

Лекция 4 . Системный подход к проектированию информационных систем.

ОСНОВНЫЕ ПОНЯТИЯ

Система – это совокупность связанных между собой из внешней среды элементов или частей, функционирование которых направленно на получение конкретного полезного результата.

Для системы характерны свойства:

- сложность;
- делимость;
- целостность;
- многообразии элементов и различие их природы;
- структурированность.

Система с управлением включает 3 подсистемы:

- управляющую систему;
- объект управления;
- систему связи.

Управляющая система совместно с системой связи образует систему управления.

Основные группы функций системы управления.

1. Функция принятия решений – это функция преобразования содержания информации.
2. Рутинные функции обработки информации.
3. Функции обмена информацией.

ОСНОВНЫЕ ПОНЯТИЯ. ПРОДОЛЖЕНИЕ

Системный анализ – это логически связанная совокупность теоретических и эмпирических положений в области математики, естественных наук и опыта разработки сложных систем, обеспечивающая повышение обоснованности решения конкретной проблемы.

Системный анализ – это методология решения проблем основанная на структуризации систем и количественном сравнении альтернатив. В состав задач системного анализа в процессе создания информационных систем входят задачи декомпозиции, анализа и синтеза.

Системный подход включает в себя выявление структуры системы, типизацию связей, определение атрибутов, анализ влияния внешней среды.

Основной общий принцип системного подхода заключается в рассмотрении частей явления или сложной системы с учетом их взаимодействия.

Системный подход рассматривают как направление научного познания и социальной политики.

Теория систем - дисциплина, в которой конкретизируются положения системного подхода; она посвящена исследованию и проектированию сложных экономических, социальных, технических систем, чаще всего слабоструктурированных.

В технике дисциплину, в которой исследуются сложные технические системы, их проектирование и которая аналогична теории систем, чаще называют системотехникой (не путать со схемотехникой!).

ПОНЯТИЕ СЛОЖНОЙ СИСТЕМЫ

Сложная система – система, характеризующаяся большим числом элементов и большим числом взаимосвязей элементов. Сложность системы определяется также видом взаимосвязей элементов, свойствами целенаправленности, целостности, членимости, иерархичности, многоаспектности.

Элемент – такая часть системы, представление о которой нецелесообразно подвергать при проектировании дальнейшему членению.

Подсистема – часть системы (подмножество элементов и их взаимосвязей), которая имеет свойства системы.

Надсистема – система, по отношению к которой рассматриваемая система является подсистемой.

Структура – отображение совокупности элементов системы и их взаимосвязей; понятие структуры отличается от понятия самой системы также тем, что при описании структуры принимают во внимание лишь типы элементов и связей без конкретизации значений их параметров.

Параметр – величина, выражающая свойство или системы, или ее части, или влияющей на систему среды. Обычно в моделях систем в качестве параметров рассматривают величины, не изменяющиеся в процессе исследования системы. Параметры подразделяют на внешние, внутренние и выходные, выражающие свойства элементов системы, самой системы, внешней среды соответственно.

ХАРАКТЕРИСТИКИ СЛОЖНОЙ СИСТЕМЫ

- **Целенаправленность** – свойство искусственной системы, выражающее назначение системы. Это свойство необходимо для оценки эффективности вариантов системы.
- **Целостность** — свойство системы, характеризующее взаимосвязанность элементов и наличие зависимости выходных параметров от параметров элементов, при этом большинство выходных параметров не является простым повторением или суммой параметров элементов.
- **Иерархичность** — свойство сложной системы, выражающее возможность и целесообразность ее иерархического описания, т. е. представления в виде нескольких уровней, между компонентами которых имеются отношения целое — часть.

ПРИНЦИПЫ И ЗАДАЧИ ПРОЕКТИРОВАНИЯ

Проектирование информационной системы связано с созданием, приобретением и предоставлением принятой форме образа этой системы. Данный образ объекта (информационной системы) может создаваться в процессе взаимодействия человека и ЭВМ.

Проектирование - это процесс заключающегося преобразования исходного описания объекта в окончательное описание объекта на основе выполнения комплекса работ исследовательского, расчётного и конструктивного характера.

ОСНОВНЫЕ ПОДХОДЫ К ПРОЕКТИРОВАНИЮ

Выделяют следующие подходы к проектированию:

- структурный;
- блочно-иерархический;
- объектно-ориентированный.

При **структурном подходе**, как разновидности системного, требуется синтезировать варианты системы из компонентов (блоков) и оценивать варианты при их частичном переборе с предварительным прогнозированием характеристик компонентов.

Блочно-иерархический подход к проектированию использует идеи декомпозиции сложных описаний объектов и соответственно средств их создания на иерархические уровни и аспекты, вводит понятие стиля проектирования (восходящее и нисходящее), устанавливает связь между параметрами соседних иерархических уровней.

Объектно-ориентированный подход к проектированию имеет ряд важных структурных принципов, используемых при разработке информационных систем, и преимуществ в решении проблем управления сложностью и интеграции ПО.

ПРЕИМУЩЕСТВА ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА

1. Вносит в модели приложений большую структурную определенность, распределяя представленные в приложении данные и процедуры между классами объектов.
2. Сокращает объем спецификаций благодаря введению в описания иерархии объектов и отношений наследования между свойствами объектов разных уровней иерархии.
3. Уменьшает вероятность искажения данных вследствие ошибочных действий за счет ограничения доступа к определенным категориям данных в объектах.

Описание в каждом классе объектов допустимых обращений к ним и принятых форматов сообщений облегчает согласование и интеграцию ПО.

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

1. Структуризация процесса проектирования, выражаемая декомпозицией проектных задач и документации, выделением стадий, этапов, проектных процедур. Эта структуризация является сущностью блочно-иерархического подхода к проектированию.
2. Итерационный характер проектирования.
3. Типизация и унификация проектных решений и средств проектирования.

Пример сложной системы.

Компьютер является *сложной системой* в силу наличия у него большого числа элементов, разнообразных связей между элементами и подсистемами, свойств целенаправленности, целостности, иерархичности. К *подсистемам* компьютера относятся процессор (процессоры), оперативная память, кэш-память, шины, устройства ввода/вывода.

В качестве *надсистем* могут выступить вычислительная сеть, автоматизированная и (или) организационная система, к которым принадлежит компьютер. *Внутренние параметры* - времена выполнения арифметических операций, чтения (записи) в накопителях, пропускная способность шин и др. *Выходные параметры* - производительность компьютера, емкость оперативной и внешней памяти, себестоимость, время наработки на отказ и др. *Внешние параметры* — напряжение питания сети и его стабильность, температура окружающей среды и др.

ОСНОВНЫЕ ЗАДАЧИ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ

Целью методологии проектирования ИС является регламентация процесса проектирования ИС и обеспечении управления этим процессом с тем, чтобы гарантировать выполнение требований как к самой ИС, так и к характеристикам процесса разработки.

Основными задачами, решению которых должна способствовать методология проектирования корпоративных ИС, являются следующие:

- обеспечивать создание корпоративных ИС, отвечающих целям и задачам организации, а также предъявляемым требованиям по автоматизации деловых процессов заказчика;
- гарантировать создание системы с заданным качеством в заданные сроки и в рамках установленного бюджета проекта;
- поддерживать удобную дисциплину сопровождения, модификации и наращивания системы;
- обеспечивать преемственность разработки, т.е. использование в разрабатываемой ИС существующей информационной инфраструктуры организации (задела в области информационных технологий).

Внедрение методологии должно приводить к снижению сложности процесса создания ИС за счет полного и точного описания этого процесса, а также применения современных методов и технологий создания ИС на всем жизненном цикле ИС - от замысла до реализации.

ОСНОВНЫЕ ОБЛАСТИ И ЦЕЛИ ПРОЕКТИРОВАНИЯ ИС

Проектирование ИС охватывает три основные области:

1. проектирование объектов данных, которые будут реализованы в базе данных;
2. проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
3. учет конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределенной обработки данных и т.п.

Проектирование информационных систем всегда начинается с определения цели проекта. В общем виде цель проекта можно определить как решение ряда взаимосвязанных задач, включающих в себя обеспечение на момент запуска системы и в течение всего времени ее эксплуатации:

- требуемой функциональности системы и уровня ее адаптивности к изменяющимся условиям функционирования;
- требуемой пропускной способности системы;
- требуемого времени реакции системы на запрос;
- безотказной работы системы;
- необходимого уровня безопасности;
- простоты эксплуатации и поддержки системы.

ОСНОВНЫЕ КЛАССЫ СТАНДАРТОВ ПРОЕКТИРОВАНИЯ ИС

Существует несколько классов стандартов, определяющих основные (средства) требования к технологии проектирования:

1. стандарт проектирования процесса;
2. стандарт проектирования документации;
3. стандарт проектирования интерфейсов предполагает, что информация выводится на экран, удовлетворяя эргономические характеристики, т.е. требования психологического и физического удобства работы с системой.

ОСНОВЫ ПРЕДПРОЕКТНОГО АНАЛИЗА ИС

Начальным этапом процесса создания ИС является моделирование бизнес-процессов, протекающих в организации и реализующих ее цели и задачи.

Модель организации, описанная в терминах бизнес-процессов и бизнес-функций, позволяет сформулировать основные **требования к ИС**. Это фундаментальное положение методологии обеспечивает объективность в выработке требований к проектированию системы. Множество моделей описания требований к ИС затем преобразуется в систему моделей, описывающих **концептуальный проект ИС**. Формируются модели архитектуры ИС, требований к программному обеспечению (ПО) и информационному обеспечению (ИО). Затем формируется архитектура ПО и ИО, выделяются корпоративные БД и отдельные приложения, формируются модели требований к приложениям и проводится их разработка, тестирование и интеграция.

Целью начальных этапов создания ИС, выполняемых на стадии анализа деятельности организации, является **формирование требований к ИС**, корректно и точно отражающих цели и задачи организации-заказчика.

На стадии предпроектного анализа формируются основные ориентиры будущего проекта:

1. принимаются различные решения по структуре и принципам функционирования системы;
2. формируются требования к обеспеченности ресурсами по взаимодействию с внешней средой;
3. определяются наиболее значимые цели организационной системы и уровни их достижимости, исходя их реальных ограничений на ресурсы.

СТАНДАРТЫ, РЕГЛАМЕНТИРУЮЩИЕ ЖИЗНЕННЫЙ ЦИКЛ ПО ИС

Среди наиболее известных стандартов можно выделить следующие:

- ГОСТ 34.601-90 - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания.
- ISO/IEC 12207:1995 - стандарт на процессы и организацию жизненного цикла. Распространяется на все виды заказного ПО.
- Custom Development Method (методика Oracle) по разработке прикладных информационных систем.
- Rational Unified Process (RUP) предлагает итеративную модель разработки, включающую четыре фазы: начальная стадия (inception), стадия разработки (elaboration), стадия конструирования (construction), стадия ввода в действие (transition).
- Microsoft Solution Framework (MSF) сходна с RUP, так же включает четыре фазы: анализ, проектирование, разработка, стабилизация, является итерационной, предполагает использование объектно-ориентированного моделирования.
- Extreme Programming (XP). Экстремальное программирование (самая новая среди рассматриваемых методологий) сформировалось в 1996 году.

СТАДИИ И ЭТАПЫ ПРОЦЕССА КАНОНИЧЕСКОГО ПРОЕКТИРОВАНИЯ ИС ПО ГОСТ 34.601-90

Организация канонического проектирования ИС ориентирована на использование главным образом **каскадной модели жизненного цикла (ЖЦ)** ИС.

Стадии и этапы работы описаны в стандарте ГОСТ 34.601-90.

В зависимости от сложности объекта автоматизации и набора задач, требующих решения при создании конкретной ИС, стадии и этапы работ могут иметь различную трудоемкость. Допускается объединять последовательные этапы и даже исключать некоторые из них. Допускается также начинать выполнение работ следующей стадии до окончания предыдущей.

Проектирование как процесс развивается во времени, расчленяется на стадии, этапы, проектные процедуры и операции.

При проектировании сложных систем выделяют 8 **стадий**, каждая из которой может включать в себя несколько этапов.

СТАДИИ И ЭТАПЫ ПРОЦЕССА КАНОНИЧЕСКОГО ПРОЕКТИРОВАНИЯ ИС ПО ГОСТ 34.601-90

Стадия 1. Формирование требований к ИС.

- обследование объекта и обоснование необходимости создания ИС;
- формирование требований пользователей к ИС;
- оформление отчета о выполненной работе и тактико-технического задания на разработку.

Стадия 2. Разработка концепции ИС.

- изучение объекта автоматизации;
- проведение необходимых научно-исследовательских работ;
- разработка вариантов концепции ИС, удовлетворяющих требованиям пользователей;
- оформление отчета и утверждение концепции.

Стадия 3. Техническое задание.

- разработка и утверждение технического задания на создание ИС.

Стадия 4. Эскизный проект.

- разработка предварительных проектных решений по системе и ее частям;
- разработка эскизной документации на ИС и ее части.

СТАДИИ И ЭТАПЫ ПРОЦЕССА КАНОНИЧЕСКОГО ПРОЕКТИРОВАНИЯ ИС ПО ГОСТ 34.601-90

Стадия 5. Технический проект.

- разработка проектных решений по системе и ее частям;
- разработка документации на ИС и ее части;
- разработка и оформление документации на поставку комплектующих изделий;
- разработка заданий на проектирование в смежных частях проекта.

Стадия 6. Рабочая документация.

- разработка рабочей документации на ИС и ее части;
- разработка и адаптация программ.

СТАДИИ И ЭТАПЫ ПРОЦЕССА КАНОНИЧЕСКОГО ПРОЕКТИРОВАНИЯ ИС ПО ГОСТ 34.601-90

Стадия 7. Ввод в действие.

- подготовка объекта автоматизации;
- подготовка персонала;
- комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);
- строительно-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний ;
- проведение опытной эксплуатации ;
- проведение приемочных испытаний.

Стадия 8. Сопровождение ИС.

- выполнение работ в соответствии с гарантийными обязательствами;
- послегарантийное обслуживание.

ПРОЦЕССЫ ЖИЗНЕННОГО ЦИКЛА СОЗДАНИЯ ПО ИС СОГЛАСНО СТАНДАРТА ISO/IEC 12207

В соответствии со стандартом ISO/IEC №12207 все процессы жизненного цикла разделены на 3 группы:

1. основные процессы;
2. вспомогательные процессы;
3. организационные процессы.

Основные процессы: приобретение, поставка, разработка, эксплуатация, сопровождение.

Вспомогательные процессы: документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем.

Организационные процессы: управление, создание инфраструктуры, усовершенствование, обучение.

Стандарт не предлагает обязательного использования конкретной модели жизненного цикла ПО. Стандартом предлагается некоторый **базовый набор аспектов рассмотрения ЖЦ** процессов с различных точек зрения.

Таковыми аспектами являются: договорной аспект, аспект управления, аспект эксплуатации, инженерный аспект, аспект поддержки.

СТАДИИ ЖИЗНЕННОГО ЦИКЛА СОЗДАНИЯ ПО ИС СОГЛАСНО СТАНДАРТА ISO/IEC 12207

Состав жизненного цикла включает следующие стадии:

1. формирование требований к программному обеспечению;
2. проектирование;
3. реализация;
4. тестирование;
5. ввод в действие;
6. эксплуатация и сопровождение;
7. снятие с эксплуатации.

ОСНОВНЫЕ МОДЕЛИ ЖЦ ИС

Наибольшее распространение получили следующие модели жизненного цикла ИС:

- каскадная (классическая или водопадная);
- итерационная;
- спиральная.

ОСНОВНЫЕ МОДЕЛИ ЖЦ ИС. КАСКАДНАЯ МОДЕЛЬ

Предложена в 1970 году Уинстоном Ройсом.

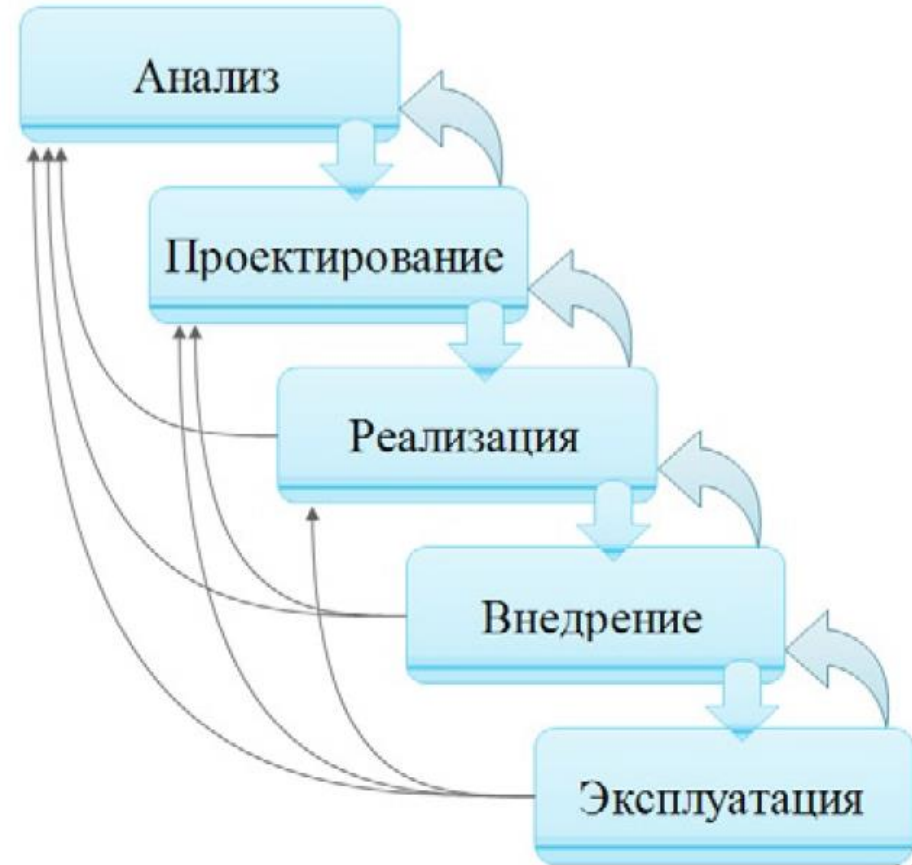
Переход на следующий этап осуществляется после полного окончания работ по предыдущему этапу. Оформляется полный комплект рабочей документации. Все этапы выполняются в строгой последовательности с утвержденными сроками и четкими затратами.



ОСНОВНЫЕ МОДЕЛИ ЖЦ ИС. ИТЕРАЦИОННАЯ МОДЕЛЬ

Каждый этап имеет обратные связи в процессе корректировки и создает условия для корректировки ранее созданных этапов.

Трудоемкость работ и временные затраты существенно сокращаются по сравнению с водопадной моделью жизненного цикла. Создание информационной системы – это организованный процесс построения и последовательного преобразования согласованных моделей на всех этапах жизненного цикла. При этом все разработанные модели находятся в репозитории проекта и доступны всем разработчикам, что позволяет эффективно вести одновременную работу над проектированием и созданием информационной системы.



ОСНОВНЫЕ МОДЕЛИ ЖЦ ИС. СПИРАЛЬНАЯ МОДЕЛЬ

Предложена Барри Боэм в 1988 году и определяет, в основном стартовые этапы жизненного цикла информационной системы. При этом обосновывается и проверяется возможность реализации спроектированных технических решений.

На каждом витке создается прототип проектируемой информационной системы, который на следующих витках спирали ЖЦ ИС совершенствуется, дополняется и доводится до полного внедрения. При этом не обязательно дожидаться окончания каждого этапа. Модель позволяет переходить на следующие витки спирали и решать проблемы или недоделки на следующем уровне, что делает работу над проектом более эффективной, гибкой и завершить в более сжатые сроки.

Новый виток спирали соответствует поэтапной модели создания фрагмента информационной системы.

При использовании спиральной модели ЖЦ:

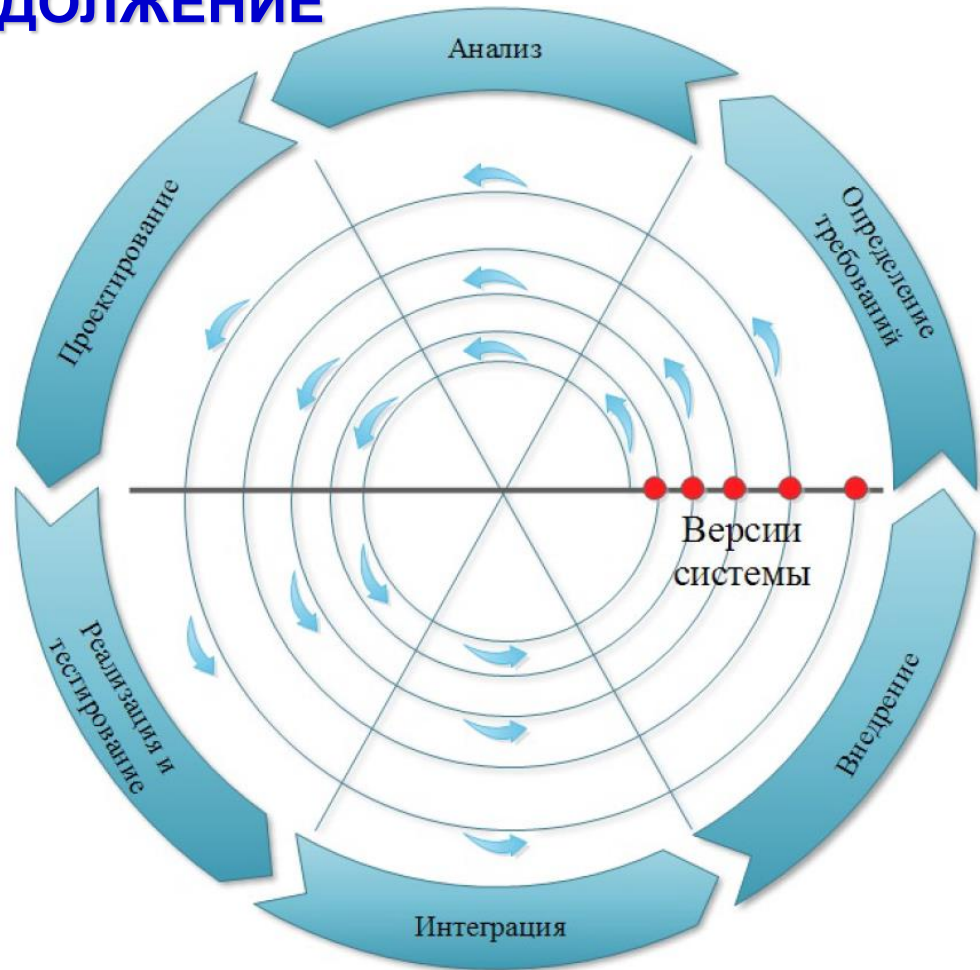
- происходит ориентация на модернизацию информационной системы;
- осуществляется аккумулирование всех решений в процессе проектирования и создания моделей и
- прототипов информационной системы;
- проводится анализ издержек и всех рисков в процессе проектирования ИС.

ОСНОВНЫЕ МОДЕЛИ ЖЦ ИС. СПИРАЛЬНАЯ МОДЕЛЬ. ПРОДОЛЖЕНИЕ

Спиральный процесс состоит из следующей повторяющейся последовательности:

- Определение требований.
- Анализ.
- Проектирование.
- Реализация и тестирование.
- Интеграция.
- Внедрение.

Этот многократный цикл, завершающийся созданием новой версии ИС.



ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ СИСТЕМ

Поиск рациональных путей проектирования идет по следующим направлениям:

- разработка типовых проектных решений, зафиксированных в пакетах прикладных программ;
- разработка автоматизированных систем проектирования.

Автоматизированные системы проектирования - второй путь ведения проектировочных работ. В этой области сформировалось новое направление: CASE-технологии (Computer Aided Software Engineering). Важное направление в развитие этой технологии составили разработки интегрированных инструментальных средств, представляющие собой комплексные технологии, ориентированные на создание сложных автоматизированных управленческих систем.

Основная цель CASE - технологий - отделить проектирование ИС от кодирования и последующих этапов разработки, а также максимально автоматизировать процессы разработки и функционирования систем. При использовании CASE - технологий изменяется технология ведения работ на всех этапах жизненного цикла ИС.

ОСБОЕННОСТИ ИСПОЛЬЗОВАНИЯ CASE-СРЕДСТВ

Под CASE-средством понимается специальное программное обеспечение, поддерживающее процессы создания и сопровождения информационных систем: анализ и формулировка требований, проектирование прикладного программного обеспечения и баз данных, генерация кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.

Полная среда разработки информационной системы представляет из себя совокупность используемых CASE-средств, системного программного обеспечения и технических средств.

Негативные факторы внедрения CASE-средств:

не всегда сразу же дают ожидаемый эффект;

бюджет, требуемый на внедрение, в большинстве случаев существенно превышает их рыночную стоимость.

Факторы влияющие на успешное внедрение CASE-средств:

- технология (ограниченность существующих возможностей);
- культура (готовность к внедрению новых процессов);
- управление (четкое руководство важными этапами и процессами внедрения).

МЕТОДОЛОГИЯ RAD

Методология создания ИС, основанная на использовании средств быстрой разработки приложений, получило широкое распространение в последнее время, и приобрели название методологии быстрой разработки приложений Rapid Application Development (RAD). Она охватывает все этапы жизненного цикла современных ИС. Методология RAD - это комплекс специальных инструментальных средств, позволяющих оперировать с определенным набором объектов функционально отражающих отдельные информационные компоненты приложения. Концепцию RAD также часто связывают с концепцией визуального программирования.

Под методологией быстрой разработки приложений обычно понимается процесс, основанный на трех элементах:

1. небольшая команда программистов (2-10 чел.);
2. хорошо проработанный производственный график, рассчитанный на короткий срок разработки (2-6 мес.);
3. итерационная модель разработки, основанная на тесном взаимодействии заказчика.

ОСНОВНЫЕ ПРИНЦИПЫ И ЖЦ МЕТОДОЛОГИИ RAD

1. используется спиральная модель разработки;
2. полное завершение работ на каждом из этапов жизненного цикла;
3. процесс разработки ИС обеспечивается взаимодействием с заказчиком и будущим пользователем;
4. применяются CASE - средства и средства быстрой разработки приложений;
5. применяются средства управления конфигураций, облегчающих внесение изменений в проект и сопровождение готовой системы;
6. используются прототипы, позволяющие полнее выявить и реализовать потребности конечного пользователя.

Жизненный цикл разработки ПО по методологии RAD состоит из четырех фаз:

- фаза анализа и планирования требований;
- фаза проектирования;
- фаза построения;
- фаза внедрения.

В методологии RAD каждый прототип развивается в часть будущей системы.

ОГРАНИЧЕНИЯ МЕТОДОЛОГИИ RAD

Методология RAD не может претендовать на универсальность, она предпочтительна для относительно небольших проектов, разрабатываемых для конкретного заказчика.

Она не подходит для проектов, когда необходим высокий уровень планирования и жесткая дисциплина проектирования, строгое следование заранее разработанным протоколам и интерфейсам. К таким проектам можно отнести:

разработку типовых систем, представляющих собой комплекс типовых компонент, централизованно сопровождаемых, адаптируемых к программно-техническим платформам, СУБД, средств телекоммуникации и др., когда на первый план выступают такие показатели проекта, как управляемость и качество, которые могут войти в противоречие с простотой и скоростью разработки.

Методология RAD неприменима для построения сложных расчетных программ, операционных систем или программ управления космическими кораблями, т.е. программ, требующих написания большого объема (сотни тысяч строк) уникального кода.

Не подходят для разработки по методологии RAD приложения, в которых отсутствует ярко выраженная интерфейсная часть, наглядно определяющая логику работы системы (например, приложения реального времени) и приложения, от которых зависит безопасность людей (например, управление самолетом или атомной электростанцией), так как итеративный подход предполагает, что первые несколько версий наверняка не будут полностью работоспособны, что в данном случае исключается.

МЕТОДОЛОГИЯ RUP

В методологии RUP реализуются следующие подходы:

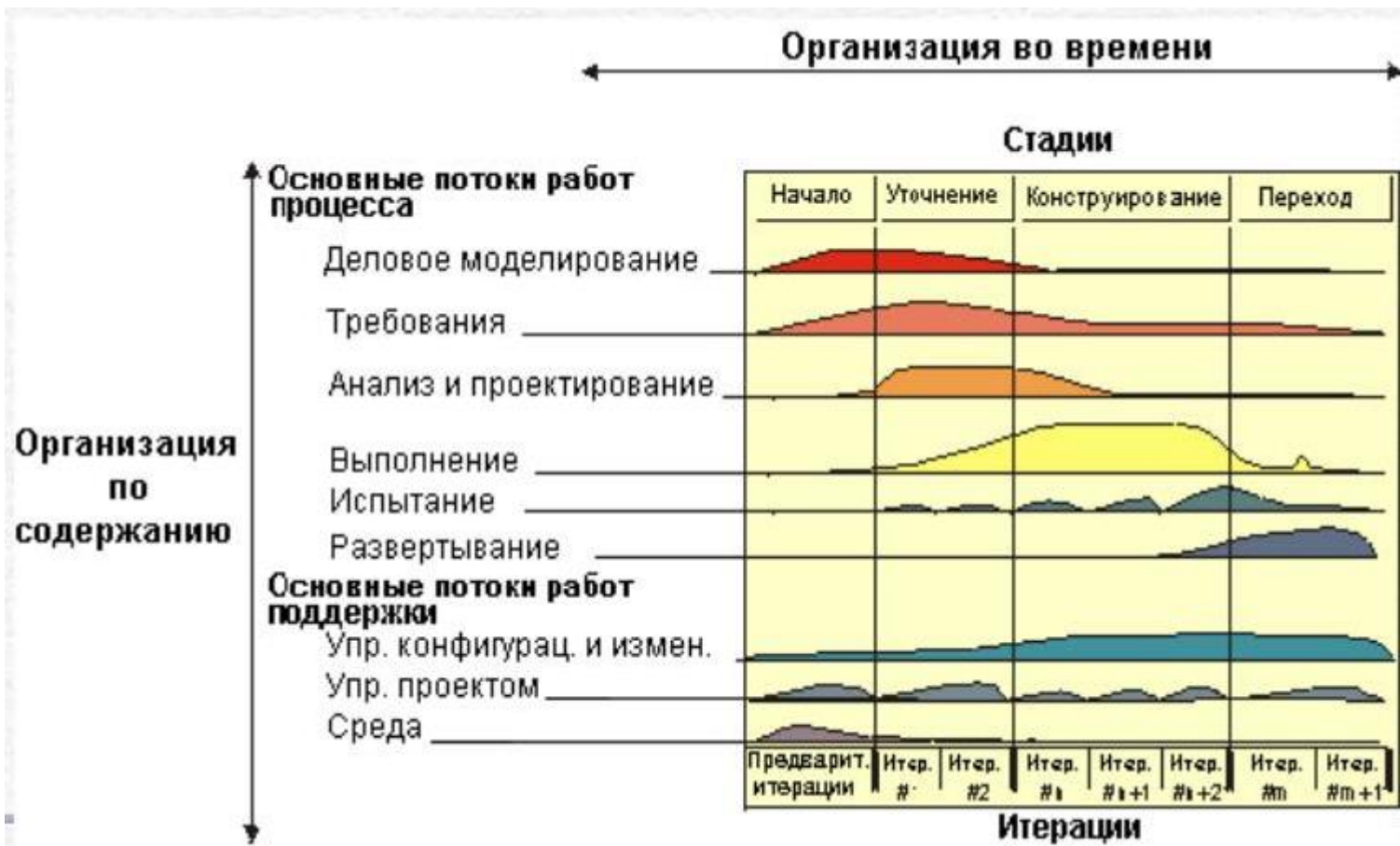
1. Итерационный и инкрементный (наращиваемый).
2. Построение системы на базе архитектуры информационной системы.
3. Планирование и управление проектом на основе функциональных требований к информационной системе.

Разработка информационной системы выполняется итерациями. Это отдельные проекты небольшие по объему и содержанию, которые включают свои собственные этапы анализа требований, проектирования, реализации, тестирования, интеграции. Заканчиваются итерации созданием работающей информационной подсистемы.

Итерационный цикл характеризуется периодической обратной связью и может адаптироваться к ядру разрабатываемой системы.

Создаваемая информационная система постепенно растет и совершенствуется.

ОБЩЕЕ ПРЕДСТАВЛЕНИЕ ЖЦ ПО В СТАНДАРТЕ RUP



ТИПОВОЕ ПРОЕКТИРОВАНИЕ ИС

Типовое проектирование ИС предполагает создание системы из готовых типовых элементов. Основопологающим требованием для применения методов типового проектирования является возможность декомпозиции проектируемой ИС на множество составляющих компонентов (подсистем, комплексов задач, программных модулей и т.д.). Для реализации выделенных компонентов выбираются имеющиеся на рынке типовые проектные решения, которые настраиваются на особенности конкретного предприятия.

Типовое проектное решение (ТПР)- это тиражируемое (пригодное к многократному использованию) проектное решение.

Принятая классификация ТПР основана на уровне декомпозиции системы. Выделяются следующие классы ТПР:

1. элементные ТПР - типовые решения по задаче или по отдельному виду обеспечения задачи (информационному, программному, техническому, математическому, организационному);
2. подсистемные ТПР - в качестве элементов типизации выступают отдельные подсистемы, разработанные с учетом функциональной полноты и минимизации внешних информационных связей;
3. объектные ТПР - типовые отраслевые проекты, которые включают полный набор функциональных и обеспечивающих подсистем ИС.

Каждое типовое решение предполагает наличие, кроме собственно функциональных элементов (программных или аппаратных), документации с детальным описанием ТПР и процедур настройки в соответствии с требованиями разрабатываемой системы.

ДОСТОИНСТВА И НЕДОСТАТКИ ТПР ИС

Класс ТПР / Реализация ТПР	Достоинства	Недостатки
<p>Элементные ТПР Библиотеки методо-ориентированных программ</p>	<ul style="list-style-type: none"> •обеспечивается применение модульного подхода к проектированию и документированию ИС 	<ul style="list-style-type: none"> •большие затраты времени на сопряжение разнородных элементов вследствие информационной, программной и технической несовместимости •большие затраты времени на доработку ТПР отдельных элементов
<p>Подсистемные ТПР Пакеты прикладных программ</p>	<ul style="list-style-type: none"> •достигается высокая степень интеграции элементов ИС •позволяют осуществлять: модульное проектирование; параметрическую настройку программных компонентов на различные объекты управления •обеспечивают: сокращение затрат на проектирование и программирование взаимосвязанных компонентов; хорошее документирование отображаемых процессов обработки информации 	<ul style="list-style-type: none"> •адаптивность ТПР недостаточна с позиции непрерывного инжиниринга деловых процессов •возникают проблемы в комплексировании разных функциональных подсистем, особенно в случае использования решений нескольких производителей программного обеспечения
<p>Объектные ТПР Отраслевые проекты ИС</p>	<ul style="list-style-type: none"> •комплексирование всех компонентов ИС за счет методологического единства и информационной, программной и технической совместимости •открытость архитектуры — позволяет устанавливать ТПР на разных программно-технических платформах •масштабируемость — допускает конфигурацию ИС для переменного числа рабочих мест •конфигурируемость — позволяет выбирать необходимое подмножество компонентов 	<ul style="list-style-type: none"> •проблемы привязки типового проекта к конкретному объекту управления, что вызывает в некоторых случаях даже необходимость изменения организационно-экономической структуры объекта автоматизации

ПОДХОДЫ ТИПОВОГО ПРОЕКТИРОВАНИЯ

При **реализации типового проектирования** применяются такие подходы, как: *параметрически-ориентированное и модельно-ориентированное проектирование*.

Этапами *параметрически-ориентированного проектирования* являются:

- постановка задач и определение пригодности пакетов прикладных программ (ППП) для их решения через систему критериев оценки;
- анализ доступных ППП исходя из критериев;
- выбор и приобретение подходящего ППП;
- настройка параметров приобретенного ППП.

При *модельно-ориентированном проектировании* выполняется адаптация существующих характеристик типовой ИС, исходя из модели объекта автоматизации, построение которой предполагает использование специального программного инструментария. При этом технология проектирования должна иметь средства как для работы с моделью конкретного предприятия, так и с моделью типовой ИС. В репозитории типовой ИС содержится модель объекта автоматизации, которая является основой для конфигурирования программного обеспечения, а также базовая (ссылочная) модель ИС и типовая (референтная) модели ее определенных классов.

Модель конкретного предприятия может быть построена либо в результате выбора фрагментов типовой модели с учетом особенностей объекта автоматизации (например, с использованием BAAN Enterprise Modeler), либо с использованием автоматизированной адаптации этих модулей с учетом мнений экспертов (например, SAP Business Engineering Workbench).

ОСНОВНЫЕ ЭТАПЫ ТИПОВОГО ПРОЕКТИРОВАНИЯ

Внедрение типовой ИС начинается с **анализа результатов предпроектного обследования предприятия**, сформированных в виде требований к конкретной ИС, для оценки которых может быть использована методика оценки ППП.

На следующем этапе необходимо построить **предварительную модель ИС**, которая должна полно отражать особенности реализации ИС для конкретного объекта автоматизации. Предварительная модель – основа для **выбора типовой модели системы**, а также для формирования перечня компонентов, для реализации которых потребуются другие программные средства или инструментальные средства, имеющиеся в составе типовой ИС.

При реализации типового проекта имеет место выполнение следующих операций:

- установку глобальных параметров системы;
- задание структуры объекта автоматизации;
- определение структуры основных данных;
- задание перечня реализуемых функций и процессов;
- описание интерфейсов;
- описание отчетов;
- настройку авторизации доступа;
- настройку системы архивирования.

СТРУКТУРНЫЙ ПОДХОД К ПРОЕКТИРОВАНИЮ ИС

Структурным принято называть такой метод исследования системы который начинается с общего обзора объекта исследования а затем предполагается его последовательную детализацию.

Структурный метод имеет 3 основных особенности:

1. Расчленение сложной системы на части представляемые как черные ящики каждый из которого выполняет определенную функцию систем управления.
2. Иерархическое упорядочение выделенных элементов системы с определенной взаимосвязи между ними.
3. Использование графического представления взаимосвязи элементов системы для повышения наглядности модели и облегчения ее восприятия.

К наиболее распространенным методологиям структурного анализа относят:

1. Методология SADT (IDEFD);
2. DFD - диаграмма потоков данных;
3. ERD - диаграмма сущность-связь (для разработки моделей данных);
4. STD - диаграммы переходов состояний.

Сущность структурного подхода (разработка "сверху-вниз") к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции. При этом автоматизируемая система сохраняет целостное представление, котором все составляющие компоненты взаимосвязаны.

СТРУКТУРНАЯ МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ

В основе проектирования ИС лежит моделирование предметной области. Для того чтобы получить адекватный предметной области проект ИС в виде системы правильно работающих программ, необходимо иметь **целостное, системное представление модели**, которое отражает **все аспекты функционирования** будущей информационной системы.

Под **моделью предметной области** понимается некоторая система, имитирующая структуру или функционирование исследуемой предметной области и отвечающая основному требованию – быть адекватной этой области.

Для отображения **структурного аспекта** моделей предметных областей в основном используются **графические методы**, которые должны гарантировать представление информации о компонентах системы. Главное требование к графическим методам документирования — **простота**. Графические методы должны обеспечивать возможность структурной декомпозиции спецификаций системы с максимальной степенью детализации и согласований описаний на смежных уровнях декомпозиции.

Главный **критерий адекватности структурной модели** предметной области заключается в **функциональной полноте** разрабатываемой ИС.

В основе различных методологий моделирования предметной области ИС лежат принципы последовательной детализации абстрактных категорий. Обычно модели строятся на трех уровнях:

- на внешнем уровне (определении требований),
- на концептуальном уровне (спецификации требований),
- внутреннем уровне (реализации требований).

ПРОЦЕССНЫЙ ПОДХОД

- Процессный подход обеспечивает связи между отдельными процессами в рамках системы процессов, а также их комбинацию и взаимодействие. Преимущество процессного подхода перед функциональным подходом, заключается в лучшей реализации системных свойств.
- *Процессный подход ориентирован, в первую очередь, не на организационную структуру организации, а на бизнес-процессы, конечными целями которых является создание продуктов или услуг, представляющих ценность для внешних или внутренних потребителей.* Идея процессного подхода в том, что успешное управление деятельностью предприятия должно быть реализовано посредством управления через процессы, а не через классические функциональные отделы. Такой подход называют сегментированием, задача которого - функции, выполняемые в подразделениях, четко распределить по процессам.
- Ключевым в понимании процессного подхода является переход от вертикального построения организационной структуры к горизонтальной. *Суть процессного подхода сводится к тому, чтобы доставить клиенту желаемое в кратчайшие сроки, высочайшего качества и по максимально низкой цене.*

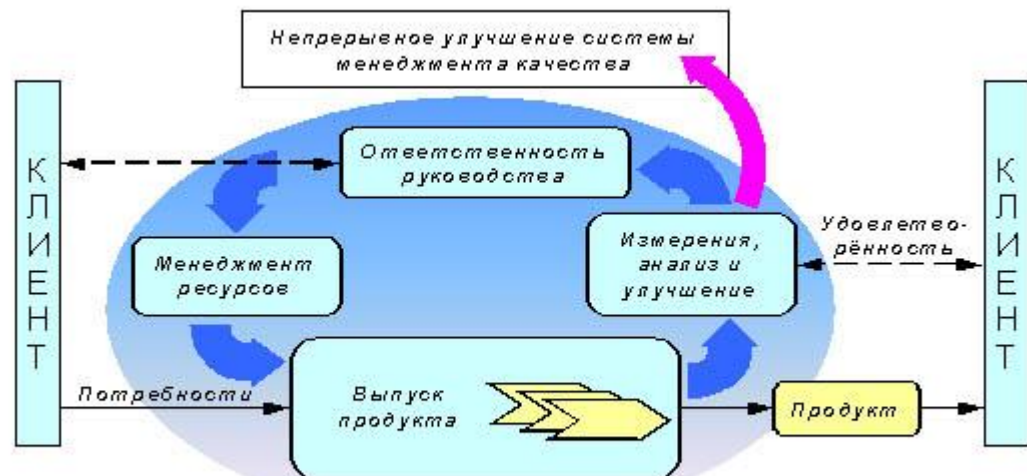
ПУТИ РЕАЛИЗАЦИИ ПРОЦЕССНОГО ПОДХОДА

I. Комплексное, системное рассмотрение деятельности организации как совокупности процессов, разработке системы управления процессами с использованием принципов ISO серии 9001:2000. Этот подход называют «полным» или системным подходом к выделению процессов предприятия.

Схема процессного подхода



Модель системы менеджмента качества, основанной на процессах



II. Выделение в организации "сквозных" процессов, их описании и последующей реорганизации.

МЕТОДОЛОГИИ ОПИСАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ

В соответствии с различными представлениями об организации методике проектирования принято делить на **объектные** и **функциональные** (структурные).

Объектные методики рассматривают моделируемую организацию как набор взаимодействующих объектов – производственных единиц. **Объект** определяется как осязаемая реальность – предмет или явление, имеющие четко определяемое поведение. Целью применения данной методики является выделение объектов, составляющих организацию, и распределение между ними ответственностей за выполняемые действия.

Функциональные методики, наиболее известной из которых является методика IDEF, рассматривают организацию как набор функций, преобразующий поступающий поток информации в выходной поток. Процесс преобразования информации потребляет определенные ресурсы. Основное отличие от объектной методики заключается в четком отделении функций (методов обработки данных) от самих данных.

Объектный подход позволяет построить более устойчивую к изменениям систему, лучше соответствует существующим структурам организации.

Функциональное моделирование хорошо показывает себя в тех случаях, когда организационная структура находится в процессе изменения или вообще слабо оформлена. Подход от выполняемых функций интуитивно лучше понимается исполнителями при получении от них информации об их текущей работе.

МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ IDEF0

Методологию IDEF0 можно считать следующим этапом развития графического языка описания функциональных систем SADT (Structured Analysis and Design Technique).

Исторически IDEF0 как стандарт был разработан в 1981 году в рамках обширной программы автоматизации промышленных предприятий, которая носила обозначение ICAM (Integrated Computer Aided Manufacturing). Семейство стандартов IDEF унаследовало свое обозначение от названия этой программы (IDEF=Icam DEFinition), и последняя его редакция была выпущена в декабре 1993 года Национальным Институтом по Стандартам и Технологичам США (NIST).

Целью методологии является построение функциональной схемы исследуемой системы, описывающей все необходимые процессы с точностью, достаточной для однозначного моделирования деятельности системы.

В основе методологии лежат четыре основных понятия: **функциональный блок, интерфейсная дуга, декомпозиция, глоссарий**.

Необходимо отметить, что любой функциональный блок по требованиям стандарта должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую. Это и понятно – каждый процесс должен происходить по каким-то правилам (отображаемым управляющей дугой) и должен выдавать некоторый результат (выходящая дуга), иначе его рассмотрение не имеет никакого смысла.

Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (Data Flow Diagram) и WFD (Work Flow Diagram).

ОСОБЕННОСТИ МОДЕЛИ IDEFO

Модель IDEFO всегда начинается с представления системы как единого целого – одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется контекстной диаграммой.

В пояснительном тексте к контекстной диаграмме должна быть указана **цель (Purpose)** построения диаграммы в виде краткого описания и зафиксирована **точка зрения (Viewpoint)**.

Фактически **цель** определяет соответствующие области в исследуемой системе, на которых необходимо фокусироваться в первую очередь.

Точка зрения определяет основное направление развития модели и уровень необходимой детализации.

В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется дочерней (Child Diagram) по отношению к нему (каждый из функциональных блоков, принадлежащих дочерней диаграмме, соответственно называется дочерним блоком – Child Box).

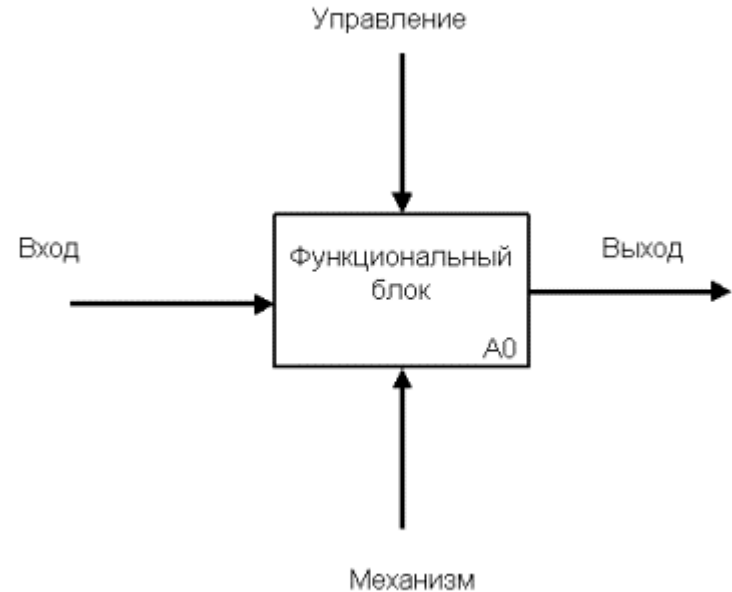
В свою очередь, функциональный блок — предок называется родительским блоком по отношению к дочерней диаграмме (Parent Box), а диаграмма, к которой он принадлежит – родительской диаграммой (Parent Diagram).

ФУНКЦИОНАЛЬНЫЙ БЛОК IDEF0

Функциональный блок графически изображается в виде прямоугольника и олицетворяет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении (например, “производить услуги”). Каждая из четырех сторон функционального блока имеет своё определенное значение (роль), при этом:

- Верхняя сторона имеет значение “Управление” (Control);
- Левая сторона имеет значение “Вход” (Input);
- Правая сторона имеет значение “Выход” (Output);
- Нижняя сторона имеет значение “Механизм” (Mechanism).

Каждый функциональный блок в рамках единой рассматриваемой системы должен иметь свой уникальный идентификационный номер.

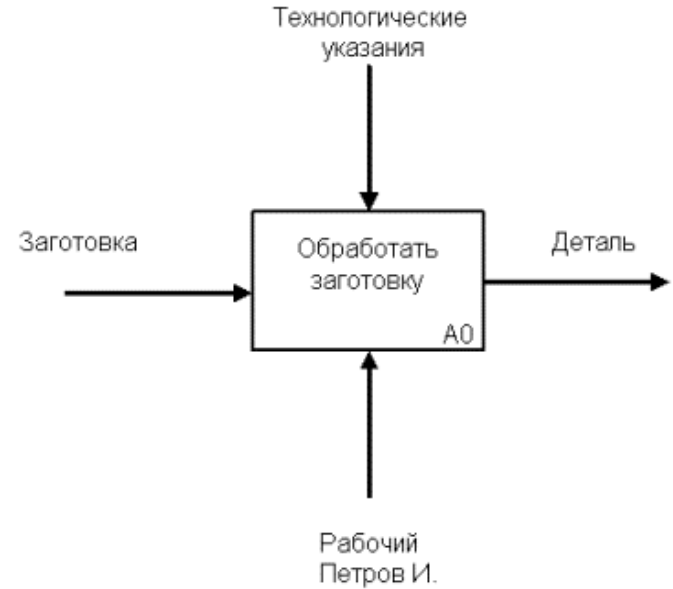


ПОНЯТИЕ ИНТЕРФЕЙСНОЙ ДУГИ IDEFO

Интерфейсные дуги часто называют потоками или стрелками. Интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, отображенную данным функциональным блоком. Графическим отображением интерфейсной дуги является однонаправленная стрелка. Каждая интерфейсная дуга должна иметь свое уникальное наименование (Arrow Label). По требованию стандарта, наименование должно быть оборотом существительного.

С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Такими объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т.д.) или потоки данных и информации (документы, данные, инструкции и т.д.).

В зависимости от того, к какой из сторон подходит данная интерфейсная дуга, она носит название “входящей”, “исходящей” или “управляющей”.

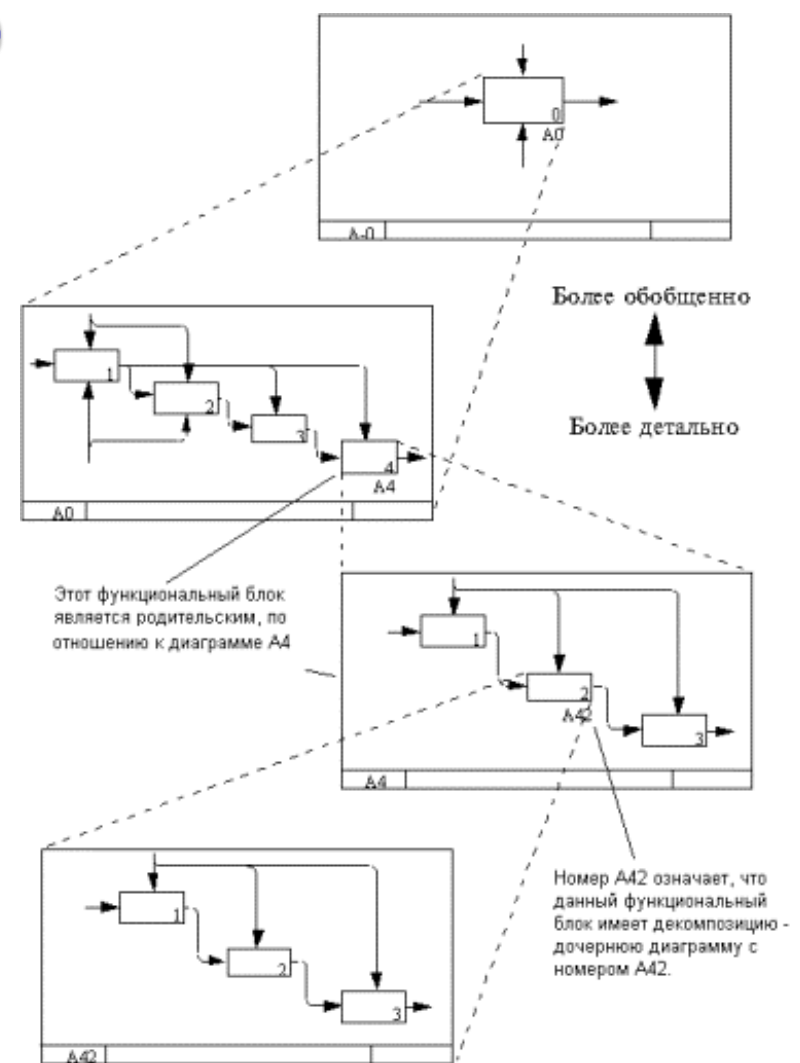


Любой функциональный блок должен иметь по крайней мере одну управляющую интерфейсную дугу и одну исходящую. При построении IDEF0 – диаграмм важно правильно отделять входящие интерфейсные дуги от управляющих.

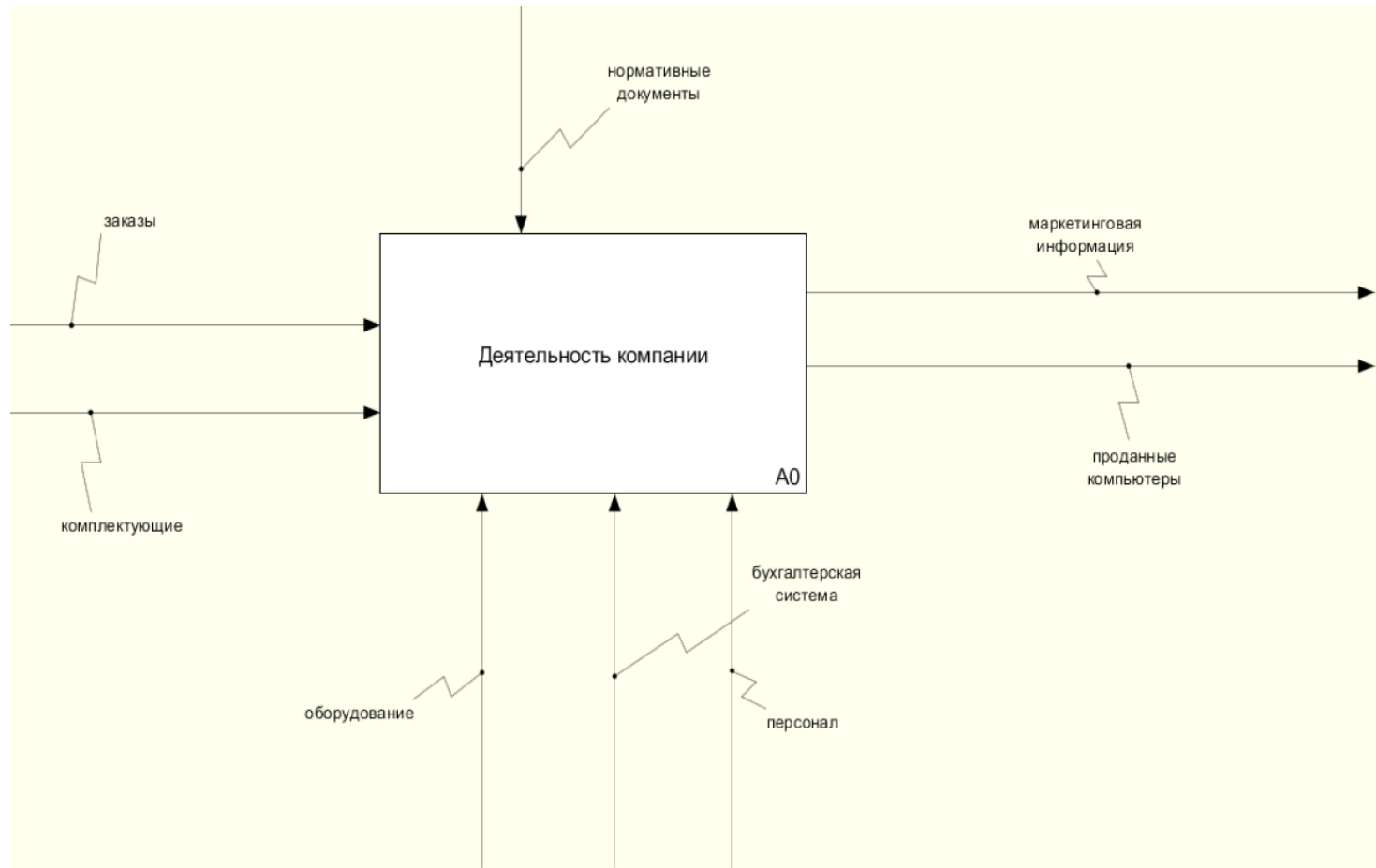
ПРИМЕРЫ ДИАГРАММ IDEFO

Для достижения структурной целостности модели, используются следующие правила:

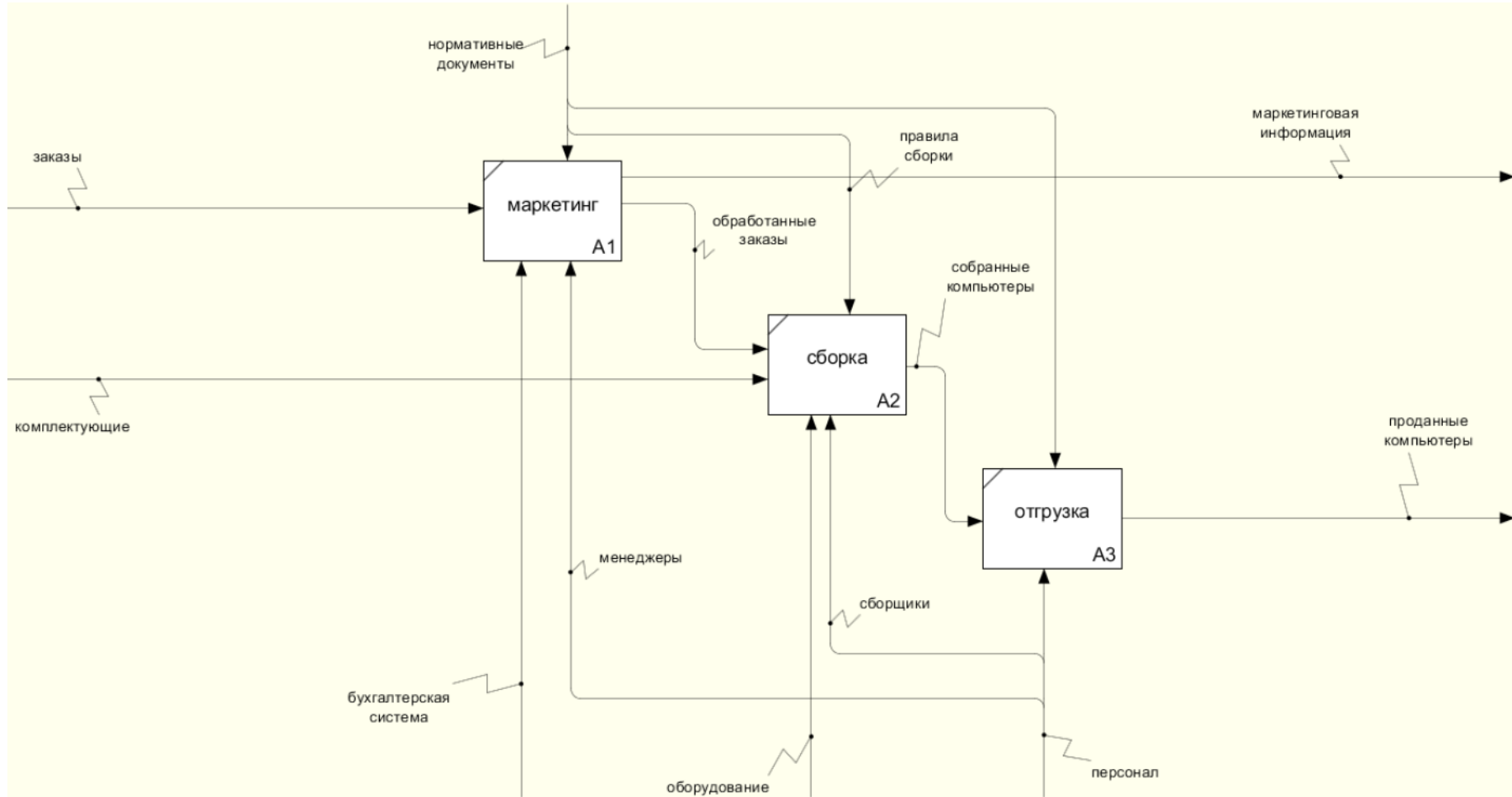
- Все интерфейсные дуги, входящие в данный блок, или исходящие из него фиксируются на дочерней диаграмме.
- При нумерации блоков, цифра в правом нижнем углу прямоугольника указывает на уникальный порядковый номер самого блока на диаграмме, а обозначение под правым углом указывает на номер дочерней для этого блока диаграммы.



ПРИМЕР ДИАГРАММ IDEFO0 ВЕРХЕГО УРОВНЯ



ПРИМЕР ДЕКОМПОЗИЦИИ ДИАГРАММ IDEFO



ОСОБЕННОСТИ ПРАКТИКИ ПРИМЕНЕНИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ СРЕДСТВАМИ IDEF0

Разработка моделей в стандарте IDEF0 позволяет наглядно и эффективно отобразить весь механизм деятельности предприятия в нужном разрезе. При этом стандарт предоставляет **возможность коллективной работы**.

При проведении моделирования, сотрудники обследуемой компании должны были ответить на следующие вопросы:

- Что поступает в подразделение “на входе”?
- Какие функции, и в какой последовательности выполняются в рамках подразделения?
- Кто является ответственным за выполнение каждой из функций?
- Чем руководствуется исполнитель при выполнении каждой из функций?
- Что является результатом работы подразделения (на выходе)?

В результате, за достаточно короткое время и при привлечении минимума человеческих ресурсов получается IDEF0-модель предприятия по принципу “Как есть”. Она представляет предприятие с позиции сотрудников, которые в нем работают и досконально знают все нюансы, в том числе неформальные. В дальнейшем, эта модель будет передана на анализ и обработку к бизнес-аналитикам, которые будут заниматься поиском “узких мест” в управлении компанией и оптимизацией основных процессов, трансформируя модель “Как есть” в соответствующее представление “Как должно быть”.

МЕТОДОЛОГИЯ МОДЕЛИРОВАНИЯ ДАННЫХ IDEF1X

Целью методологии IDEF1X является выработка непротиворечивого интегрированного определения семантических характеристик данных, которые могут использоваться для организации потоков данных и управления при создании распределенных баз данных и интеграции информационных систем.

Компонентами IDEF1X-модели являются:

1. Сущности:

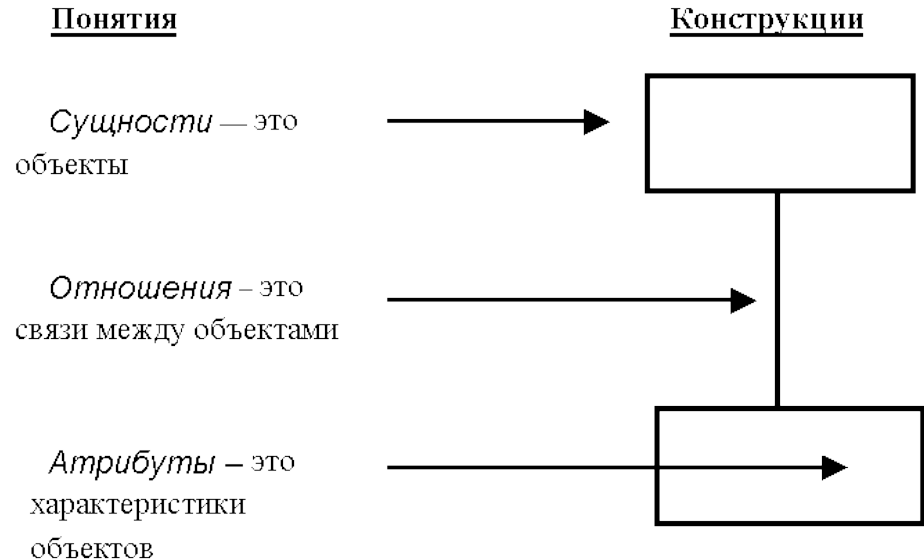
- Независимые от идентификаторов сущности;
- Зависимые от идентификаторов сущности.

2. Отношения:

- Отношения, идентифицирующие связи;
- Отношения, не идентифицирующие связи;
- Отношения категоризации;
- Неспецифические отношения.

3. Атрибуты/ключи:

- Атрибуты,
- Первичные ключи,
- Альтернативные ключи,
- Внешние ключи.



ОСОБЕННОСТИ МЕТОДОЛОГИИ МОДЕЛИРОВАНИЯ ДАННЫХ IDEF1X

Методология разработана с учетом следующих **требований**:

1. Поддерживает разработку концептуальных схем;
2. Обеспечивает ясный язык;
- 3 Проста для изучения;
4. Надежно проверена на практике;
5. Возможность автоматизации.

IDEF1X использует подход сущностей-отношений к семантическому моделированию данных.

Исходная разработка IDEF1 заключалась в расширении понятий сущности-отношения по методу П.Ченна, объединенных с понятиями **реляционной теории Т. Кодда**.

Основным преимуществом IDEF1X является жесткая и строгая стандартизация моделирования.

Недостатком можно считать, то что средства моделирования IDEF1X специально разработаны для построения **реляционных ИС**, и их **не следует применять для других типов ИС**.

Обоснование. IDEF1X требует от проектировщика определить ключевые атрибуты, для того чтобы отличить одну сущность от другой, в то время как объектно-ориентированные системы не требуют задания ключевых ключей, в целях идентифицирования объектов. Проектировщик должен определить один из этих атрибутов первичным ключом, а все остальные вторичными.

ОСНОВНЫЕ СТАДИИ МЕТОДОЛОГИИ IDEF1X

Стадия 0 - начало работы над проектом.

IDEF1X-модель данных должна быть описана и определена в терминах как ее ограничений, так и целей. Разработчик модели является одним из главных лиц при определении назначения модели. Вместе с руководителем проекта разработчик модели составляет план для достижения цели на Стадии 0. Реализация этого плана наряду с другой описательной и пояснительной информацией является результатом работы на Стадии 0.

Стадия 1 - определение сущностей.

Целью Стадии 1 является выявление и определение сущностей, находящихся в пределах моделируемой проблемной области. Первым шагом в этом процессе является идентификация сущностей.

Стадия 2 - определение отношений.

Целью Стадии 2 является выявление и определение основных отношений между сущностями. На этой стадии моделирования некоторые отношения могут быть неспецифическими и потребуют дополнительной детализации на последующих стадиях.

ОСНОВНЫЕ СТАДИИ МЕТОДОЛОГИИ IDEF1X. ПРОДОЛЖЕНИЕ

Стадия 3 - определения ключей.

На Стадии 3 методологии IDEF1X идентифицируются и определяются элементы данных об экземплярах сущностей, называемых возможными ключами, первичными ключами, альтернативными ключами и внешними ключами.

Цель этого этапа - установить значения атрибутов, однозначно определяющих каждый экземпляр сущности.

Стадия 4 - определение атрибутов.

Результаты Стадии 4 изображаются на одной или нескольких диаграммах (диаграммах уровня атрибутов).

В конце Стадии 4 модель данных полностью детализируется (в соответствии с пятой нормальной формой в реляционной теории).

МЕТОДОЛОГИЯ ОПИСАНИЯ ПРОЦЕССОВ IDEF3

IDEF3 - методология моделирования и стандарт документирования процессов, происходящих в системе. Метод документирования технологических процессов представляет собой механизм документирования и сбора информации о процессах. IDEF3 показывает причинно-следственные связи между ситуациями и событиями в понятной эксперту форме, используя структурный метод выражения знаний о том, как функционирует система, процесс или предприятие.

Методология описания набора данных IDEF3 является частью структурного анализа. IDEF3 не ограничивает аналитика чрезмерно жесткими рамками синтаксиса, что может привести к созданию неполных или противоречивых моделей.

IDEF3 может быть также использован как метод создания процессов. IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые в дальнейшем могут быть использованы для имитационного анализа.

Точка зрения на модель должна быть документирована. Также необходимо документировать цель модели — те вопросы, на которые призвана ответить модель.

Диаграмма является основной единицей описания в IDEF3.

Единицы работы — Unit of Work (UOW) — также называемые работами (activity), являются центральными компонентами модели.

Связи показывают взаимоотношения работ. Все связи в IDEF3 однонаправленные.

ВОЗМОЖНОСТИ МОДЕЛИРОВАНИЯ IDEF3

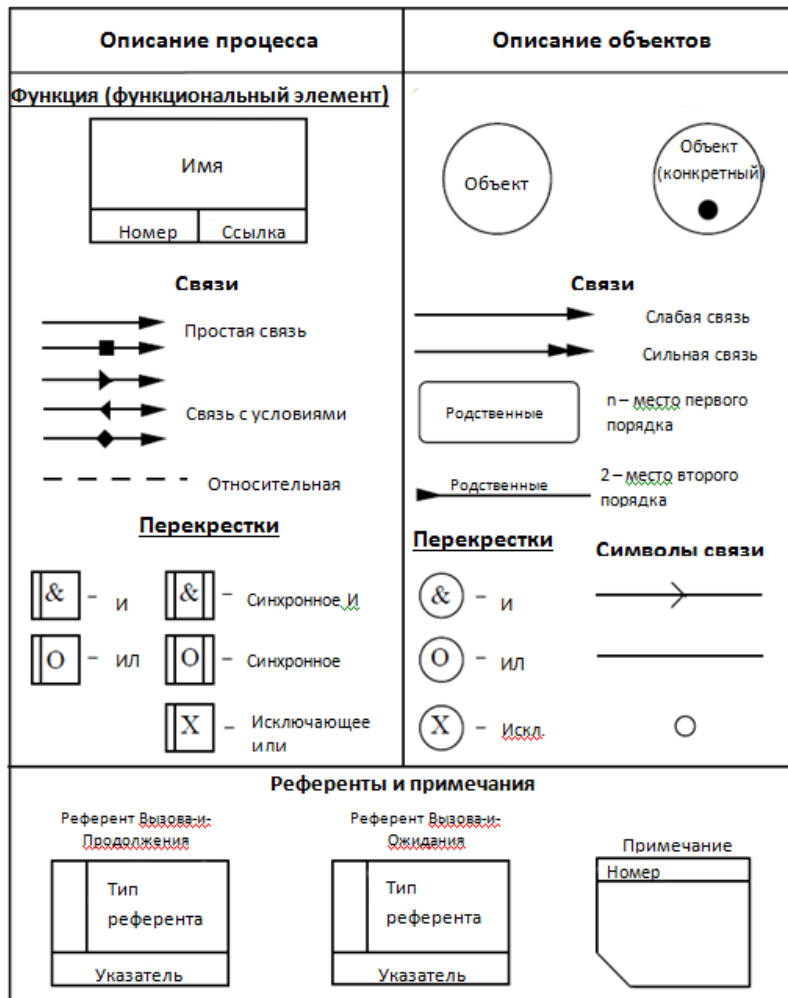
IDEF3 как инструмент моделирования фиксирует следующую информацию о процессе:

- объекты, которые участвуют при выполнении сценария;
- роли, которые выполняют эти объекты, например, агент, транспорт;
- отношения между работами в ходе выполнения сценария процесса;
- состояния и изменения, которым подвергаются объекты;
- время выполнения и контрольные точки синхронизации работ;
- ресурсы, которые необходимы для выполнения работ.

Средства документирования и моделирования IDEF3 позволяют **выполнять следующие задачи**:

- документировать имеющиеся данные о технологии выполнения процесса;
- анализировать существующие процессы и разрабатывать новые;
- определять и анализировать точки влияния потоков сопутствующего документооборота на сценарий технологических процессов;
- определять ситуации, в которых требуется принятие решения, влияющего на **жизненный цикл процесса**;
- содействовать принятию оптимальных решений при реорганизации процессов;
- разрабатывать имитационные модели технологических процессов.

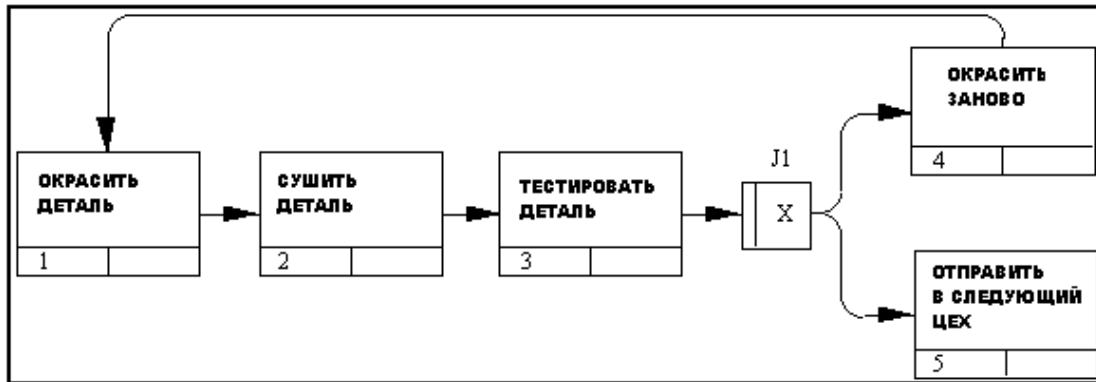
ОСНОВНЫЕ ЭЛЕМЕНТЫ IDEF3-ДИАГРАММ



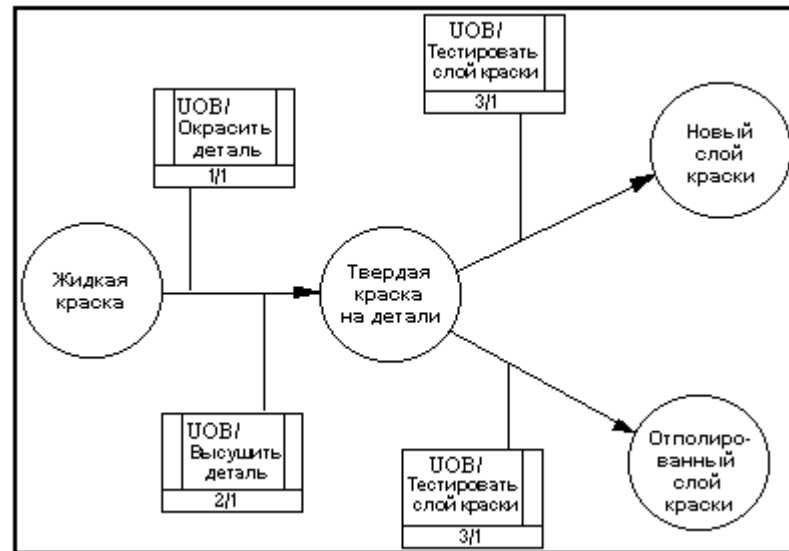
ПРИМЕРЫ IDEF3- ДИАГРАММ

Существуют два типа диаграмм в стандарте IDEF3, представляющие описание одного и того же сценария технологического процесса в разных ракурсах. Диаграммы относящиеся к первому типу называются диаграммами Описания Последовательности Этапов Процесса (Process Flow Description Diagrams, PFDD), а ко второму - диаграммами Состояния Объекта в и его Трансформаций Процессе (Object State Transition Network, OSTN).

PFDD диаграмма

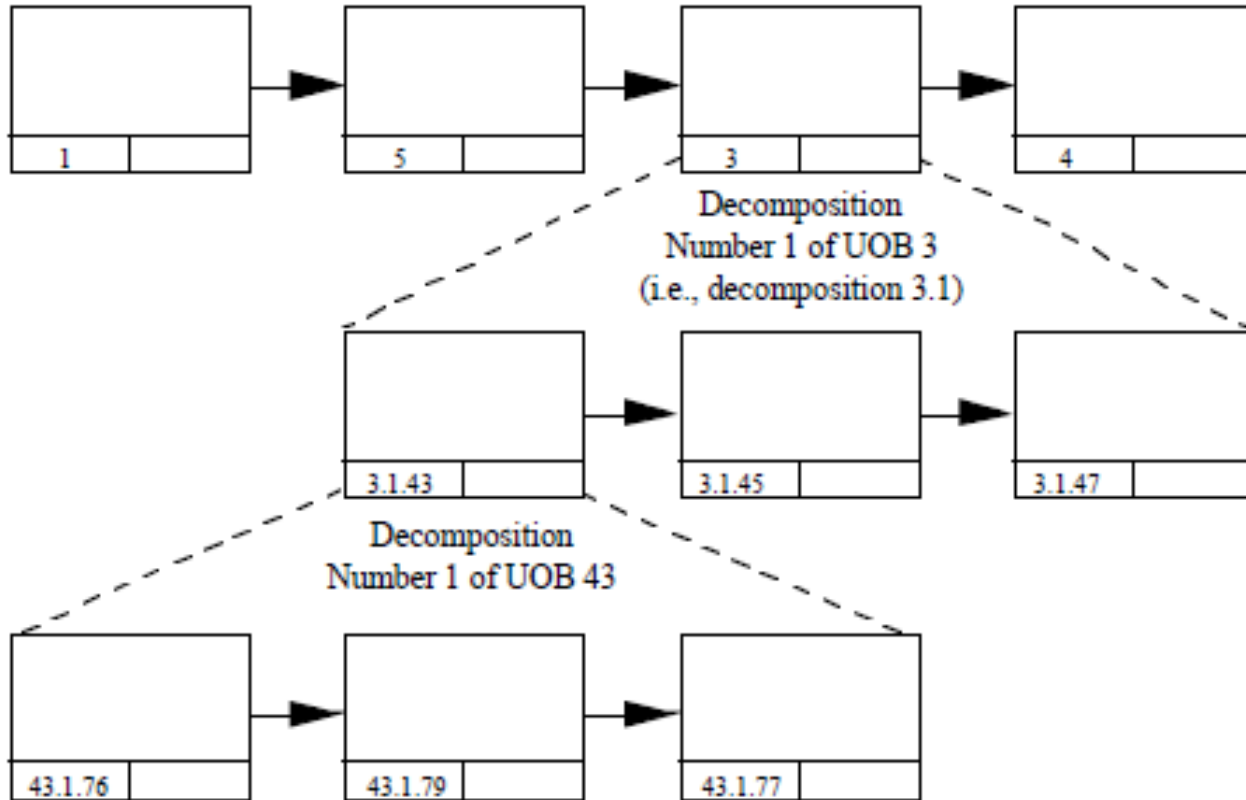


OSTN диаграмма



ДЕКОМПОЗИЦИЯ ОПИСАНИЯ ПРОЦЕССА В МЕТОДОЛОГИИ IDEF3

Методология IDEF3 дает возможность представлять процесс в виде иерархически организованной совокупности диаграмм. Диаграммы состоят из нескольких элементов описания процесса IDEF3, причем каждый функциональный элемент (UOB) потенциально может быть детализирован на другой диаграмме.



МЕТОДОЛОГИЯ ОНТОЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ IDEF5

Методология IDEF5 обеспечивает наглядное представление данных, полученных в результате обработки онтологических запросов в простой естественной графической форме.

Процесс построения онтологии, согласно методологии IDEF5 состоит из пяти основных действий:

- 1) Изучение и систематизирование начальных условий.
- 2) Сбор и накапливание данных.
- 3) Анализ данных.
- 4) Начальное развитие онтологии.
- 5) Уточнение и утверждение онтологии - Заключительная стадия процесса.

Для поддержания построения онтологии в IDEF5 существуют специальные онтологические языки: язык доработок и уточнений (Elaboration Language - EL) и схематический язык (Schematic Language - SL).







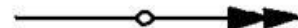




Наиболее важные и заметные зависимости между объектами описываются четырьмя видами основных видов схем, которые используются для накопления информации об онтологии в графической форме.

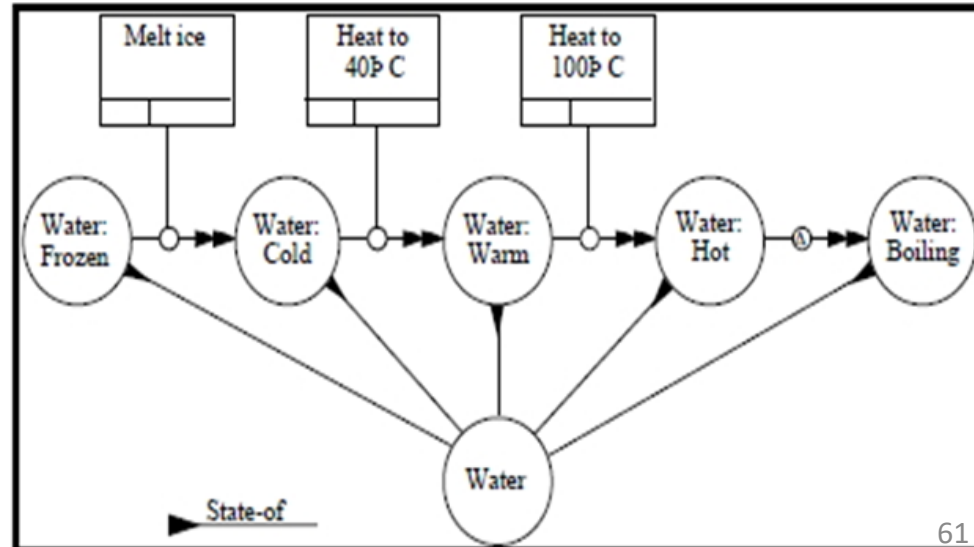
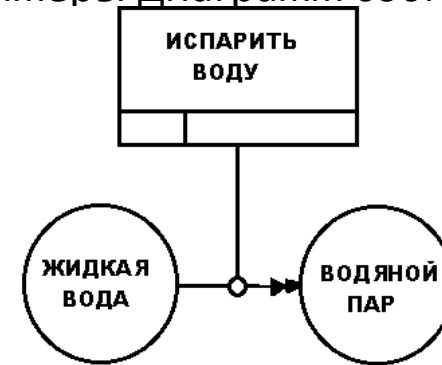
МЕТОДОЛОГИЯ ОНТОЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ IDEF5. ОСНОВНЫЕ ВИДЫ СХЕМ

1. Диаграмма классификаций, обеспечивающая механизм для логической систематизации знаний, накопленных при изучении системы. Существует два типа таких диаграмм: Диаграмма строгой классификации (Description Subsumption - DS) и Диаграмма естественной или видовой классификации (Natural Kind Classification - NKC).
2. Композиционные схемы, являющиеся механизмом графического представления состава классов онтологии и фактически представляют собой инструменты онтологического исследования по принципу ``Что из чего состоит``.
3. Схемы взаимосвязей позволяют разработчикам визуализировать и изучать взаимосвязи между различными классами объектов в системе.
4. Диаграммы состояния объекта, позволяющие документировать тот или иной процесс с точки зрения изменения состояния или вида объекта.

СХЕМАТИЧЕСКИЕ ГРАФИЧЕСКИЕ ИЗОБРАЖЕНИЯ IDEF5

Примеры диаграмм состояний

Обозначения классов, отдельных элементов	Обозначение взаимосвязей и изменения состояния	Обозначение процессов, соединений и перекрестков
<p>Обозначение класса:</p>  <p>Обозначение отдельного элемента:</p> 	<p>Обозначение первичных взаимосвязей:</p> <p>1) Взаимосвязь многие со многими</p>  <p>2) Взаимосвязь двух классов</p>  <p>Обозначение вторичных взаимосвязей между двумя классами:</p>  <p>Обозначения изменения состояния:</p> <p>1) Медленное изменение</p>  <p>2) Быстрое изменение</p>  <p>3) Мгновенное изменение</p> 	<p>Обозначение процесса</p>  <p>Обозначение соединений:</p>  <p>Обозначение перекрестков:</p> 



СТЕК СТАНДАРТОВ IDEF

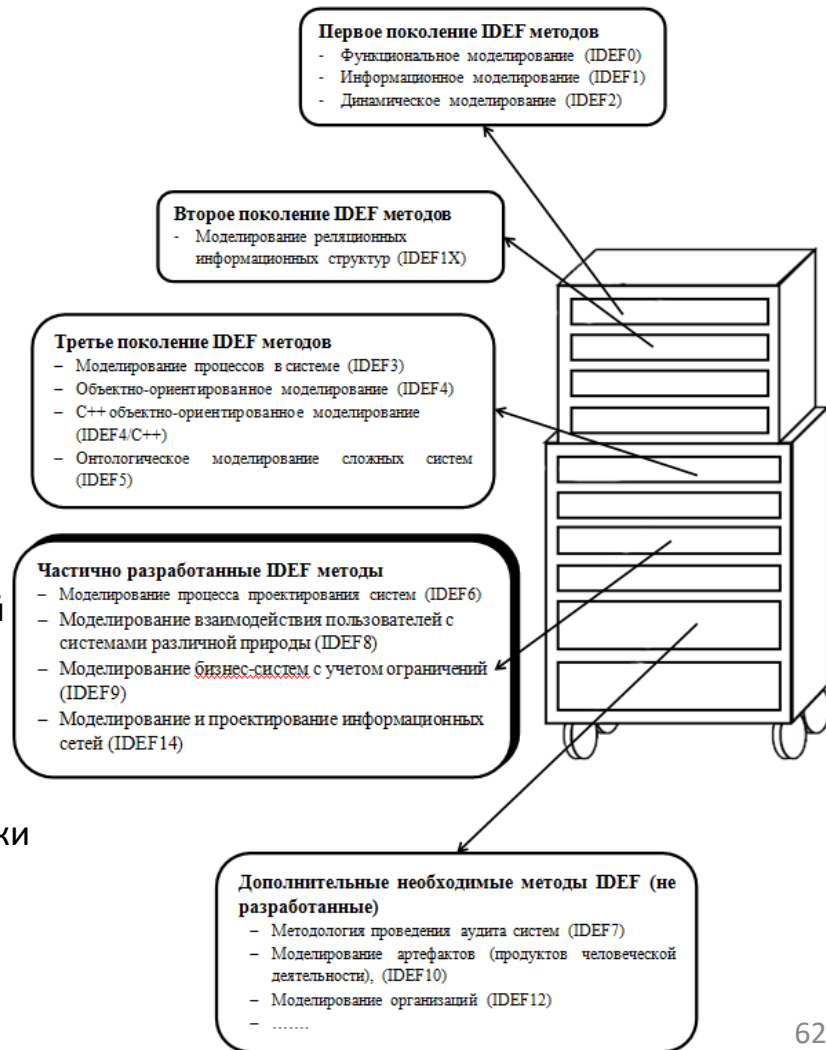
IDEF0 (Function Modeling) – данный метод используется для создания функциональной модели, которая является структурированным отображением функций производственной системы или среды, а также информации и объектов, связывающих эти функции.

IDEF1 (Information Modeling) – данный метод применяется для построения информационной модели, которая представляет собой структурированную информацию, необходимую для поддержки функций производственной системы или среды.

IDEF2 (Simulation Model Design) – данный метод позволяет построить динамическую модель меняющегося во времени поведения функций, информации и ресурсов производственной системы или среды.

IDEF3 (Process Description Capture) – данный метод используется для сбора информации о состоянии моделируемой системы.

IDEF4 (Object-Oriented Design) – данный метод объектно-ориентированного планирования был разработан для поддержки объектно-ориентированной идеологии. Реализован в виде технологий UML.



СТЕК СТАНДАРТОВ IDEF. ПРОДОЛЖЕНИЕ

IDEF5 (Ontology Description Capture) – данный метод позволяет разрабатывать, изучать и поддерживать онтологию моделируемой системы.

IDEF6 (Design Rational Capture Method) - данный метод позволяет использовать рациональный опыт проектирования.

IDEF7 (Information System Auditing) - данный метод описывает проведение методологии аудита информационной системы.

IDEF8 (User Interface Modeling) – данный метод позволяет разрабатывать необходимые модели Графического Интерфейса Пользователя (Human-System Interaction Design). Метод предназначена для проектирования взаимодействия человека и технической системы.

IDEF9 (Business Constraint Discovery) - данная модель предназначена для анализа имеющихся условий и ограничений (в том числе физических, юридических или любых других) и их влияния на принимаемые решения в процессе реинжиниринга.

IDEF10 – (Implementation Architecture Modeling) - Моделирование архитектуры выполнения. Этот метод определён как востребованный, однако так и не был полностью разработан.

IDEF11 – (Information Artifact Modeling) – Моделирование артефактов проектирования. Этот метод определён как востребованный, однако так и не был полностью разработан.

IDEF12 – (Organization Modeling) – Организационное моделирование. Этот метод определён как востребованный, однако так и не был полностью разработан.

IDEF13 – (Three Schema Mapping Design) - Трёхсхемное проектирование преобразования данных. Этот метод определён как востребованный, однако так и не был полностью разработан.

IDEF14 (Network Design Method) – метод проектирования вычислительных сетей, позволяющий устанавливать требования, определять сетевые компоненты, анализировать существующие сетевые конфигурации и формулировать желаемые характеристики сети.

ПРОФИЛИ ОТКРЫТЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

При создании и развитии сложных, распределенных, тиражируемых информационных систем требуется гибкое формирование и применение гармонизированных **совокупностей базовых стандартов** и нормативных документов разного уровня, выделение в них требований и рекомендаций, необходимых для реализации заданных функций системы. Для унификации и регламентирования такие совокупности базовых стандартов должны адаптироваться и конкретизироваться применительно к определенным классам проектов, функций, процессов и компонентов системы. В связи с этим выделилось и сформировалось **понятие профиля информационной системы** как **основного инструмента функциональной стандартизации**.

Профиль – это совокупность нескольких (или подмножество одного) базовых стандартов с четко определенными и гармонизированными подмножествами обязательных и факультативных возможностей, предназначенная для реализации заданной функции или группы функций.

Профиль формируется исходя из функциональных характеристик объекта стандартизации.

Профиль не должен противоречить входящим в него базовым стандартам и нормативным документам.

На базе одной совокупности базовых стандартов могут формироваться и утверждаться различные профили для разных проектов ИС.

Базовые стандарты и профили в зависимости от проблемно-ориентированной области применения ИС могут использоваться как непосредственные директивные, руководящие или рекомендательные документы, а также как нормативная база, необходимая при выборе или разработке средств автоматизации технологических этапов или процессов создания, сопровождения и развития ИС.

СОВЕРШЕНСТВОВАНИЕ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Одной из сложнейших проблем, препятствующих широкому внедрению ИС, является недостаточное знание системными аналитиками и программистами предметных областей, в рамках которых готовятся проекты. Создание и внедрение интеллектуальных систем выдвинуло проблему совершенствования методологии проектирования информационных систем на передний план.

Анализ современного состояния программных средств, приобретения знаний и поддержки деятельности инженера по знаниям позволяет выявить две группы проблем:

1. Методологические проблемы;
2. Технологические проблемы.

Наиболее совершенная и прогрессивная технология это клиент-сервер, с использованием возможности WEB/database. WEB предлагает стандартизацию пользовательского интерфейса, возможность совместной работы разных приложений от разных платформ, простоту разработки приложений, легкость поддержки, хорошо стандартизированные отношения клиент-сервер. Одновременно Базы данных предлагают мощный метод упорядочения и сопровождения информации, представляемой на WEB страницах.

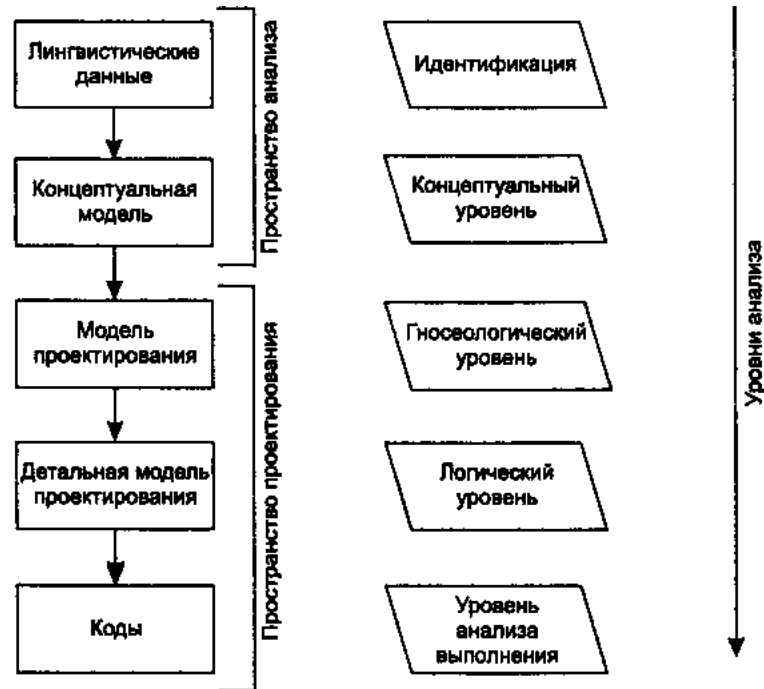
Методология KADS (Knowledge Acquisition and Documentation Structuring), в основе которой лежит понятие интерпретационной модели, позволяющей процессы извлечения, структурирования и формализации знаний рассматривать как "интерпретацию" лингвистических знаний в другие представления и структуры.

KADS-МЕТОДОЛОГИЯ

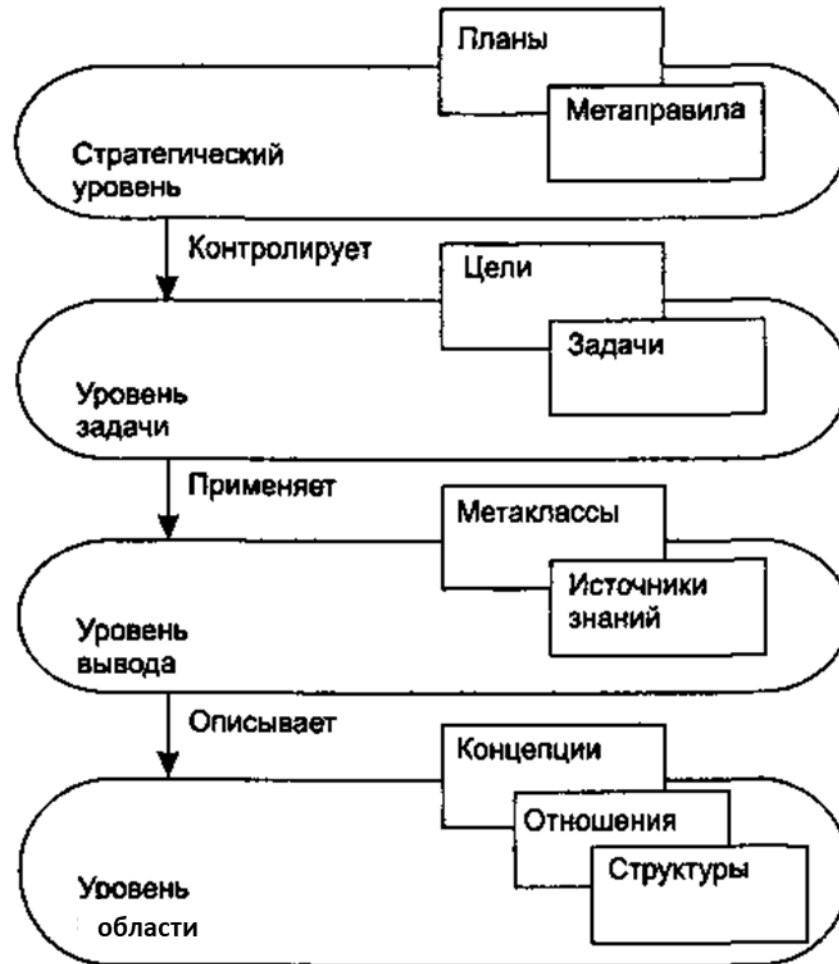
Согласно методологии KADS преобразование знаний осуществляется через спецификацию пяти шагов анализа «идентификация - концептуализация - гносеологический уровень - логический уровень - уровень анализа выполнения» и стадии или пространства проектирования.

Результатом анализа является концептуальная модель экспертизы, состоящая из четырех уровней (уровня области - уровня вывода - уровня задачи - стратегического уровня), которая затем вводится в пространство проектирования и преобразуется в трехуровневую модель проектирования.

При решении реальных задач KADS использует библиотеку интерпретационных моделей, описывающих общие экспертные задачи, без конкретного наполнения объектами предметной области. На основании извлеченных лингвистических данных происходит отбор, комбинация и вложение верхних уровней модели, то есть уровней вывода и задачи, которые наполняются конкретными объектами и атрибутами из уровня области и представляют в результате концептуальную модель рассматриваемой задачи.



ОСНОВНЫЕ МОДЕЛИ KADS



ЖИЗНЕННЫЙ ЦИКЛ KADS

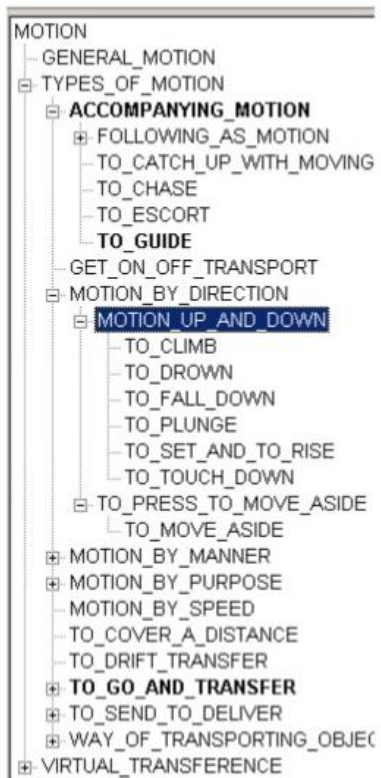


ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИИ ИЗВЛЕЧЕНИЯ ЗНАНИЙ ДЛЯ ПРОЕКТИРОВАНИЯ

Универсальная Семантическая Иерархия

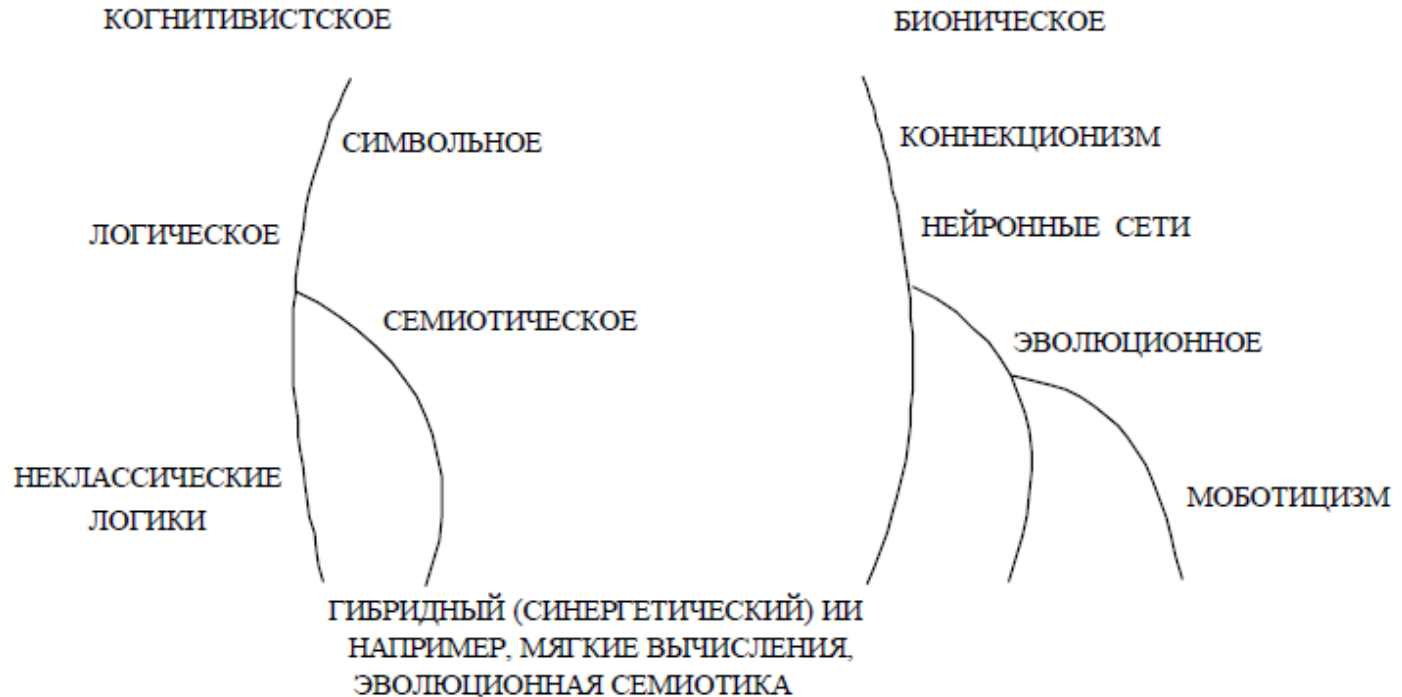
Система понимания и перевода
естественного языка ABBYY Comprendo

- Универсальные семантические описания позволяют легко переводить текст с любого языка в универсальный язык и с универсального – на любой.



СИСТЕМНО-ОРГАНИЗАЦИОННЫЙ ПОДХОД В ИИ В.Б.ТАРАСОВА

В работе «Системно-организационный подход в искусственном интеллекте» (*Программные продукты и системы. – 1999 - №3. – С.6-13*) В.Б.Тарасов отмечает расхождение отдельных направлений ИИ и подчеркивает необходимость их объединения.



КЛАССИЧЕСКИЕ МОДЕЛИ В ИИ

Классические модели ИИ проникнуты рационализмом, поскольку в них интеллект связывается главным образом с рациональными методами решения задач с использованием эвристических процедур. Подобные трактовки интеллекта носят ограничительный и статичный характер, игнорируют реальности возникновения и развития интеллекта человека, связанные с коммуникативными и регулятивными процессами. Классическую парадигму развития ИИ можно рассматривать как несистемную, асоциальную, индивидуалистическую.

Основные постулаты, характеризующих классические интеллектуальные (экспертные) системы первого поколения:

- 1) постулат замкнутости (closed world assumption), определяющий изучение интеллектуальных систем как изолированных, без учета процессов коммуникации с внешней средой и, в частности, с другими интеллектуальными системами;
- 2) постулат централизации, согласно которому интеллектуальная система рассматривается как централизованно управляемая;
- 3) постулат представления (эпистемология ИИ), в соответствии с которым одна из главных задач работ по ИИ состоит в разработке методов внутреннего представления внешнего мира;
- 4) постулат знаниецентризма в ИИ, согласно которому ведущую роль в ИИ играет инженерия знаний, в частности, автоматическое формирование понятий и рассуждений, важнейшее место в моделировании познания занимает логический вывод;
- 5) постулат бестелесности ИИ, утверждающий возможность отрыва интеллекта от тела, например, изучения когнитивных процессов и знаний, абстрагируясь от особенностей регуляции действий и перцептивно-моторных процессов;
- 6) постулат относительной независимости познания и эволюции в ИИ, согласно которому когнитивные процессы можно рассматривать вне и независимо от обучения и эволюции;
- 7) постулат соответствия познания и языка в ИИ: кинематика познания представима в пропозициональных терминах, т.е. траекторию состояний знаний можно описать лингвистически в терминах естественного языка.

ТЕНДЕНЦИИ В ИИ

Двумя наиболее важными тенденциями развития искусственного интеллекта в 90-е годы стали интеграция и децентрализация.

1. В 80-е годы экспертные системы стали объединяться с традиционными информационными технологиями, в результате чего появились **гибридные экспертные системы**. В таких интегрированных системах должны поддерживаться не только различные модели знаний, но и различные типы рассуждений, и реализовываться как инженерия знаний, так и обработка образов .

2. Пошел более фундаментальный процесс **интеграции различных моделей интеллекта** в интересах взаимной компенсации недостатков и объединения преимуществ разнородных моделей и, как следствие, получения синергетических (нелинейных эффектов). Примерами могут служить интегрированные нейро-логические , нейро-нечеткие модели и "мягкие вычисления", нейро-оптические модели, интегрированные модели совмещенной разработки, варианты интеграции нейронных сетей и сетей Петри и пр.

3. Распределение или децентрализация ИИ - связана в первую очередь с исследованием процессов коммуникации, кооперации и координации. При этом распределенные системы ИИ могут иметь единый орган управления, а в децентрализованных системах управление происходит только за счет локальных взаимодействий . В обоих этих случаях интеллектуальные процессы рассматриваются в контексте коллективного поведения, а центральным объектом исследования становится **группа или сообщество интеллектуальных систем**.

ПРИНЦИПЫ СИСТЕМНО-ОРГАНИЗАЦИОННОГО ПОДХОДА В ИИ

1. **Принцип исследования интеллекта в иерархии взаимодействующих систем**, что означает необходимость рассмотрения микро-, мезо- и макроуровней интеллектуальной организации и целесообразность изучения мета-интеллектуальных процедур. Эти уровни соответствуют триаде "субъект – группа - организация". Метаинтеллектуальные процедуры и системы определяют, например, нормы взаимоотношений агентов в многоагентных системах (МАС). Основным признаком метаинтеллектуальной системы является способность предсказать, организовать и управлять коллективным поведением и групповой динамикой агентов в МАС, что означает поддержку рекурсивных и рефлексивных процессов.
2. **Принцип учета коллективной природы интеллекта** (исследование генезиса и развития интеллектуального поведения в процессах коммуникации и совместной деятельности), что предполагает обращение к семиотическим аспектам интеллекта.
3. **Принцип определения рекурсивных связей между интеллектом и деятельностью**, согласно которому интеллект агента выступает как подсистема управления деятельностью, позволяющая ему организовать свои действия или действия другого агента. Интеллект имеет коммуникативную природу и формируется в процессах взаимодействия (коммуникации) агента с другими агентами, а потребность в коммуникации связана с реализацией целенаправленной деятельности.

ПРИНЦИПЫ СИСТЕМНО-ОРГАНИЗАЦИОННОГО ПОДХОДА В ИИ.

ПРОДОЛЖЕНИЕ

4. Принцип приоритета координационных (горизонтальных) связей над вертикальными в интеллектуальных системах и организациях.
5. Принцип дополнительности различных моделей интеллекта (аналогичный принципу Н.Бора), согласно которому невозможно отразить в одной модели многомерный характер понятия интеллект, для этого требуется построение семейства взаимодополняющих моделей.
6. Принцип выделения системных единиц интеллекта. В основу системного синтеза распределенного (децентрализованного) интеллекта целесообразно положить формирование функционально-структурной единицы (ФСЕ) как "универсального строительного блока" или "клетки" МАС. Системные единицы следует отличать от элементов: структурный элемент - это простейшая, неделимая часть системы, которая обычно не сохраняет свойства системы как целого, тогда как важнейшим требованием к функционально-структурной единице является сохранение важнейших свойств организации всей системы.

Основные требования к системной единице интеллекта таковы: а) единица должна быть целостным функционально-структурным образованием, связанной организационно-управленческой структурой; б) единица должна быть способной к развитию (и саморазвитию), т.е. должна обладать порождающими свойствами и возможностями для ее трансформации; в) единица должна быть неоднородной, т.е. отображать в некотором смысле противоположные функции, структуры и свойства интеллекта.

УРОВНИ ОПИСАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Базовыми единицами описания на микроуровне являются **объекты, акторы и агенты**. Классические экспертные системы, базы знаний и подсистемы рассуждения также относятся к этому уровню представления характеристик индивидуального интеллекта. Традиционная проблематика ИИ будет обозначаться термином "**искусственный интеллект-1**".

На мезоуровне основными объектами исследования оказываются группы единиц - **различные интегрированные и многоагентные системы** ("**искусственный интеллект-2**"). Теория МАС зародились на пересечении теории систем и распределенного искусственного интеллекта. С одной стороны, речь идет об **открытых, активных, развивающихся системах**, в которых главное внимание уделяется процессам взаимодействия агентов как причинам возникновения систем с новыми качествами. С другой стороны, достаточно часто МАС строятся по принципам распределенного интеллекта как **объединение отдельных интеллектуальных систем**, обладающих своими базами знаний и средствами рассуждений.

На макроуровне центральное место занимает изучение сообществ (популяций) МАС и моделирование социальных законов, определяющих существование и эволюцию таких сообществ, понимаемых как "искусственные" (виртуальные) или, в ряде случаев, как интеллектуальные организации. Такой организационный уровень исследований в ИИ ("**искусственный интеллект-3**") значительно перекрывается с проблематикой искусственной жизни в широком смысле этого термина.

ПОНЯТИЕ ИНТЕЛЛЕКТУАЛЬНОЙ ОРГАНИЗАЦИИ В ИИ

Интеллектуальная организация определяется как **метаинтеллектуальная МАС**, возникающая в процессах коммуникации, происходящих как внутри организации, так и между отдельными организациями, и обеспечивающих организационный семиозис, которая функционирует в соответствии с принципами максимальной адаптации к динамической среде, самореорганизации в интересах самосохранения и саморегуляции деятельности благодаря сохранению и накоплению знаний и опыта, прогнозированию потребностей и стратегическому планированию, разделенной мотивации и коллективному обучению. В частности, интеллектуальные организации характеризуются процессами интенсивной переработки знаний, связанными с преобразованием интеллектуальных ресурсов в преимущества над конкурентами.

ОПРЕДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА

Под **интеллектуальными агентами** в информатике и ИИ понимаются любые физические или виртуальные единицы, способные:

- а) действовать на объекты в некоторой среде, на других агентов, а также на самих себя (действие);
- б) исходить из некоторых потребностей и образовывать свои собственные цели (потребностно-целевая основа - причина действия);
- в) общаться с другими агентами (общение);
- г) накапливать собственные ресурсы, обеспечивающие их автономию (автономность);
- д) воспринимать некую область среды (восприятие с ограниченным разрешением);
- е) строить частичное представление этой среды на основе ее восприятия, т.е. исходя из имеющихся перцептивных навыков и умений (локальное представление среды);
- ж) нести обязанности и предоставлять услуги (наличие обязательств);
- з) обучаться, адаптироваться к среде и эволюционировать (эволюционный потенциал);
- и) самоорганизоваться и самовоспроизводиться (самосохранение).

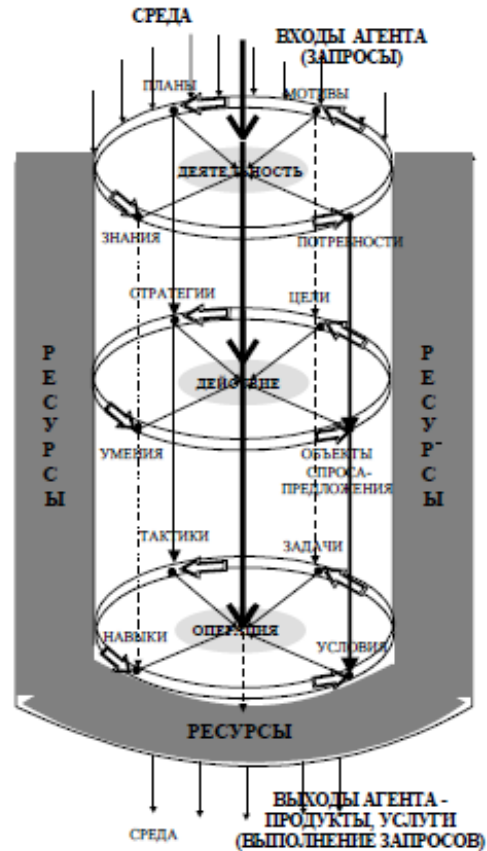
НОВАЯ ПАРАДИГМА В ИИ

Основным объектом исследований в ИИ является группа (или сообщество) неоднородных, взаимодействующих агентов.

Основное содержание разработок связано либо с синтезом индивидуальных свойств и поведения агентов, исходя из заданной групповой динамики, определяемой отношениями кооперации и конкуренции, конфликта и сотрудничества, субординации и координации и пр. (нисходящее проектирование), либо с построением организационных структур из агентов на базе анализа основных функций организации, определения состава агентов и их ролей (восходящее проектирование).

Предметом исследования являются: а) конечное множество агентов; б) ситуации взаимодействия агентов, определяющие возникновение и структуру МАС (тип сообщества агентов) в зависимости от таких параметров, как совместимость целей, наличие взаимных обязательств и ответственности, ограниченность ресурсов, недостаток индивидуального опыта, и пр.); в) множество факторов, обеспечивающих установление локальных пространственных и временных отношений между агентами (непосредственное общение или телекоммуникации, совпадение целей или столкновение интересов, и пр.); г) множество ресурсов взаимодействия, обеспечивающих поддержание некоторых отношений между агентами и их трансформацию; д) множество действий (совместных действий) агентов.

ОБЩАЯ СХЕМА ДЕЯТЕЛЬНОСТИ АГЕНТОВ



- > ПОДСИСТЕМА ПРЕДСТАВЛЕНИЯ
- - -> ИНТЕНЦИОНАЛЬНАЯ ПОДСИСТЕМА
- > ПОДСИСТЕМА ПОВЕДЕНИЯ
- - -> ПОДСИСТЕМА ОПЫТА