

2.1. Процесс выстраивания архитектуры

К настоящему моменту в мировой практике накоплен значительный опыт в области построения архитектуры организации. Существующие подходы и методы во многом базируются на использовании, обобщении и интеграции имеющихся результатов в области бизнес-моделирования и системного анализа и проектирования (прежде всего, в части проектирования информационных систем).

Цикл выстраивания архитектуры организации основными участниками процесса приведен на рис.2.1.



Рис. 2.1. Цикл выстраивания архитектуры организации

Следует отметить, что архитектурная концепция привела к появлению принципиально новой роли в составе основных участников процесса – роли архитектора. Названная роль характеризуется высоким статусом, отражающим степень важности наличия у организации архитектуры, архитектор, как правило, является главным заместителем ИТ-директора и

является постановщиком задач как для аналитиков бизнес-процессов, так и для системных аналитиков.

Основными этапами процесса построения архитектуры организации являются следующие:

1. осознание необходимости построения архитектуры;
2. формирование рабочей группы;
3. выбор среды моделирования, средств моделирования и репозитория;
4. наполнение среды фактическим материалом (формирование архитектуры);
5. использование;
6. расширение и сопровождение.

При этом на этапе формирования архитектуры, как наиболее трудоемком, решаются следующие задачи, собственно, относящиеся к моделированию:

1. определение бизнес-целей и требований;
2. моделирование бизнеса с позиции менеджера;
3. моделирование бизнес-процессов;
4. моделирование бизнес-функций;
5. моделирование оргструктуры, включая логические схемы принятия решений;
6. моделирование ресурсов;
7. преобразование бизнес-моделей в модели приложений и технологической архитектуры.

Отметим, что задача моделирования бизнеса с позиции менеджера, фактически, впервые идентифицируется в качестве самостоятельного элемента методологии. И хотя ряд инструментов "доархитектурного" периода обеспечивал построение так называемых "презентационных диаграмм", данный этап не декларировался практически ни одной из известных методологий бизнес-моделирования.

2.2. Базовые модели классических подходов

Основой современных подходов к построению моделей бизнес-слоя и системного слоя архитектуры являются методологии структурного и объектно-ориентированного анализа и проектирования.

Структурным анализом принято называть метод исследования системы, которое начинается с ее общего обзора и затем детализируется, приобретая иерархическую структуру со все большим числом уровней. Для методов данного класса характерно:

1. разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 6-7, при этом верхняя граница соответствует возможностям человеческого мозга по восприятию определенного количества взаимоувязанных объектов, а нижняя выбрана из соображений здравого смысла);
2. ограниченный контекст, включающий лишь существенные на каждом уровне детали;
3. использование строгих формальных правил записи;
4. последовательное приближение к конечному результату.

Все методологии структурного анализа базируются на ряде общих принципов, *регламентирующих организацию работ по моделированию*. В качестве двух базовых принципов используются следующие: принцип "разделяй и властвуй" и принцип иерархического упорядочивания. Первый является принципом решения трудных проблем путем разбиения их на множество меньших независимых задач, легких для понимания и решения (так называемых "черных ящиков" – его пользователю не требуется знать, как он работает, необходимо знать лишь его входы и выходы, а также его назначение, т.е. функцию, которую он выполняет). Второй принцип в дополнение к тому, что легче понимать систему, когда она разбита на части, декларирует, что устройство этих частей также существенно для понимания. Понимаемость системы резко повышается при организации ее частей в древовидные иерархические структуры, т.е. система может быть понята и построена по уровням, каждый из которых добавляет новые детали.

Важным моментом облегчения понимаемости сложных систем является широкое использование структурными методами графических нотаций. Известно, что "одна картинка стоит тысячи слов", читатель впитывает такое описание с первого взгляда. Структурные методы также позволяют дополнить "картинки" любой дополнительной информацией, которая не может быть отражена при использовании соответствующей графической нотации.

Для целей структурного анализа и проектирования традиционно используются три группы средств, иллюстрирующих:

1. функции, которые система должна выполнять (более точно – функциональную структуру системы, последовательность выполняемых действий, передачу информации между элементами функциональной структуры);
2. отношения между данными;
3. динамическое поведение системы (аспекты реального времени).

Среди всего многообразия графических нотаций, используемых для решения первой задачи, в структурных методологиях наиболее часто и эффективно применяемыми являются следующие:

1. DFD (DataFlowDiagrams) - диаграммы потоков данных совместно со спецификациями процессов нижнего уровня (миниспецификациями);
2. SADT (StructuredAnalysisandDesignTechnique) – диаграммы (точнее, их стандартизованное подмножество – модель IDEF0);
3. модель IDEF3.

Диаграммы потоков данных представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами. Для построения DFD традиционно используются две различные нотации, соответствующие методам Йордона-ДеМарко и Гейна-Сэрсона. Эти нотации незначительно отличаются друг от друга графическим изображением символов. В соответствии с данными методами модель системы определяется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи потребителю (при этом для описания функционирования процесса в случае отсутствия необходимости детализировать его с помощью DFD используется миниспецификация, фактически представляющая собой алгоритм описания задачи, выполняемой процессом, при этом множество всех миниспецификаций является полной спецификацией системы). Практически любой класс систем успешно моделируется при помощи DFD-ориентированных методов. Они с самого начала создавались как средство проектирования информационных систем (тогда как SADT - как средство моделирования систем вообще) и имеют более богатый набор элементов, адекватно отражающих специфику таких систем (например, хранилища данных являются прообразами файлов или баз данных, внешние сущности отражают взаимодействие моделируемой системы с внешним миром).

Метод SADT представляет собой совокупность правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями. Метод SADT был разработан Дугласом Россом для моделирования систем средней сложности. Метод SADT поддерживается Министерством обороны США, которое было инициатором разработки семейства стандартов IDEF (IcamDEFinition), являющегося основной частью программы ICAM (интегрированная компьютеризация производства), проводимой по инициативе ВВС США. Метод SADT реализован в одном

стандартов этого семейства - IDEF0, который был утвержден в качестве федерального стандарта США в 1993 г.

Модели SADT/IDEF0 (в отличии от DFD) традиционно используются для моделирования бизнес-слоя архитектуры. Следует отметить, что метод SADT успешно работает только при описании хорошо специфицированных и стандартизованных бизнес-процессов в зарубежных корпорациях, поэтому он и принят в США в качестве типового. Достоинствами применения моделей SADT для описания бизнес-слоя являются:

1. полнота описания бизнес-процесса (управление, информационные и материальные потоки, обратные связи);
2. соответствие подхода к описанию процессов стандартам ISO 9000;
3. жесткие требования метода, обеспечивающих получение моделей стандартного вида (здесь необходимо отметить, что в большинстве российских организаций бизнес-процессы начали формироваться и развиваться сравнительно недавно, они слабо типизированы, поэтому разумнее ориентироваться на менее жесткие модели).

Метод моделирования IDEF3, являющийся частью семейства стандартов IDEF, предназначен для таких моделей процессов, в которых важно понять последовательность выполнения действий и взаимозависимости между ними. Хотя IDEF3 и не достиг статуса федерального стандарта США, он приобрел широкое распространение среди аналитиков как дополнение к методу функционального моделирования IDEF0 (модели IDEF3 могут использоваться для детализации функциональных блоков IDEF0, не имеющих диаграмм декомпозиции). Основой модели IDEF3 служит так называемый сценарий процесса, который выделяет последовательность действий и подпроцессов анализируемой системы.

Следует особо отметить, что вышеперечисленные три разновидности структурного анализа и проектирования примерно одинаковы с точки зрения возможностей изобразительных средств функционального моделирования. При этом одним из основных критериев выбора того или иного метода является степень владения им со стороны консультанта или аналитика, грамотность выражения своих мыслей на языке моделирования. В противном случае в моделях, построенных с использованием любого метода, будет невозможно разобраться.

Наиболее распространенным средством моделирования отношений между данными (информационного моделирования) является диаграмма "сущность-связь" (Entity-RelationshipDiagram- ERD), известная в двух нотациях – Чена и Баркера. Она предназначена для обеспечения стандартного способа определения данных и отношений между ними, с ее помощью документируются информационные аспекты системы, включая

идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их связей с другими объектами (отношений). ERD традиционно используется в структурном анализе и проектировании, однако, по существу, представляет собой подмножество объектной модели предметной области. Одна из разновидностей модели "сущность-связь" используется в методе IDEF1/IDEF1X, входящем в семейство стандартов IDEF.

Для моделирования поведенческих аспектов системы традиционно применяется диаграмма переходов состояний (StateTransitionDiagram - STD), описывающая множество системных состояний и определяющая перемещение моделируемой системы из одного состояния в другое (при этом идентифицируется условие, являющееся причиной перехода и управляющее им, а также действие, производимое при переходе из одного состояния в другое).

Все вышеперечисленные модели содержат графические и текстовые средства моделирования: первые - для удобства отображения основных компонент модели, вторые - для обеспечения точного определения ее компонент и связей.

Перечисленные выше графические нотации используются (в том или ином наборе) практически во всех современных структурных методологиях. Роль этих методологий заключается в регламентации основ моделирования бизнес-слоя и системного слоя ЕА. Они описывают последовательность шагов, модели и подходы, тщательное следование которым приведет к хорошим результатам. Хотя методологии, вообще говоря, не гарантируют качества построенных моделей, тем не менее, они помогают охватить и учесть все важные этапы, шаги и моменты разработки, помогают справиться с проблемами размерности, и в конечном итоге оценить продвижение вперед. Более того, методологии обеспечивают организационную поддержку, позволяющую большим коллективам исполнителей функционировать скоординированным образом.

Другими словами, структурная методология определяет руководящие указания для построения (и дальнейшей) оценки архитектурных моделей, шаги работы, которые должны быть выполнены, их последовательность, а также правила распределения и назначения применяемых при этом операций и методов.

Важнейшей характеристикой структурной методологии является порядок построения модели. В соответствии с ним все методологии классифицируются на два вида - функционально-ориентированные и информационно-ориентированные. Традиционный функционально-ориентированный подход регламентирует первичность проектирования

функциональных компонент по отношению к проектированию структур данных: требования к данным раскрываются через функциональные требования. При информационно-ориентированном подходе вход и выход являются наиболее важными - структуры данных определяются первыми, а процедурные компоненты являются производными от данных. Предпочтительное использование функционально-ориентированных подходов связано с тем, что современная организация характеризуется переносом центра тяжести на слой бизнес-правил. Модель процесса является ценным средством для размышлений и совместной работы над перспективами развития организации и системной разработкой, поскольку руководство прекрасно ориентируется в технологиях и бизнес-процессах организации, и функциональные модели (в отличие от информационных) интуитивно понимаемы неспециалистами. Кроме того, информационная модель, как правило, представляет собой единственную диаграмму, возможно содержащую несколько сотен объектов, тогда как функциональная иерархическая модель может включать десятки тысяч объектов. Тем не менее, информационная модель продолжает оставаться важной и соответствующим образом влиять на разрабатываемую функциональную модель. Подтверждением первичности функциональной модели является тот факт, что на Западе, где различные методики реорганизации (BSP, CPI/TQM, BPR) применяются уже длительное время, большинство методологий являются функционально-ориентированными.

Важное место в разработках архитектурных моделей занимают объектно-ориентированные методологии, основанные на объектной декомпозиции предметной области, представляемой в виде совокупности объектов, взаимодействующих между собой посредством передачи сообщений. В качестве объектов предметной области могут рассматриваться конкретные предметы, а также абстрактные или реальные сущности (например, клиент, заказ, организация и т.п.). Каждый объект характеризуется своим состоянием (точнее, набором атрибутов, значения которых определяют состояние), а также набором операций для проверки и изменения этого состояния. Каждый объект является представителем некоторого класса однотипных объектов, определяющего их общие свойства. Все представители (экземпляры) одного и того же класса имеют один и тот же набор операций и могут реагировать на одни и те же сообщения.

Объекты и классы организуются с использованием следующих принципов:

1. Принцип инкапсуляции (упрятывания информации) декларирует запрещение любого доступа к атрибутам объекта, кроме как через его операции. В соответствии с этим внутренняя структура объекта скрыта от пользователя, а любое его действие инициируется внешним сообщением, вызывающим выполнение соответствующей операции.

2. Принцип наследования декларирует создание новых классов от общего к частному. Такие новые классы сохраняют все свойства классов-родителей и при этом содержат дополнительные атрибуты и операции, характеризующие их специфику.
3. Принцип полиморфизма декларирует возможность работы с объектом без информации о конкретном классе, экземпляром которого он является. Каждый объект может выбирать операцию на основании типов данных, принимаемых в сообщении, т.е. реагировать индивидуально на это (одно и то же для различных объектов) сообщение.

Таким образом, объектно-ориентированный подход заключается в представлении моделируемого процесса в виде совокупности классов и объектов предметной области. При этом иерархический характер сложного процесса отражается с использованием иерархии классов, а его функционирование рассматривается как взаимодействие объектов.

Большинство современных методов объектно-ориентированного подхода основано на использовании унифицированного языка моделирования UML (Unified Modeling Language) - фактического приемника наиболее распространенных объектно-ориентированных методов Буча, Рамбо и Якобсона, представляющего собой язык для определения, представления, проектирования и документирования программных систем, организационно-экономических систем, технических систем и других систем различной природы.

UML находится в процессе стандартизации, проводимом OMG (Object Management Group) - организацией по стандартизации в области объектно-ориентированных методов и технологий, в настоящее время принят в качестве стандартного языка моделирования и получил широкую поддержку в индустрии ПО. Стандарт UML версии 1.1, принятый OMG в 1997 г., содержит следующий набор диаграмм:

1. Структурные (structural) модели:
 - диаграммы классов (classdiagrams) - для моделирования статической структуры классов системы и связей между ними;
 - диаграммы компонентов (componentdiagrams) - для моделирования иерархии компонентов (подсистем) системы;
 - диаграммы размещения (deploymentdiagrams) - для моделирования физической архитектуры системы.
2. Модели поведения (behavioral):
 - диаграммы вариантов использования (usecasediagrams) - для моделирования функциональных требований к системе (в виде сценариев взаимодействия пользователей с системой);

- диаграммы взаимодействия (interaction diagrams): диаграммы последовательности (sequence diagrams) и кооперативные диаграммы (collaboration diagrams) - для моделирования процесса обмена сообщениями между объектами;
- диаграммы состояний (state chart diagrams) - для моделирования поведения объектов системы при переходе из одного состояния в другое
- диаграммы деятельности (activity diagrams) - для моделирования поведения системы в рамках различных вариантов использования, или потоков управления.

UML обладает механизмами расширения для адаптации языка моделирования к конкретным нуждам. Наличие механизмов расширения принципиально отличает UML от таких средств моделирования, как IDEF0, IDEF1, IDEF3, DFD, STD и ERD. Перечисленные языки моделирования можно определить как сильно типизированные, поскольку они не допускают произвольной интерпретации семантики элементов моделей. UML, допуская такую интерпретацию, является слабо типизированным языком.

Основой взаимосвязи между структурным и объектно-ориентированным подходами является общность ряда категорий и понятий обоих подходов (процесс и вариант использования, сущность и класс и др.). Эта взаимосвязь может проявляться в различных формах. Так, одним из возможных вариантов является использование структурного анализа как основы для объектно-ориентированного проектирования. При этом структурный анализ следует прекращать при переходе от бизнес-слоя к системному слою архитектуры. Другой формой проявления взаимосвязи можно считать интеграцию объектной и реляционной технологий. Реляционные СУБД являются на сегодняшний день основным средством реализации крупномасштабных баз данных и хранилищ данных. Причины этого достаточно очевидны: реляционная технология используется достаточно долго, освоена огромным количеством пользователей и разработчиков, стала промышленным стандартом, в нее вложены значительные средства и создано множество корпоративных БД в самых различных отраслях, реляционная модель проста и имеет строгое математическое основание; существует большое разнообразие промышленных средств проектирования, реализации и эксплуатации реляционных БД. Вследствие этого реляционные БД в основном используются для хранения и поиска объектов в так называемых объектно-реляционных системах.

2.3. Особенности языка ARIS

В настоящее время наблюдается тенденция интеграции разнообразных методов моделирования и анализа систем, проявляющаяся в форме создания интегрированных средств моделирования. Одним из таких средств является продукт, носящий название ARIS - Architecture of Integrated Information System, разработанный германской фирмой IDS Scheer. Его методическую основу составляет совокупность различных методов моделирования, отражающих разные взгляды на исследуемую систему. Одна и та же модель может разрабатываться с использованием нескольких методов, что позволяет использовать ARIS специалистам с различными теоретическими знаниями и настраивать его на работу с системами, имеющими свою специфику.

ARIS поддерживает четыре типа моделей, отражающих различные аспекты исследуемой системы:

1. организационные модели, представляющие структуру системы - иерархию организационных подразделений, должностей и конкретных лиц, связи между ними, а также территориальную привязку структурных подразделений;
2. функциональные модели, содержащие иерархию целей, стоящих перед аппаратом управления, с совокупностью деревьев функций, необходимых для достижения поставленных целей;
3. информационные модели, отражающие структуру информации, необходимой для реализации всей совокупности функций системы;
4. модели управления, представляющие комплексный взгляд на реализацию бизнес-процессов в рамках системы.

Все эти модели представляют собой диаграммы, элементами которых являются разнообразные объекты - "функция", "событие", "структурное подразделение", "документ" и т.п. Между объектами устанавливаются разнообразные связи. Каждому объекту соответствует определенный набор атрибутов, которые позволяют ввести дополнительную информацию о конкретном объекте. Значения атрибутов могут использоваться при имитационном моделировании или для проведения стоимостного анализа.

Для построения перечисленных типов моделей используются как собственные методы моделирования ARIS, так и различные известные методы и языки моделирования. В процессе моделирования каждый аспект деятельности организации сначала рассматривается отдельно, а после детальной проработки всех аспектов строится интегрированная модель, отражающая все связи между различными аспектами.

Относительно новой моделью ARIS является диаграмма eEPC (extended Event Driven Process Chain - расширенная модель цепочки процессов, управляемых событиями). По существу, она расширяет возможности IDEF0, IDEF3 и DFD, обладая всеми их достоинствами и недостатками. eEPC-

диаграмма предназначена для детального описания бизнес-процесса и отражает логику его выполнения. Бизнес-процесс в нотации eEPC представляет собой поток последовательно выполняемых работ (процедур, функций), расположенных в порядке их выполнения. Используемые при построении модели символы логики позволяют отразить ветвление и слияние ветвей бизнес-процесса. Исполнители, документы и элементы прикладных комплексов привязываются к бизнес-функциям. Условия выполнения бизнес-функций, а также их результаты отражаются посредством событий. Другими словами, детальная модель бизнес-процесса представляет собой последовательность событий и бизнес-функций, обеспечивающую достижение заданного результата.

Следует отметить, что Архитектура Интегрированных Информационных Систем ARIS предназначена, прежде всего, для разработки информационных систем – от определения требований до описания реализации. Тем не менее, она охватывает и предшествующие этапы, связанные с документированием корпоративных знаний, а именно, описанием организационной структуры, моделированием бизнес-процессов, документированием архитектуры приложений, данных и оборудования и т.п.