

Построение систем с использованием информационных технологий

Этапы проектирования информационных систем

Выделим следующие этапы проектирования ИС:

- I. Исследование предметной области.
- II. Разработка архитектуры системы.
- III. Реализация проекта.
- IV. Внедрение системы.
- V. Сопровождение системы.

I. Исследование предметной области.

Современный уровень инструментальных средств позволяет работать на компьютере на определенном уровне (например, при использовании редакторов) многим пользователям. Однако при углубленной работе с информацией, связанной с ее сбором, созданием базы данных, обработкой информации, представлением для дальнейшего использования, возникают значительные трудности. Это связано с невозможностью работы в компьютерной среде на естественном языке. Вся информация, описывающая конкретную предметную область, должна быть определенным образом абстрагирована и формализована.

Основными направлениями формализации информации о предметной области являются:

- теория классификации, базирующаяся на таксономическом и мерономическом описании информации. Таксономическое описание основано на теории множеств, а мерономическое осуществляется через строго формализованное определение классов;
- теория измерений, предлагающая базу для качественных и количественных измерений через классификационные и порядковые шкалы;
- семиотика, изучающая знаковые системы с точки зрения синтаксиса, семантики и прагматики.

Понятие информации в общем плане должно быть связано с определенной предметной областью, свойства которой она отражает. В более узком плане понятие информации связано с определенным объектом. Информация о любом материальном объекте может быть получена путем наблюдения, натурного либо вычислительного эксперимента, а также на основе логического вывода, поэтому говорят о доопытной, или априорной информации и послеопытной, т. е. апостериорной, полученной в итоге эксперимента.

Предметная область – реальный мир, который должен быть отражен в информационной базе.

Факты – результат наблюдения за состоянием предметной области.

Данные – вид информации, отличающийся высокой степенью форматированности в отличие от более свободных структур, характерных для речевой, текстовой и визуальной информации.

Информационная база (база данных) – совокупность данных, предназначенных для совместного применения.

Знания – итог теоретической и практической деятельности человека, отражающий накопление предыдущего опыта и отличающийся высокой степенью структуризации.

В знаниях можно выделить три основные составные части:

- декларативные (факторальные знания), представляющие общее описание объекта, что не позволяет их использовать без предварительной структуризации в конкретной предметной области;
- понятийные (системные) знания, содержащие помимо первой части взаимосвязи между понятиями и свойства понятий;
- процедурные (алгоритмические) знания, позволяющие получить алгоритм решения.

Предмет – всякая материальная вещь, объект познания. В логике предметом называется все то, на что направлена наша мысль; все то, что может быть как-то воспринято, названо и т. д. В этом смысле предметом считаются также суждение, понятие, умозаключение. В математической логике предметы обозначаются символами – предметными константами и предметными переменными.

Свойство – то, что присуще предметам, что отличает их от других предметов или делает их похожими на другие предметы. Каждый предмет обладает бесчисленным множеством свойств. Свойства проявляются в процессе взаимодействия предметов.

Признак – все то, в чем предметы, явления сходны друг с другом или в чем они отличаются друг от друга; показатель, сторона предмета или явления, по которой можно узнать, определить или описать предмет или явление.

Атрибут (лат. *attributum* – предназначенное, наделенное, присовокупленное) – неотъемлемое, существенное, необходимое свойство, признак предмета или явления, без которого они не могут существовать, быть самими собой, в отличие от случайных, преходящих, несущественных свойств, или акциденций.

Представление предметной области на уровне данных, осуществляется на основе декомпозиции, абстракции, агрегирования.

Декомпозиция – это разбиение предметной области на составные части, осуществляемое по определенным правилам.

Абстракция предполагает продуманный выбор компонент. Процесс абстракции может быть рассмотрен как некоторое обобщение. Он

позволяет забыть о различиях и рассматривать предметы и явления так, как если бы они были эквивалентны.

Обобщение основано на объединении родственных информационных объектов в родовую объект. Например, объекты автомобиль, самолет, корабль, велосипед, мотоцикл объединяем в объект транспортное средство. Одним из атрибутов этого объекта будет атрибут «тип транспортного средства».

Способы абстрагирования:

- абстракция через параметризацию;
- абстракция через спецификацию.

Абстракция через параметризацию – выделение формальных параметров с возможностью их замены на фактические в различных контекстах.

Выделение формальных параметров позволяет абстрагироваться от конкретного приложения и базируется на общности определенных свойств конкретных приложений.

Абстракция через спецификацию позволяет абстрагироваться от внутренней структуры до знания свойств внешних проявлений (результата).

Агрегирование – процесс объединения информационных объектов в один на основе семантических связей между объектами. Например, самолет типа X перевозит груз из пункта отправления А в пункт назначения В. Используя агрегацию создаем информационный объект рейс с атрибутами «тип самолета», «пункт отправления», «пункт назначения», «рейс самолета».

Модель данных – это модель, используемая при абстрагировании. *Концептуальная модель* – абстрагированное описание предметной области.

После знакомства с вопросами терминологии вы получили возможность разговаривать на профессиональном языке и можно перейти к конструированию информационного обеспечения. Первым этапом проектирования ИС является анализ предметной области.

При анализе предметной области принято выделять три стадии:

- анализ требований и информационных потребностей;
- определение информационных объектов и связей между ними;
- конструирование концептуальной модели предметной области.

II. Разработка архитектуры системы.

Среди средств разработки информационных систем выделяют следующие основные группы:

- традиционные системы программирования;
- инструменты для создания файл-серверных приложений;

- средства разработки приложений «клиент – сервер»;
- средства автоматизации делопроизводства и документооборота;
- средства разработки Интернет / интранет-приложений;
- средства автоматизации проектирования (CASE-технологии).

Традиционные системы программирования представлены средствами создания приложений на алгоритмических языках программирования (Си, Паскаль, Бейсик и др.). Инструментальные средства программирования могут быть представлены набором утилит (редактор текстов, компилятор, компоновщик и отладчик) или интегрированной программной средой. Развитием традиционных систем программирования является объектно-ориентированное и визуальное программирование.

Основой разработки файл-серверных приложений для локальных сетей ПК являются инструментальные средства «персональных» СУБД, реализованные в виде диалоговой интегрирующей среды, предоставляющей три уровня доступа:

- программирование и создание приложений;
- создание и ведение структуры БД, а также интерактивная генерация макетного приложения и его компонентов (меню, формы или окна, отчеты, запросы и программные модули);
- использование диалоговой среды и генераторов конечным пользователем для создания, ведения и просмотра БД, а также формирования простых отчетов и запросов.

Среди инструментальных средств реализации приложений с архитектурой «клиент – сервер» выделяют следующие:

- среды разработки приложений для серверов баз данных;
- независимые от СУБД инструменты для создания приложений «клиент – сервер»;
- средства поддержки распределенных информационных приложений.

Среди этой группы следует выделить инструментальные средства быстрой разработки приложений RAD (Rapid Application Development), обеспечивающие реализацию удаленного доступа к СУБД по двухзвенной схеме «клиент – сервер»; связь клиентских приложений с серверами БД с помощью непроцедурного языка структурированных запросов SQL; целостность БД, включая целостность транзакций; поддержку хранимых процедур на серверах БД; реализацию клиентских и серверных триггеров-процедур; генерацию элементов диалогового интерфейса и отчетов.

Средства автоматизации делопроизводства и документооборота подразделяются на следующие подгруппы:

- средства автоматизации учрежденческой деятельности Office Automation;
- системы управления электронным документооборотом EDMS
- электронный документооборот – EDI и европейский стандарт EDI в задачах логистики – UN/EDIFACT;
- средства обеспечения коллективной работы Groupware;
- средства автоматизации документооборота Workflow.

Данная группа средств включает в свой состав: текстовые редакторы для подготовки и корректировки документов; процессоры электронных таблиц для расчетов, анализа и графического представления данных; программы генерации запросов по образцу из различных БД; сетевые планировщики для назначения рабочих встреч и совещаний; средства разработки и демонстрации иллюстративных материалов для презентаций; словари и системы построчного перевода и др. Эти средства представляют собой отдельные пакеты (Win Word, Word Perfect, Excel, Lotus), интегрированный пакет программ (MS Works) или согласованный набор пакетов (Microsoft Office, Corel Perfect Office).

Средства программирования Интернет/интранет-приложений представлены различными системами программирования на интерпретируемых языках Java, Java Script, Tel и др. Построенные с использованием этих средств приложения могут загружаться с любого web-сервера сети и интерпретироваться на клиентском узле. Это обеспечивает платформенную независимость при расширении функциональных возможностей.

Средства автоматизации проектирования приложений (CASE-технологии) предназначены для анализа предметной области, проектирования и генерации программных реализаций. Новые тенденции в реализации приложений связаны с промышленным характером разработки программного обеспечения. Среди существующих инструментальных средств такого типа целесообразно выделить следующие:

- комплект специальных инструментальных средств быстрой разработки прикладных ИС RAD (*Rapid Application Development*);
- технологический комплекс разработки программного обеспечения RUP (*Rational Unified Process*) фирмы Rational Software;
- технология разработки программного обеспечения XP (*Extreme Programming*).

Средства RAD базируются на объектно-ориентированном подходе, когда информационные объекты формируются как действующие модели и их функционирование согласовывается с пользователем. Разработка

приложений на основе RAD ведется с использованием множества готовых объектов, хранимых в виде базы данных. Объектно-ориентированные инструменты RAD в среде GUI позволяют на основе набора стандартных объектов, для которых инкапсулированы атрибуты и внутренние процедуры, формировать простые приложения без написания кода программы. Использование в RAD визуального программирования позволяет еще более упростить и ускорить процесс создания информационных систем. Логика приложения, реализованного с помощью RAD, является событийно-ориентированной, что подразумевает наличие определенного набора событий: открытие и закрытие окон, нажатие клавиши клавиатуры, срабатывание системного таймера, получение и передача управления каждым элементом экрана, некоторые элементы управления базой данных.

Наиболее полным описанием процесса разработки программного обеспечения, включающим методики выполнения работ на каждой стадии жизненного цикла системы, является *Rational Unified Process* (RUP), уникальность которого заключается в том, что это стандартизованный процесс разработки программного обеспечения, используемый многими крупными компаниями по всему миру. RUP обладает следующими преимуществами по сравнению с другими процессами:

- обеспечивает четко организованный подход к назначению задач и требований в рамках организации разработки;
- основан на объектно-ориентированных технологиях разработки программного обеспечения и может использоваться для широкого круга проектов и организаций;
- является итеративным процессом, который допускает расширение проблемы и круга задач по мере последовательного усовершенствования модели и программного обеспечения, позволяя увеличить коэффициент эффективности на протяжении нескольких итераций, что дает большую гибкость в приспособлении к новым требованиям и допускает идентификацию и разрешение рисков разработки заранее;
- создает описание программного продукта, позволяющего восстановить процесс его разработки;
- осуществляет полную поддержку различными инструментальными средствами, позволяющими автоматизировать работы на всех стадиях жизненного цикла программы и сохранить артефакты разработки в электронном виде;
- предоставляет возможность гибкой и перенастраиваемой конфигурации, позволяющей использовать его как для малых групп разработчиков, так и для больших организаций.

В качестве графической нотации в RUP используется *Unified Modeling Language* (UML), являющийся стандартом для представления

объектных моделей. В UML артефакты разработки представляются диаграммами, описывающими структуру программы и ее поведение.

Другим подходом к разработке программного обеспечения является технология *Extreme Programming* (XP). Основными элементами данного подхода являются:

- быстрая разработка программного продукта на основе стандартных шаблонов проектирования;
- постоянное взаимодействие разработчиков с заказчиками системы;
- минимизация затрат на документирование проекта;
- максимальное использование программных тестов («unittests») для проверки функциональности и корректности исходных кодов программ;
- использование рефакторингов для расширения функциональности системы и устранения ее недостатков.

Технология разработки программного обеспечения XP предназначена для использования небольшими группами разработчиков, которым необходимо быстро создать программное обеспечение в условиях постоянно изменяющихся требований.

Разработка программной системы редко начинается «с нуля». Обычно программная система имеет некоторую предысторию в виде совокупности программ, реализующих – частично или полностью – требования к системе. Разработка программ на основе ранее созданных компонент базируется на процессе реинжиниринга программных кодов, при котором путем анализа текстов программ восстанавливается исходная модель программной системы, которая затем используется в новой программе. Главная цель реинжиниринга программного обеспечения – облегчить процесс разработки программных систем за счет повторного использования проверенных решений, а также при переходе на другую аппаратную платформу или на другую среду программирования. Основными задачами реинжиниринга программного обеспечения являются:

- восстановление информации о программной системе, ее документации и спецификаций;
- обнаружение аномалий в архитектуре программной системы, моделях и исходном коде;
- проверка соответствия исходного кода программы решениям, принятым на этапах анализа и проектирования;
- перевод исходных кодов программ с одного языка программирования на другой.

III. Реализация информационных систем на основе информационных технологий.

Рассмотрим кратко основные аспекты и сложившиеся подходы к реализации ИС.

Информационные системы принято разделять по масштабу выполняемых функций на одиночные, групповые и корпоративные.

Одиночные ИС реализуются на автономном компьютере (чаще всего ПК), могут содержать несколько простых приложений, рассчитаны на работу одного пользователя или группы пользователей, разделяющих по времени одно рабочее место. Подобные приложения создаются с помощью так называемых «настольных» СУБД или с помощью файловой системы и диалоговой оболочки для ввода, редактирования и обработки данных.

Групповые ИС ориентированы на коллективное использование информации членами рабочей группы, обычно строятся как локальная вычислительная сеть ПК или реже как многотерминальная вычислительная система. Однотипные или специализированные рабочие места обеспечивают вызов одного или нескольких приложений. Общий информационный ресурс представляет собой базу данных или совокупность файловых структур. При разработке таких систем используются «настольные» СУБД, серверы БД для рабочих групп и соответствующие инструменты разработки.

Корпоративные ИС ориентированы на использование в масштабе предприятия (организации) для различных рабочих групп, могут поддерживать территориально разнесенные узлы или сети. Отличительная особенность таких систем – обеспечение доступа из подразделений к центральной или распределенной БД предприятия (организации), а также к информационным ресурсам рабочей группы. Такие системы реализуются на основе архитектуры «клиент – сервер» со специализацией серверов. При этом используются корпоративные SQL-серверы и соответствующие инструментальные средства.

Групповые и корпоративные информационные системы могут строиться на основе

- многотерминальных централизованных вычислительных систем;
- локальной сети ПК;
- системы с архитектурой «клиент – сервер»;
- системы с распределенными вычислениями;
- офисных систем;
- Интернет / интранет-технологий.

IV. Внедрение системы в действие.

Этот этап проектирования информационных систем состоит из следующих видов работ:

- 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.

V. Сопровождение ИС.

Это выполнение работ в соответствии с гарантийными обязательствами и послегарантийное обслуживание.

Оценка качества информационных систем

Качество ИС связано с дефектами, заложенными на этапе проектирования и проявляющимися в процессе эксплуатации. Любые свойства ИС, в том числе и дефектологические, могут проявляться лишь во взаимодействии с внешней средой, включающей технические средства, персонал, информационное и программное окружение.

В зависимости от целей исследования и этапов жизненного цикла ИС дефектологические свойства разделяют на дефектогенность, дефектабельность и дефектоскопичность.

Дефектогенность определяется влиянием следующих факторов:

- численностью разработчиков ИС, их профессиональными и психофизиологическими характеристиками;
- условиями и организацией процесса разработки ИС;
- характеристиками инструментальных средств и компонент ИС;
- сложностью задач, решаемых ИС;
- степенью агрессивности внешней среды, потенциальной возможностью внешней среды вносить преднамеренные дефекты (например, воздействие вирусов).

Дефектабельность характеризует наличие дефектов ИС и определяется их количеством и местонахождением. Другими факторами, влияющими на дефектабельность, являются:

- структурно-конструктивные особенности ИС;
- интенсивность и характеристики ошибок, приводящих к дефектам.

Дефектоскопичность характеризует возможность проявления дефектов в виде отказов и сбоев в процессе отладки, испытаний или эксплуатации. На дефектоскопичность влияют:

- количество, типы и характер распределения дефектов в ИС;
- устойчивость ИС к проявлению дефектов;

- характеристики средств контроля и диагностики дефектов;
- квалификация обслуживающего персонала.

Оценка качества ИС является крайне сложной задачей ввиду многообразия интересов пользователей. Поэтому невозможно предложить одну универсальную меру качества и приходится использовать ряд характеристик, охватывающих весь спектр предъявляемых требований. Наиболее близки к задачам оценки качества ИС оценки качества программного обеспечения, являющегося одной из важных составных частей ИС. В настоящее время используется несколько абстрактных моделей качества программного обеспечения, основанных на определениях характеристики качества, показателя качества, критерия и метрики.

Критерий может быть определен как независимый атрибут ИС или процесса ее создания. С помощью такого критерия может быть измерена характеристика качества ИС на основе той или иной метрики. Совокупность нескольких критериев определяет показатель качества, формируемый исходя из требований, предъявляемых к ИС. В настоящее время наибольшее распространение получила иерархическая модель взаимосвязи компонент качества ИС. Вначале определяются характеристики качества, в числе которых могут быть, например, общая полезность, исходная полезность, удобство эксплуатации. Далее формируются показатели, к числу которых могут быть отнесены: практичность, целостность, корректность, удобство обслуживания, оцениваемость, гибкость, адаптируемость, мобильность, возможность взаимодействия. Каждому показателю качества ставится в соответствие группа критериев. Для указанных показателей приведем возможные критерии (надо отметить, что один и тот же критерий может характеризовать несколько показателей):

- практичность – работоспособность, возможность обучения, коммуникативность, объем ввода, скорость ввода-вывода;
- целостность – регулирование доступа, контроль доступа;
- эффективность – эффективность использования памяти, эффективность функционирования;
- корректность – трассируемость, завершенность, согласованность;
- надежность – точность, устойчивость к ошибкам, согласованность, простота;
- удобство обслуживания – согласованность, простота, краткость, информативность, модульность;
- оцениваемость – простота, наличие измерительных средств, информативность, модульность;
- гибкость – распространяемость, общность, информативность, модульность;

- адаптируемость – общность, информативность, модульность, аппаратная независимость, программная независимость;
- мобильность – информативность, модульность, аппаратная независимость, программная независимость;
- возможность взаимодействия – модульность, унифицируемость процедур связи, унифицируемость данных.

Одним из путей обеспечения качества ИС является сертификация. В США Радиотехническая комиссия по авиации в своем руководящем документе определяет процесс сертификации следующим образом: «Сертификация – процесс официального утверждения государственным полномочным органом ... выполняемой функции системы ... путем удостоверения, что функция ... удовлетворяет всем требованиям заказчика, а также государственным нормативным документам». К сожалению, в настоящее время не существует стандартов, полностью удовлетворяющих оценке качества ИС. В западно-европейских странах имеется ряд стандартов, определяющих основы сертификации программных систем. Стандарт Великобритании (BS750) описывает структурные построения программных систем, при соблюдении которых может быть получен документ, гарантирующий качество на государственном уровне. Имеется международный аналог указанного стандарта (ISO9000) и аналог для стран членов НАТО (AQAP1).

Контрольные вопросы

1. Какими средствами описывается информационно-логическая модель?
2. Что такое декомпозиция, абстракция, агрегирование?
3. Что такое абстрагирование и формализация информации?
4. Каковы основные этапы анализа предметной области?
5. Какие виды документов используются при анализе предметной области?
6. В чем суть функционально-модульного и объектно-ориентированного подходов при декомпозиции предметной области?
7. Какие типы связей используются для описания взаимодействия объектов предметной области?
8. Что такое концептуальная модель предметной области?
9. Каковы основные виды концептуальных моделей?
10. Каковы основные этапы проектирования информационных систем?
11. Какие модели представления проектных решений вы знаете?
12. Каковы основные критерии качества информационных систем?