходных сигналах — информации, являющейся источником изменений состояния сущностей.

Движение состояний - осуществляет учет и хранение информации об изменениях состояний не аддитивных и не сбалансированных сущностей.

Баланс - служит для формирования и автоматической поддержки баланса аддитивных сущностей.

Прогноз - служит для формирования возможных состояний и анализа отклонений с помощью наборов математических методов предсказания и прогнозирования будущих изменений сигналов и состояний сущностей.

Показатели - интегральные характеристики, описывающие изменения сигналов и состояний сущностей.

Приведенные базовые классы могут задаваться как формальные языки.

Имя класса имеет следующий вид: $A_1/A_2/.../A_i/...$, где A_i одно из слов уровня наследования классов. В частности, A_1 – одно из трех слов (содержание, форма, поведение), обозначающих тип класса, A_2 – название одного из девяти базисных классов.

7.1.4. Предметно-ориентированный уровень

Классы предметно-ориентированного уровня создаются наследованием от классов системно-ориентированного уровня и содержат свойства, определенные в результате изучения, исследования и анализа предметов и процессов конкретной предметной области.

При рассмотрении и формализации предметной области необходимо использовать язык базовых классов третьего уровня архитектуры модели – системно-ориентированного уровня гиперграфа классов.

Первые три уровня базовых классов относительно фиксированы, однако, они могут быть дополнены свойствами и классами, полученными в результате нормализации свойств классов нижних уровней наследования при обобщении опыта моделирования, анализе спроектированных моделей.

Вся модель предметной области может быть также эволюционно оптимизирована в ходе проектирования реальных моделей, их использования, сопровождения, развития на основе нормализации идентичных свойств различных групп классов.

В ходе проектирования нескольких классов со схожими свойствами удобно использовать создание промежуточных классов прототипов для сокращения времени проектирования и дальнейшего развития модели.

Описание связей гиперграфа эволюционной модели предметной области формируется схематично в следующей последовательности: выбор контейнера проектирования, наследование классов, создание связей структуризации, формализация связей синтеза, контейнеризация для использования.

При описании в модели каждого нового свойства (переменных и методов) класса необходимо убедиться, что оно - новое и не содержится в автоматически формируемых библиотеках модели. Так же необходимо определить: в класс какого уровня наследования это новое свойство необходимо добавить для эффективного проектирования общих свойств.

При создании в модели каждого нового класса необходимо убедиться по дереву проекта в том, что такого класса еще нет, и подобрать близкого по свойствам родителя для минимизации процесса моделирования.

При эволюционном сетевом моделировании предметной области группа разработчиков модели легко ориентируется в состоянии проекта и использует совместные результаты проектирования в реальном времени.

7.2. Пример проектирования классов предметно-ориентированного уровня эволюционной модели

Одним из примеров эффективности использования теории эволюционного моделирования является решение многочисленных задач управления финансово-экономической деятельностью государственных органов и крупнейших предприятий и холдингов.

Например, для реального функционирования системы управления некоторым предприятием эволюционно проектируется модель этого предприятия, которая описывает:

- Информационные хранилища данных о состоянии ресурсов и деятельности (планируемой, оперативной, фактически совершенной...);
- Формы ввода, корректировки, отображения, анализа, печати, приема-передачи и т.п. данных о деятельности предприятия и окружающей среды;
- Законы, по которым ведется управленческий, бухгалтерский, налоговый учет; планирование, прогнозирование, бюджетирование; мониторинг состояния ресурсов и результатов деятельности; финансово-экономический анализ и т.п.

Модель каждого проекта насчитывает десятки тысяч классов предметно-ориентированного уровня и еще большее множество разнотипных связей.

Приведенный на Рис. 15 пример упрощен до ряда схематических элементов модели, но разных типов базовых классов гиперграфа.

Применяя принципы инкапсуляции и наследования, создаются новые предметно-ориентированные классы, т.е. элементы реальной модели, отражающие свойства и методы конкретных используемых на предприятии классификаторов, нормативов, картотек ресурсов, входящих/исходящих документов, балансовых счетов бухгалтерского учета и т.п.

На Рис. 15 изображен гиперграф классов со следующими условными обозначениями:

- | | |
|-----|-----------------------------------|
| ○ | — классы; |
| — | — связи - наследования; |
| --- | — связи структуризации и синтеза; |

Эволюционное моделирование гиперграфа классов

на примере ряда задач ERP-систем управления бизнесом предприятий

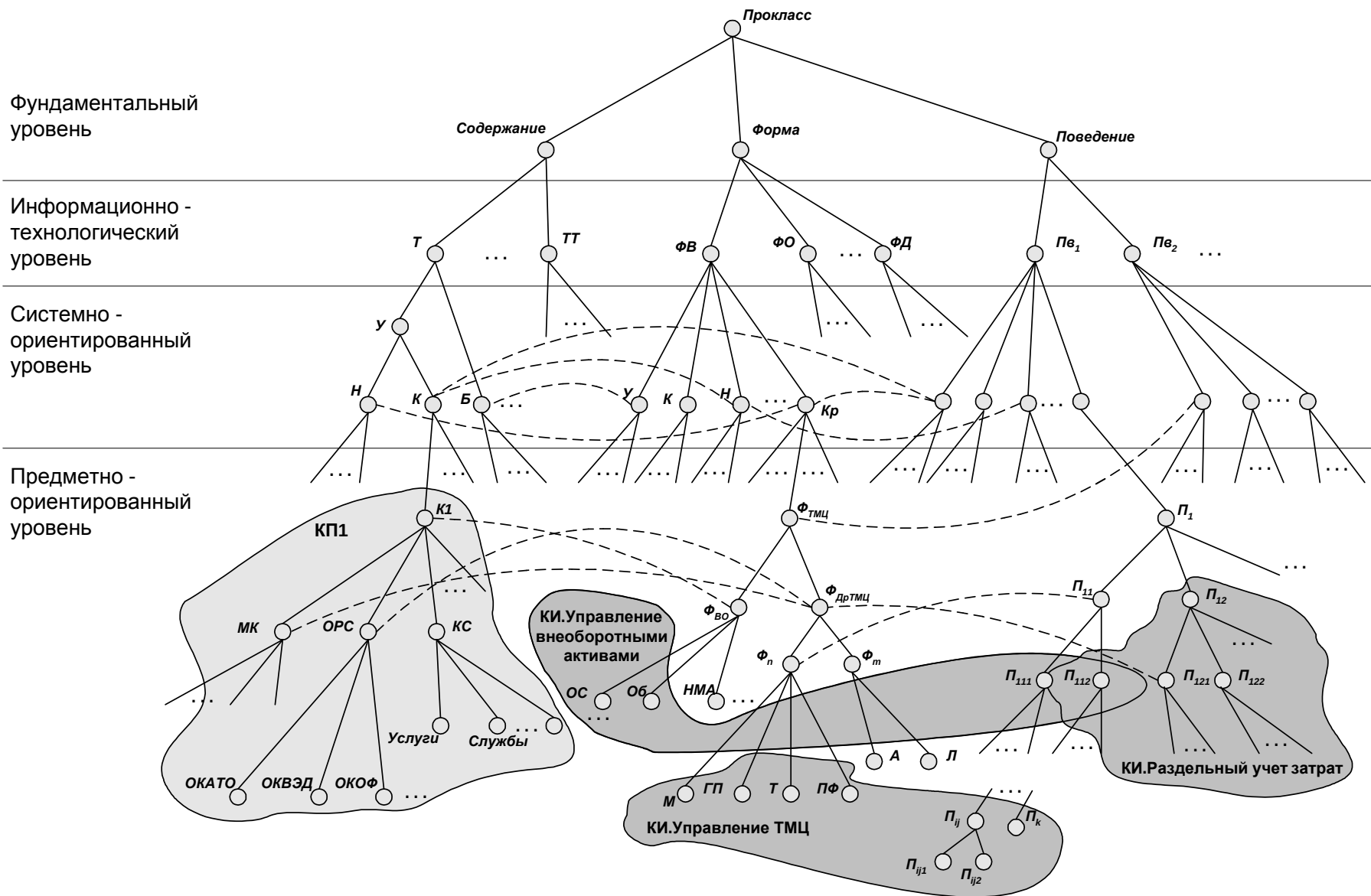


Рис. 15. Эволюционное моделирование гиперграфа классов



– контейнер проектирования;



– контейнер использования.

Последовательность вершин, связанных ребрами наследования, например, в контейнере проектирования КП1:

Прокласс – Содержание – Т – У – К – К1 – ОРС – ОКАТО –

демонстрирует последовательность проектирования в реальной модели Общероссийского классификатора, утвержденного Госкомстатом России – ОКАТО (Общероссийский классификатор объектов административно-территориального деления), который спроектирован от прототипа всех Общероссийских классификаторов (ОРС), который, в свою очередь, имеет родительским классом К1 (экономико-социальные классификаторы), отнаследованный от базового класса системно-ориентированного уровня К (классификаторы).

Аналогично можно проектировать различные формы ввода для картотек, например, товарно-материальных ценностей (ТМЦ):

Прокласс – Форма – ФВ (форма ввода) – Кр – $\Phi_{\text{ТМЦ}}$ – $\Phi_{\text{дрТМЦ}}$ – Φ_n – М

Форма ввода данных по материалам (М) имеет общего родителя Φ_n с формами ввода ГП (готовая продукция), Т (товары на складе), ПФ (полуфабрикаты), что позволяет увеличить скорость проектирования, однократно описав общие для этих задач свойства в классе Φ_n в части специфических свойств оборотных балансовых материальных ценностей. Остальные свойства пришли от класса $\Phi_{\text{дрТМЦ}}$ (другие ТМЦ), который имеет отличные от $\Phi_{\text{во}}$ (внеоборотные активы) свойства.

При таком эволюционном моделировании изменения, например, в налоговом законодательстве страны или в учетной политике предприятия в части оборотных материальных ценностей, учитываемых на балансе предприятия, влекут за собой однократное изменение свойств класса Φ_n с автоматическим изменением свойств всех отнаследованных классов, что и требовалось реализовать.

Если изменения необходимо провести во всех видах товарно-материальных ценностей – достаточно изменить свойства класса $\Phi_{\text{ТМЦ}}$.

На рисунке показано спроектированное рабочее место сотрудника КИ - Управление ТМЦ, обрабатывающего первичные документы по некоторой группе ТМЦ и использующего автоматические расчеты, например, класс Π_{ij1} (расчет средневзвешенных цен ТМЦ).

В ряде следующих работ примеры эволюционного проектирования моделей, их сопровождения и развития будут описаны более детально.

8. Основные преимущества эволюционного моделирования

Создание новой теории эволюционного моделирования основано на новой, качественно иной структуризации элементов языка базовых классов междисциплинарного общения в едином информационно-функциональном пространстве модели неограниченной предметной области.

В качестве аналога подхода приведем пример эволюции средств общения и взаимодействия людей. Одним из эффективных способов передачи информации от поколения к поколению являлась письменность.

В самом начале ее зарождения она представляла собой *произвольный рисунок* – отображение предметов и процессов с *бесконечностью интерпретаций* сообщения как в изображении, так и в понимании.

Формализация понятий, *однотипное упрощенное написание* сформировали *языки тысяч иероглифов*.

Но все, что мы пишем – мы же и говорим. *Однотипных звуков* оказалось гораздо меньше – *около 30*, соответствующих буквам различных алфавитов (кириллице, латинскому, греческому...). При этом сокращении элементов языка *увеличилась точность описания и однозначность понимания*.

И, наконец, *общемировой язык* вычислительных средств хранения и обработки любой информации насчитывает *два символа* – “1” и “0”.

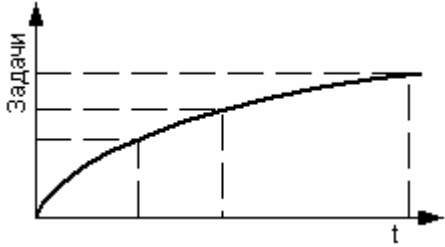
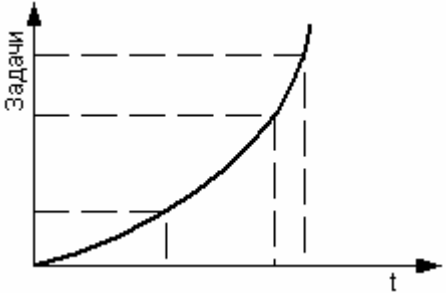
Итак, в предлагаемой теории эволюционного моделирования для построения модели сложных, открытых, динамических, слабо детерминированных систем осуществлен переход от традиционного редуцированного подхода к эволюционному проектированию и выделены базовые элементы языка описания единой модели – гиперграфа классов.

Тезисно сформулируем основные преимущества новой теории эволюционного моделирования:

<i>Традиционные подходы к проектированию моделей</i>	<i>Теория эволюционного моделирования</i>
Например, можно рассматривать ERP-системы управления деятельностью предприятия:	
SAP (R/3); Oracle (Oracle Application); Microsoft (MS Business Solution) и др.	Цефей (Эталон)
1) структурный подход к проектированию систем с множественными произвольными принципами структуризации предметной области	1) объектно-ориентированный подход к эволюционному проектированию единой модели предметной области, использование принципов наследования, инкапсуляции, полиморфизма

<i>Традиционные подходы к проектированию моделей</i>	<i>Теория эволюционного моделирования</i>
2) цели, задачи системы управления определяют структуру модели	2) единая модель предметной области используется для различных целей и задач
3) начальные основы моделирования: личностный опыт, накопленные заготовки, разрозненная информация об аналогах	3) начальные основы моделирования: базовые классы, технология, формализованные обобщенные библиотеки прототипов – реальных моделей различных разработчиков
4) организационно-технические методы использования обобщенного знания	4) единая сетевая среда эволюционного моделирования и язык базовых классов обобщения знаний
5) итерационное проектирование	5) эволюционное проектирование
6) многократная субъективная разрозненная интерпретация предметной области в различных системах	6) базовые классы, принцип наследования, единая модель предметной области в различных виртуальных системах использования
7) общий подход ко всем моделируемым процессам взаимодействия объектов управления	7) моделируемые процессы взаимодействия делятся на: - процессы жизни предметов, - процессы управления
8) моделирование сложных систем реализуется последовательной технологией многоуровневой декомпозиции, усреднения и упрощения для выделения ряда существенных (ключевых, с наибольшим весом...) характеристик и выявления основных законов зависимостей	8) не производится
9) интерфейсы взаимодействия составных частей	9) не проектируются
10) сетевые непредсказуемые реакции необходимых изменений при изменениях в структуре отношений и/или элементах одной из частей - модулей	10) эволюционное развитие
11) нет единой модели системы, набор подсистем, каждой из которых может соответствовать множество моделей: организационных, технологических, функциональных, событийных, структурных, регламентных, информационных, документооборота,	11) единая модель – гиперграф классов, где можно выделить любые элементы и подграфы

<i>Традиционные подходы к проектированию моделей</i>	<i>Теория эволюционного моделирования</i>
ресурсных, процессных, классов, прецедентов и т.д., а также совокупности различных комбинаций принципов создания	
12) отсутствует единое пространство описания системы, модели (в лучшем случае может быть единое описание структуры данных)	12) единое информационно-функциональное пространство - репозиторий всей модели (всех элементов, всех отношений)
13) не возможно	13) реализация, например, бизнес-приложений и ERP (Enterprise Resource Planning)-систем в единой виртуальной модели управления позволяет увидеть всю архитектуру управления, рассмотреть различные варианты и обоснованность реструктуризации, создать технологию осознанного реинжиниринга, гармонизировать методы централизации и децентрализации управления, минимизировать потоки данных, оптимизировать организационные структуры, повысить эффективность управления ресурсами, сбалансировать цели и методы управления и т.п. Единая модель состояний, воздействий и восприятий
14) интеграция подсистем, необъяснимые «синергетические» эффекты	14) интеграция не нужна
15) возможность бесконтрольного многократного параллельного формирования несопоставимых и неоднозначных описаний одних и тех же сущностей и процессов, либо повторений, либо под одним и тем же именем могут быть описаны различные понятия	15) единая операционная сетевая среда и язык обобщения знаний, автоматические библиотеки используемых словарей, анализ избыточности, нормализация свойств классов

Традиционные подходы к проектированию моделей	Теория эволюционного моделирования
16) проектирование развития системы, модели для каждой новой задачи предметной области занимает все больше времени  $t = t_i + \sum_{i=1}^n t_i'$ – время создания новой задачи; t_i – время создания дополнительного модуля; $\sum_{i=1}^n t_i'$ – время интеграции с n связанными модулями	16) единая библиотека моделей, гиперграфов. К имеющемуся гиперграфу - модели лишь эволюционно добавляются новые элементы, свойства, связи  t_2 – время создания новой задачи (свойств и методов)
Этот анализ взят из отчета результатов экспертизы Market-Visio Gartner Group	
17) системы управления бизнесом, например, транснациональной компании с диверсифицированными видами деятельности насчитывает сотни тысяч элементов: десятки тысяч атрибутов, тысячи таблиц, десятки тысяч форм ввода, отчетов, документов, тысячи методов и процедур обработки данных, хранят и обрабатывают информацию об экземплярах предметной области в объеме миллионов записей. Кол-во разработчиков – около 2000 чел. Время разработки <i>модулей</i> – годы. Программный код – около 100 Гб	Кол-во разработчиков – 100 чел. Время разработки <i>моделей</i> – недели. Программный код – около 100 Мб
18) нет четкой сопоставимости понятий «предмет», «система», «модель»	18) предметная область проектируется в единой модели-гиперграфе классов. Виртуальные системы управления представляют наборы подграфов единого гиперграфа – модели предметной области. При этом предмет может состоять из других предметов. Системы управления могут быть совокупностью элементов модели и других систем. В модели-гиперграфе классов можно выделить любое множество подграфов

<i>Традиционные подходы к проектированию моделей</i>	<i>Теория эволюционного моделирования</i>
19) программные приложения конечного пользователя – модули программы сформированы изначально и однократно жестко разработчиками на основе их понимания задач	19) формирование эффективных оптимальных виртуальных приложений – динамически настраиваемых рабочих мест доступа (desktops) конечного пользователя в процессе эксплуатации
20) организационно-технические методы проектирования внешних интерфейсов в общих стандартах для различных систем	20) эволюционное моделирование для всей сложной модели создает и поддерживает стандарты внешних интерфейсов, что многократно повышает системность, скорость их проектирования, обучаемость, взаимозаменяемость пользователей, прозрачность, простоту и т.п.
21) бесконечное множество неформализованных элементов - интерпретации моделей	21) гиперграф классов – модель проектируется на основе 3-х типов базовых классов, 7-ми видов бинарных связей, 3-х видов множественных связей
и многие другие	

9. Заключение

В настоящей работе описаны концептуальные основы теории эволюционного моделирования, даны ее ключевые понятия и принципы. Однако, это только первые шаги по описанию существа новой теории.

Применяя около десяти лет методы эволюционного моделирования, единый язык базовых классов, реализованы и эффективно используются сотни уникальных проектов.

Сформирована обширная библиотека реальных динамически развивающихся моделей – семейства гиперграфов.

Вот только ряд проектов и реализованных моделей:

- Система мониторинга и анализа финансово-хозяйственной деятельности предприятий государственного сектора экономики РФ (ФЦП «Электронная Россия»);
- Система мониторинга и анализа платежеспособности стратегических предприятий РФ (ФЦП «Электронная Россия»);
- Концепция единой информационно-аналитической системы Федеральной Энергетической Комиссии РФ;
- Управление расчетами за услуги по транспортировке и компаундированию нефти ОАО Акционерная компания по транспорту нефти «Транснефть»;
- Корпоративные системы управления бизнесом ОАО «Башинформсвязь», ОАО «Тулетеком», ООО Компания «Межрегионгаз», ООО «Московская областная региональная компания по реализации газа», Российское внешнеэкономическое объединение «Зарубежнефть», Воскресенский государственный казенный агрегатный завод, ОАО «Кольская горнометаллургическая компания», ЗАО «Завод по ремонту электроподвижного состава», ООО «Лукойл – Западная Сибирь», ООО «Лотэко» и многие другие.

Заслуживают особого внимания и научного интереса перспективы развития и совершенствования следующих направлений:

✓ Среда эволюционного моделирования

Развитие Эталон-решений проектирования эволюционных моделей.

Дальнейшая формализация единого языка, базовых классов.

Формирование графических средств представления масштабных моделей и легкости многоцелевой навигации в них...

И т.п.

✓ Оптимизация моделей-гиперграфов классов

Совершенствование имеющихся и создание новых методов и технологий исследования и анализа моделей – гиперграфов классов.

Дальнейшее развитие и формализация свойств базовых классов.

Развитие глобальных библиотек универсальных прототипов эффективных и эволюционно развивающихся моделей управления...

И т.п.

Теория эволюционного моделирования может быть эффективно использована для создания, исследования, анализа, прогнозирования модели различных предметов и процессов.

Например:

✓ **Создание эволюционной модели баланса глобальной экономики**

Обоснование прогноза глобального развития: «будущее жителя Земли без государств и без налогов».

✓ **Создание эволюционной модели инфраструктуры жителей Земли**

✓ **Создание эволюционной модели библиотеки национальных налоговых законодательств**

Теория эволюционного моделирования является основой для эффективного анализа и гармонизации возникающих дисбалансов в целях и методах их реализации. Приоритеты декларируемых, контролируемых, анализируемых целей балансируются мониторингом «чудодейственных источников» их достижения.

Теория эволюционного моделирования преобразует подходы к реализации управления деятельностью в различных предметных областях.

Например:

Законодательная деятельность.

Законодатели всех стран смогут работать с единой эволюционной моделью какого-либо закона. Законотворческий процесс будет представлять собой работу над частью гиперграфа с однозначным формальным описанием понятий, сущностей, алгоритмов и т.п. рассматриваемого закона. При этом можно провести прогноз правоприменения с анализом последствий и противоречий.

Уже сейчас в Эталон-модели представлены основы Трудового кодекса РФ, Гражданского кодекса, Налогового кодекса, Положений о Бухгалтерском учете, Международных стандартов финансовой отчетности (МСФО) и т.д.

Причем, представлены не тексты, а исполняемая модель.

Теория эволюционного моделирования, несомненно, повлияет на принципы развития вычислительных средств, использование сетевых ресурсов, средств разработки, систем управления базами данных и т.п.

Теория эволюционного моделирования кардинально снижает издержки на создание эффективных систем управления.

Эволюционное моделирование – приоритет развития информационных технологий.

10. Литература

1. *Хохлова М.Н.* Свидетельство об официальной регистрации финансово-экономического комплекса «Эталон». РосАПО, №950380, 1995.
2. *Эталон.* Информационная система управления ресурсами предприятия. Руководство системного аналитика. М.: Компания «Цефей», 1998.
3. *Цефей.* Эталон. (Выставочный буклет). М.: Компания «Цефей», 2001.
4. *Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б.* Практическое моделирование динамических систем. СПб.: BHV, 2002.
5. *Владимиров В.С., Волович И.В., Зеленов Е.И.* р-Адический анализ и математическая физика. М.: Наука, 1994.
6. *Гаврилова Т.А., Хорошевский В.Ф.* Базы знаний интеллектуальных систем. СПб.: Питер, 2001.
7. *Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001.
8. *Гасанов Э.Э., Кудрявцев В.Б.* Теория хранения и поиска информации. М.: Физматлит, 2002.
9. *Глушков В.М.* Синтез цифровых автоматов. М.: Физматгиз, 1962.
10. *Кватрани Т.* Rational Rose 2000 и UML. Визуальное моделирование. М.: ДМК Пресс, 2001.
11. *Краснощечков П.С., Петров А.А.* Принципы построения моделей. М.: Фазис, 2000.
12. *Марка Д.А., МакГоуен К.* Методология структурного анализа и проектирования. М.: МетаТехнология, 1993.
13. *Поспелов И.Г.* Моделирование экономических структур. М.: Фазис, 2003.
14. *Рамбо Дж., Якобсон А., Буч Г.* UML. Специальный справочник. СПб.: Питер, 2002.
15. *Хопкрофт Д., Мотвани Р., Ульман Д.* Введение в теорию автоматов, языков и вычислений. М.: Изд. дом «Вильямс», 2002.
16. *Элиенс А.* Принципы объектно-ориентированной разработки программ. М.: Изд. дом «Вильямс», 2002.
17. *Яблонский С.В.* Основные понятия кибернетики – в кн. Проблемы кибернетики, вып. 2. М.: Наука, 1959, с.7-38.
18. *G. Booch.* Object-oriented development, IEEE Transaction.
19. *OMG Unified Modeling Language Specification. Version 1.4. September 2001.* <http://cgi.omg.org/docs/formal>

20. *Simons A. J. H., Graham I.* «37 Things that don't work in object modelling with UML», Proc. 2nd. ECOOP Workshop on Precise Behavioural Semantics, pub. TUM-I9813, eds. H Kilov and B Rumpe (Brussels : TU Munich, 1998), 209-232.
21. The precise UML group. <http://www.cs.york.ac.uk/puml/maindetails.html>
on Software Engineering, 12(2), 211-221 (1986).
22. *Волович И.В., Хохлова М.Н.* О теории моделирования и гиперграфе классов. Труды Математического института им. В.А.Стеклова, т. 245, с. 281-287, 2004 г.